File Number 1800-36
Order Number GC26-3725-4

IBM Systems Reference Library

IBM 1800 Multiprogramming Executive Operating System Operating Procedures

Fifth Edition (August 1974)

This is a reprint of GC26-3725-3 incorporating changes released in the following Technical Newsletters:

GN34-0095 (October 1974)
GN34-0188 (February 1974)

This edition applies to version 3, modification 4, of the IBM 1800 Multiprogramming Executive Operating System, and to all subsequent versions and modifications unless otherwise indicated in new editions or technical newsletters. Changes may be made to the specifications in this manual at any time; before using this manual in connection with the operation of IBM systems, consult the latest SRL Newsletter, GN26-1800, for the editions that are applicable and current.

A form for readers' comments is provided at the back of this publication. If the form has been removed, send your comments to IBM Corporation, Systems Publications, Department 27T, P. O. Box 1328, Boca Raton, Forida 33432. Comments become the property of IBM.

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

Preface

This manual describes operating procedures for the IBM 1800 Multiprogramming Executive (MPX) Operating System. It contains procedures for system generation, reload, Cold Start, batch and background processing, maintenance, utility, and debugging operations.

Before using this book, you should be familiar with the basic concepts and terms associated with MPX. They're explained in the MPX System Introduction manual, Order Number GC26-3718. You should also be familiar with the machine units that make up your 1800 system. They're described in the 1800 System Summary, Order Number GA26-5920.

While using this book to operate the 1800, you'll need some other books:

1800 Functional Characteristics, Order Number GA26-5918, which describes how the physical units of the 1800 system work.

1800 Operating Procedures, Order Number GA26-5953, which tells how to operate the physical units.

MPX Error Messages and Recovery Procedures, Order Number GC26-3727, which lists all the error messages MPX prints and tells what conditions they indicate and how to recover from these conditions.

MPX Control Statements, Order Number GX26-1594, which gives the format of each of the Supervisor, DMP, Builder, Macro Assembler, FORTRAN Compiler, Cold Start, and System Loader control statements.

As you read this manual, you might want more detailed information about 1800 system physical units and MPX system programs. These books can be used for reference:

MPX Programmer's Guide, Order Number GC26-3720, which discusses MPX system organization and programming techniques.

MPX Planning for Versions 2 and 3, Order Number GC26-3731, which describes the features of MPX added in Versions 2 and 3 of the system.

MPX Subroutine Library, Order Number GC26-3724, which describes each of the system subroutines.

1130/1800 Assembler Language, Order Number GC26-3778, which tells how to use macro instructions and write programs in assembler language.

1130/1800 Basic FORTRAN IV Language, Order Number GC26-3715, which tells how to write programs in FORTRAN.

Communications Adapter Programming, Order Number GC26-3757, which tells how to write programs to carry out communications with other computers and terminals.

Binary Synchronous Communications — General Information, Order Number GA27-3004, which describes the programming conventions that govern communications between the 1800 and other computers and terminals.

1130/1800 Plotter Subroutines, Order Number GC26-3755, which describes MPX subroutines for controlling the 1627 plotter.

This book consists of eight chapters and four appendixes.

The first chapter, "I/O Device and Console Operations", describes procedures for various operations on system I/O devices and on the 1800 console.

The second chapter, "MPX System Generation," describes the process of defining and establishing the capabilities, contents and organization of your particular MPX system from materials supplied by IBM Program Information Department. You don't need to read this chapter unless you're going to be carrying out a system generation.

The third chapter, "Cold Start," describes the process of starting your batch-processing system (under BOM) or your real-time system (under the Executive) once your system has been generated.

The fourth chapter, "BOM/Executive Reload," summarizes the functions performed during a system reload and details manual operating procedures for performing a reload.

The fifth chapter, "Background-Processing Operations," discusses the procedures for passing control to the Batch-Processing Monitor Supervisor. It also discusses the types of input to the Supervisor.

The sixth chapter, "DMON — System Maintenance Program," describes the operating procedures for the DMON program, which is used to alter and update system programs.

The seventh chapter, "BOM Card Utilities," describes the functions and operating procedures for the various BOM card utility programs punched during system generation.

The eighth chapter, "MPX Debugging Aids," describes the functions and operating procedures for the various system debugging aids.

System Loader assignment cards, decimal and hexadecimal disk addresses, data formats, and an MPX sample program are described in the appendixes.

Contents

Illustrations

Figures

Tables

I/O Device and Console Operations

Operations on an 1800 MPX system require a familiarity with some I/O device and console operations.

I/O Device Operations

Loading, readying, error recovery, and unloading procedures for various I/O devices follow.

1442 CARD READ PUNCH, MODELS 6, 7

Readying and error recovery procedures for the 1442 card read punch follow.

PRE-CONDITIONS

System power ON, CHECK light OFF, CHIP BOX light OFF, stacker not full, covers closed.

READYING PROCEDURE

1. Place cards to be read or punched into the hopper, face down, 9-edge first.
2. Press reader START.

The READY light turns on when the first card is positioned at the read station.

1442 ERRORS AND RECOVERY PROCEDURES

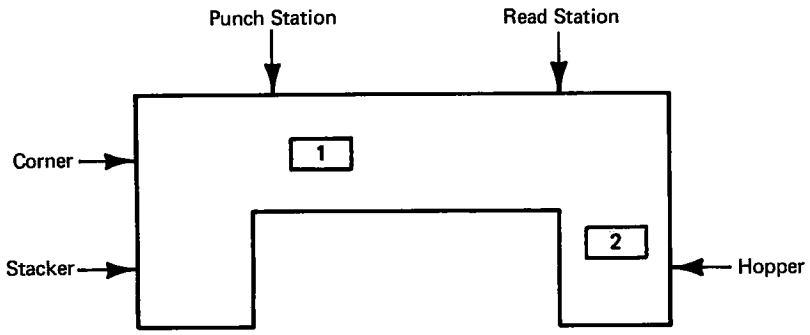
If a 1442 error occurs, the 1442 becomes not ready until the operator has intervened. Unless the stop is caused by a stacker full (no indicator) or chip box indication, the 1442 card path must be cleared before proceeding. The 1442 error indicators and the position of the cards in the feed path should be used to determine which cards must be placed back into the hopper.

As far as the card subroutines are concerned, a retry consists of positioning the cards as indicated in the following paragraphs and reinitiating the read or punch operation when the unit becomes ready. The card subroutines will skip the first card, if necessary, on a read or feed operation.

HOPPER MISFEED

Indicates that card 2 failed to pass properly from the hopper to the read station during the card 1 feed cycle.

Card positions after error:



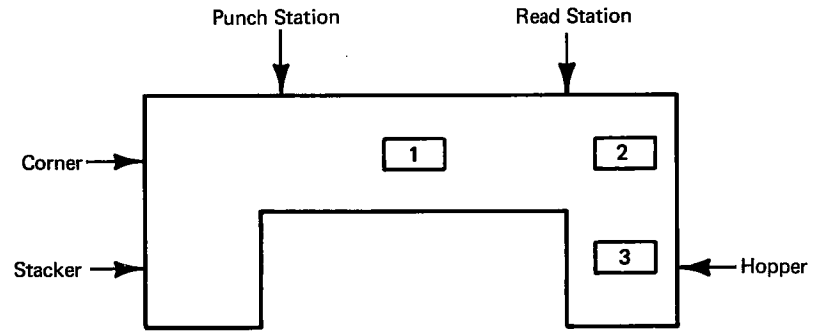
Error indicator: HOPR

Recovery procedure: Empty hopper, press NPRO to eject card 1, replace card 1 into the deck, return deck to hopper, and ready the 1442.

FEED CHECK (PUNCH STATION)

Indicates that card 1 is improperly positioned in the punch station at the completion of its feed cycle.

Card positions after error:



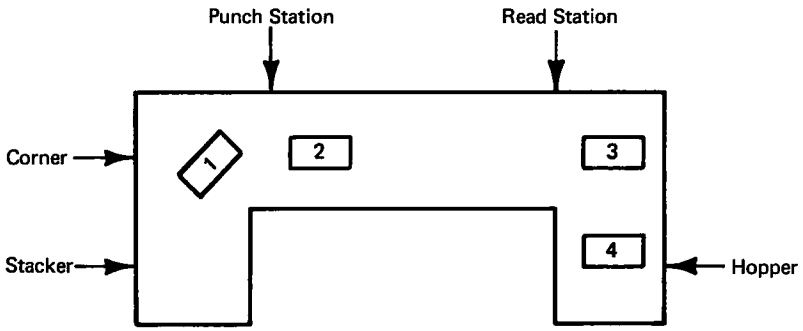
Error indicator: PUNCH STA

Recovery procedure: Empty hopper, clear 1442 card path. If reading, place card 2 into hopper before card 3, and ready the 1442. If punching, place cards 1 and 2 in hopper before card 3 and ready the 1442.

TRANSPORT

Indicates that card 1 has jammed in the stacker during the feed cycle for card 2.

Card positions after error:



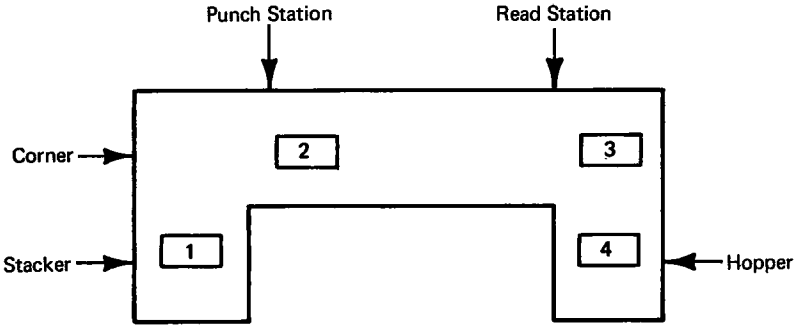
Error indicator: TRANS

Recovery procedure: Empty hopper, clear 1442 card path, place cards 2 and 3 into the hopper, and ready the 1442.

FEED CYCLE

Indicates that the 1442 did an uncalled-for feed cycle, resulting in cards 1, 2, and 3 being each one station farther ahead in the 1442 card path than they should be.

Card positions after error:



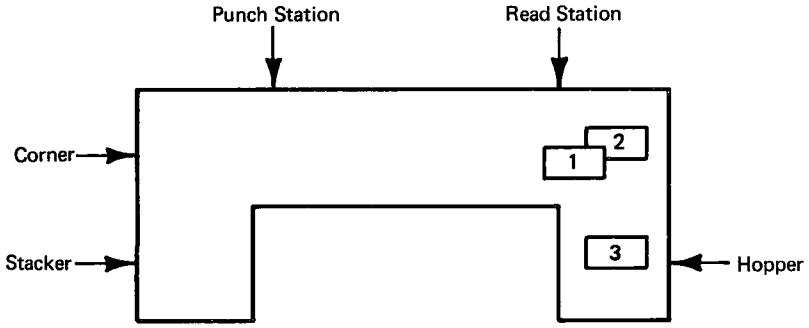
Error indicator: FEED CLU

Recovery procedure: Empty hopper, press NPRO to eject cards 2 and 3, place cards 1, 2, and 3 into the hopper, and ready the 1442.

FEED CHECK (READ STATION)

Indicates that card 1 failed to eject from the read station during its feed cycle.

Card positions after error:



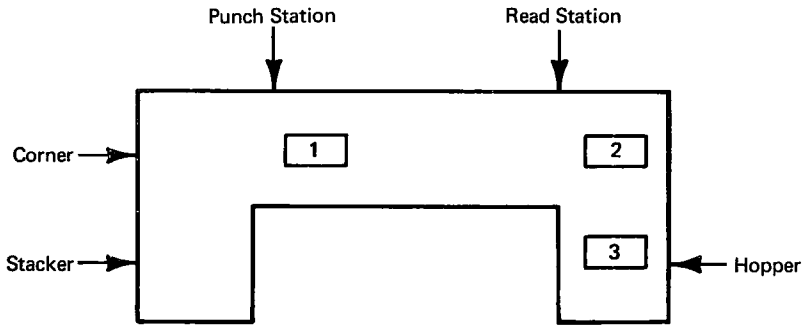
Error indicator: READ STA

Recovery procedure: Empty hopper, clear 1442 card path, place cards 1 and 2 into the hopper, and ready the 1442.

READ REGISTRATION

Indicates incorrect card registration or a difference between the first and second reading of a column.

Card positions after error:



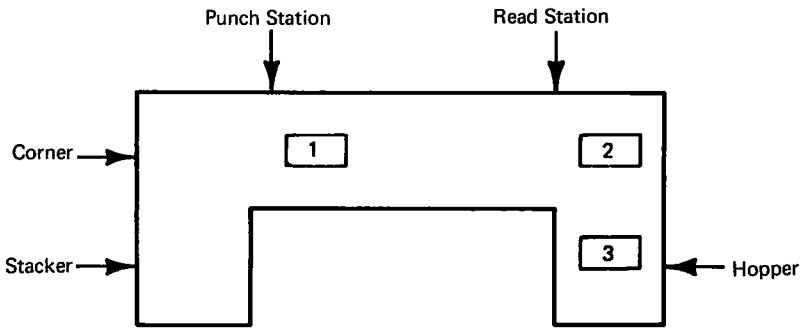
Error indicator: READ REG

Recovery procedure: Empty hopper, press NPRO to eject card 1, place card 1 into the deck in front of card 2, return deck to hopper, and ready the 1442. Repeated failures of this type might indicate a machine malfunction.

PUNCH CHECK

Indicates an error in output punching.

Card positions after error:



Error indicator: PUNCH

Recovery procedure: Empty hopper, check card position and press NPRO to clear 1442 card path. If necessary, correct card 1 to prepunched state. Place corrected card 1, and card 2 into the hopper before card 3 and ready the 1442.

OVER RUN

Indicates that a cycle steal request was not honored in time and that data has been lost. Card positions indicate the kind of operation, read or punch.

If the operation was a read, the card positions after the error and the recovery procedures are the same as for Read Registration.

If the operation was a punch, the card positions after the error and the recovery procedures are the same as for Punch Check.

Error indicator: OVER RUN

PARITY

Indicates incorrect parity detected on data transfer to or from B-register. See "Read Registration" or "Punch Check" for card positions after error and for recovery procedures. The card read punch will be ready.

Error indicator: NONE

STORAGE PROTECT

Indicates attempt to read a column into a storage protected location. See "Read Registration" for card position after error and operator recovery procedures. The card read punch will be ready.

Error indicator: NONE

1816/1053 PRINTER

Readying and loading procedures for the 1816/1053 printer follow:

PRE-CONDITIONS

System power ON, 1816 motor switch ON.

READYING PROCEDURE

Press CARRIER RETURN.

LOADING PROCEDURE

1. Open the 1816/1053 top cover.
2. Pull the paper pressure rod (the rod with three rubber rollers that leans against the platen) forward. If the paper is to be pin fed, this rod should remain in this position.
3. Lift up on the left and right platen pin feed pressure plates.
4. Set the paper release lever in the forward position. This lever is located on the top right rear corner of the 1816/1053. If the paper is to be pin fed, this lever should remain in this position.
5. Feed the paper in from the rear and guide it under the platen. Make sure that the paper lies over and closes the Forms Check microswitch.
6. Lay the paper back across the top of the 1816/1053 and guide the paper so that the holes line up with the pin feeds.
7. Close the pin feed pressure plates.
8. Looking directly down into the 1816/1053 set the left and right margins. The margin settings can be read on the scale across the front of the unit. Use the TAB key to move the carrier off the left margin.
9. Close the top cover.

1054 PAPER TAPE READER

The readying procedure follows.

PRE-CONDITIONS

System power ON.

PROCEDURE

1. Place the reel of tape on the supply reel on the right side of the reader. The tape feeds under and out towards the front of the unit.
2. Feed the tape under the guide in front of the reel and bring it up over the side and across the unit.
3. Press the TABLE RELEASE key.
4. Slip the tape under the plate and position the tape so that a delete code in the reader is over the read head. If you are starting to read at a point other than at the beginning of the tape, place the first character to be read over the read head.
5. Press down on the table to lock the tape in place.

1055 PAPER TAPE PUNCH

The readying and loading procedures follow.

PRE-CONDITIONS

System power ON.

READYING PROCEDURE

To make a leader (all delete codes):

1. Press and hold the DELETE key.
2. Press and hold the FEED key until a leader of sufficient length has been punched.
3. Release the FEED key.
4. Release the DELETE key.

LOADING PROCEDURE

1. Place a reel of tape in the supply pan so that the tape feeds out towards the punch die (see illustration below).

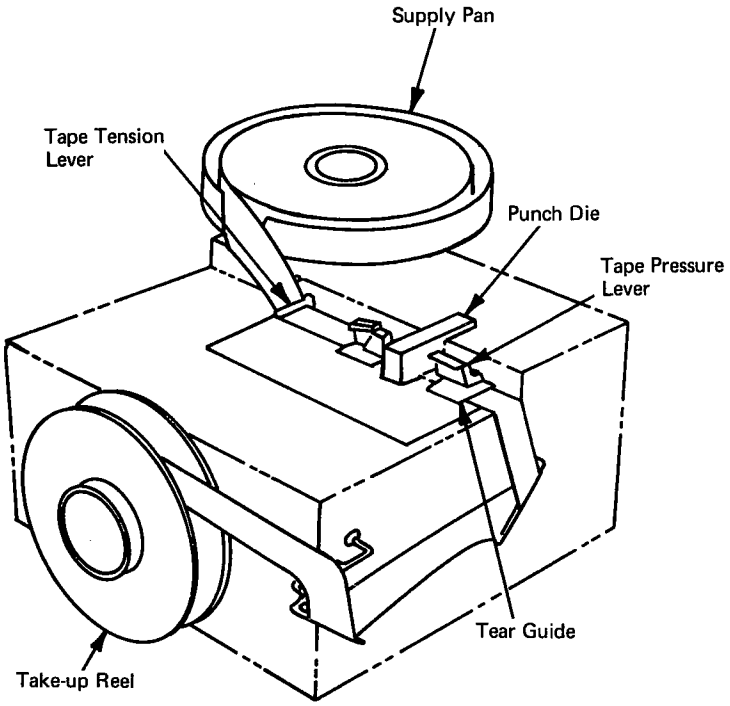
2. With the punch die facing forward (unit name plate at the front), pivot the tape pressure lever (right side of die) up and to the right.

3. Feed the tape from the supply pan over the first tape guide and under the tape tension lever, and slide the tape in under the punch die, tear guide, and tape pressure lever.

4. If the punch has a takeup reel, guide the tape over the side of the unit, over the outside of the side guide, and back up toward the front of the unit.

5. The tape now makes a half turn toward the outside and comes up and over the end guide.

6. The tape is then brought up and over to the left and wound over the top of the take-up reel.



1055 Paper Tape Punch

1810 DISK STORAGE UNIT

The readying and loading procedures follow.

PRE-CONDITIONS

System power ON, CARTRIDGE UNLOCKED lights ON.

LOADING PROCEDURE

1. Open the front door of the disk unit.
2. Grasp the handle of the access release mechanism of the drive to be loaded (drive 2 on top, 0 in the center, 1 on the bottom) and pull out and down.
3. Pick up the cartridge and, holding the cartridge with the IBM name toward you and on the left, insert the cartridge into the slot.
4. When the cartridge is positioned, raise the access release handle and lock the cartridge into place. If desired, load the other drives on the 1810 disk storage unit.
5. Press START on the desired drive.

The CARTRIDGE UNLOCKED lights will go out when the drives start to turn. When the drives come up to speed (approximately 90 seconds), the indicators showing the drive numbers will light, thus showing that the heads are loaded and the drives are ready.

1627 PLOTTER

The readying and loading procedures follow.

PRE-CONDITION

System power ON.

READYING PROCEDURE

1. If the pen is not in a raised position, turn the pen switch, first DOWN and then UP.
2. With the pen in the UP position, use the drum (X axis) and carriage (Y axis) controls to position the pen for the first plot. The 1627 plotter is now ready to be selected.

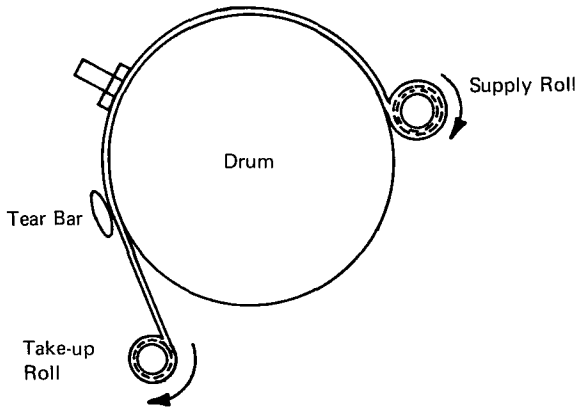
If you wish to load a new paper roll, use the following procedure:

LOADING PROCEDURE

1. Turn the 1627 power switch OFF (1627 power on indicator lamp out).
2. Remove the pen assembly, if installed, by loosening the knurled knob at the bottom of the pen holder and lifting the assembly out of the carriage.

CAUTION: Use care when handling the pen assembly; it is manufactured to close tolerances for optimum performance.

3. Rotate the right rear chart spool by hand until the drive key is pointing upward.
4. Hold the new roll of chart paper so that the key slot in the core is pointing upward. Place the roll against the spring-loaded left rear idler spool and force the spool to the left.
5. Lower the paper roll into the paper well and slide the right end onto the drive spool. Make certain the drive key engages the key slot in the core. The paper should feed out from under the roll and over the drum (see illustration).



1627 Loading Scheme

6. Install a paper roll core on the front spool below the drum, in the same manner as with the paper roll.

7. Pull a short length of paper off the roll, slide the end under the carriage rods, under the tear bar, and behind the core, and fasten it to the front side of the core with two or three short pieces of cellophane tape. Wind one or two turns of paper onto the core. Make certain the drum sprockets are properly meshed with the sprocket holes on both sides of the paper.

8. Reinstall the pen assembly in the carriage.

9. Turn the 1627 power switch to ON. The 1627 power on indicator will come on.

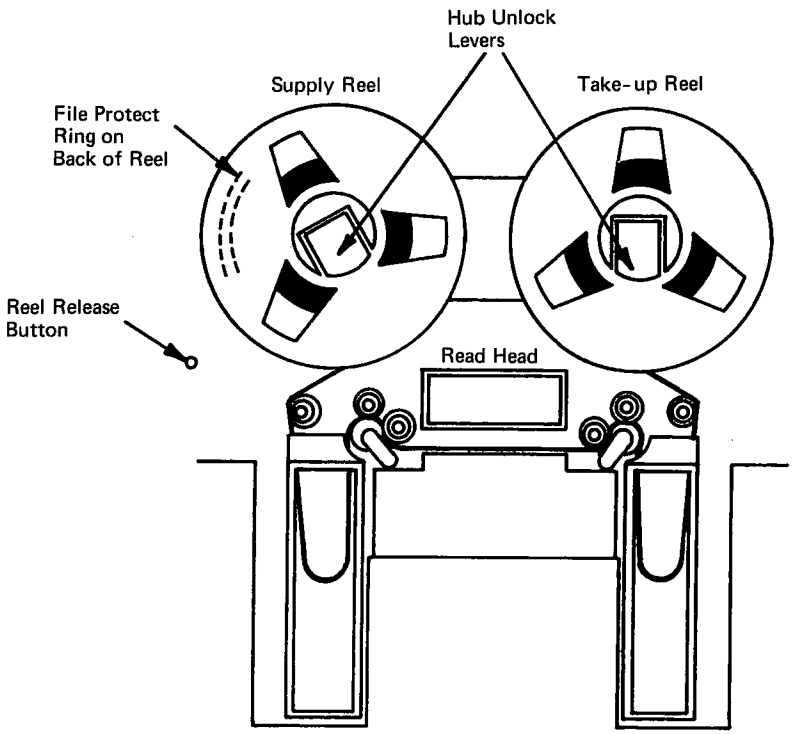
Note: The pen is down when the power is off; therefore, the pen assembly should be installed with the carriage over an area outside the "recording area." If the pen does not raise when power is turned on, turn the pen switch to DOWN, then to UP.

2401 MAGNETIC TAPE UNIT

The readying, loading, and unloading procedures follow.

PRE-CONDITIONS

System power ON.



2401 Magnetic Tape Unit

READYING PROCEDURE

1. Press LOAD REWIND. The read head will lower, the capstan drives will engage, the tape will be lowered into the vacuum columns, and the tape will rewind to load point.

2. Press START. The tape READY light will turn on. The tape drive is now ready to be selected.

LOADING PROCEDURE

1. Open the front door of the magnetic tape unit.
2. Unlock the takeup and supply reel hubs (see illustration).
3. Place the reel of tape to be read on the supply reel (left side). The end of the tape should hang down from the right side of the reel. If file protection is desired (read from this tape only), ensure that there is no file protect ring on the back of the tape. The groove around the back of the reel should be empty. If you desire to read or write with this tape, ensure that the file protect ring is in place. The FILE PROTECT light will go out when the unit is ready.
4. Lock the supply reel hub.
5. Place an empty tape reel on the takeup reel (right side).
6. Lock the takeup reel hub.
7. Press the reel release button (lower left of supply reel) and unwind about four feet of tape.
8. Thread the tape according to the illustration and, holding the end of the tape against the center of the takeup reel with one finger, wind the tape onto the takeup reel. The reel release button must be held down or the tape will not move. If this reel of tape has been used before, watch for the load point reflective marker on the outer edge (the side nearest you) of the tape. The load point marker must be to the right (takeup reel side) of the read head before the tape drive can be successfully loaded. If this is a new reel of tape, wind the tape until it is securely held on the takeup reel and attach a load point marker to the front edge of the tape.
9. Close the front door of the magnetic tape unit.

UNLOADING PROCEDURE

1. Press RESET (the unit cannot be unloaded while the unit is ready).
2. Press UNLOAD.

If you want to reload a tape you have just unloaded, manually reposition the load point marker to the right of the read head. If you attempt to load the tape with the load point marker to the left of the read head, the tape will unwind from the takeup reel.

1443 PRINTER

Readying and loading procedures follow:

PRE-CONDITIONS

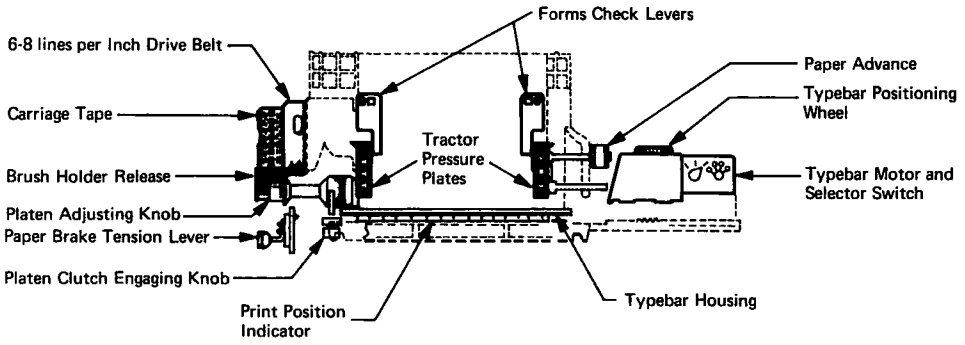
System power ON, ribbon in place, typebar in place, typebar motor switch ON.

READYING PROCEDURE

Press printer START.

LOADING FORMS INTO THE PRINTER

1. Raise the top cover. The release lever is at the bottom front center.
2. Disengage the platen clutch (see illustration for location of referenced carriage controls).
3. Raise the forms check levers. The FORM CHECK light will light.
4. Open the left and right tractor pressure plates.
5. Tip the print position indicator bar back toward you.
6. Set the paper brake tension lever below zero.
7. Turn the platen adjusting knob all the way to the rear (counterclockwise).
8. Feed the paper up and in from the left and lay it back across the carriage.
9. Line up the holes in the paper with the tractor pins and close the tractor pressure plates.
10. Lower the forms check lever. The FORM CHECK light will go out.
11. Set the paper brake tension lever (1 is a good setting for single-part paper).
12. Set the platen adjusting knob (2 is a good setting for single-part paper).
13. Turn the paper advance knob (right side of carriage) until a crease in the paper appears just above the typebar.
14. Close the print position indicator.



LOADING THE CARRIAGE CONTROL TAPE

1. Release the carriage brush holder by lifting up on the unlocking lever.
2. Insert the tape (channel one to the right) and close the brush holder.
3. Press CARRIAGE RESTORE on the operator's panel.
4. Engage the platen clutch.
5. Close the printer top cover.

CHANGING A TYPEBAR

Removing a Typebar

1. Press the 1433 STOP key.
2. Raise the printer top cover.
3. Turn the typebar motor switch (see illustration) to OFF. Allow approximately twenty seconds for the typebar motor to stop.
4. Turn the typebar motor switch to TYPEBAR REMOVAL.
5. Turn the typebar positioning wheel to the right until the end of the typebar shaft appears in the slot in the typebar housing. This slot is directly in front of the typebar motor switch.
6. Holding the shaft lightly in your right hand, exert a gentle pull to the right while continuing to turn the typebar positioning wheel.
7. When the typebar is released draw it straight out to the right.

Inserting a Typebar

1. Holding the typebar so the flag is on the right (typeface toward the machine), guide the typebar straight into its slot in the typebar housing.
2. When the typebar comes to a stop, start turning the typebar positioning wheel to the left until the drive teeth engage the typebar.
3. Position the typebar flag in accordance with the decal on top of the housing.
4. Turn the typebar motor switch ON.
5. Close the printer top cover.
6. Press the 1443 RESET key and then the 1443 START key.
7. A sync check will occur on the first line of print. Press 1443 RESET and START to continue.

2311 DISK STORAGE

IBM 2311 disk storage comprises two major components, the 2311 disk storage drive and the 1316 disk pack.

OPERATOR CONTROLS AND INDICATORS

Start/Stop Switch

Turning the Start/Stop Switch to the Stop position causes the access mechanism to retract from the disk pack and removes power from the disk drive motor. Automatic braking stops disk pack rotation in a few seconds.

Select Lock Indicator

When on, this indicates a machine condition which requires the attention of IBM Customer Engineering. This condition causes the disk storage drive to be disabled and stops the usage meter.

Enable/Disable Switch

When the CPU is stopped, this switch enables or disables the communication of the drive with the CPU. It also enables or disables the equipment usage meter.

If the CPU is running when the switch setting is changed, the drive and usage meter operating status are not changed until the CPU is stopped. (See also "Select Lock Indicator".)

OPERATING PROCEDURES

The following procedures should be followed for rapid, effective disk pack changing:

Loading and Readying Procedure.

- Open the 2311 cover.
- Remove the bottom disk pack cover by turning the bottom locking knob.
- Place the 1316 disk pack (still contained in top cover) on the 2311 spindle.
- Turn the top cover in clockwise direction until firm resistance is met.
- Lift the top cover from the disk pack.
- Close the 2311 cover.
- Set START/STOP switch to START.
- Set ENABLE/DISABLE switch to ENABLE.
- Reassemble the top and bottom covers of the disk pack.
- Store the covers in a clean cabinet or on a clean shelf.
- Turn on the 2841 control unit switch.

CAUTION: Do not leave disk pack top cover inside disk drive.

UNLOADING PROCEDURE

- Set START/STOP switch to STOP.
- Set ENABLE/DISABLE switch to DISABLE.
- Wait for the disk pack to stop rotating.
- Separate the top and bottom disk pack covers.
- Open the 2311 cover.
- Place the disk pack top cover over disk pack.
- Turn the top cover in direction of OFF arrow at least two full turns.
- Lift the top cover, now containing the disk pack, from the spindle.
- Fasten the bottom cover to disk pack (firmly).
- Close the 2311 cover.
- Store the disk pack in a clean cabinet or on a clean shelf.

2790 I/O DEVICES

A comprehensive discussion of the IBM 2791 and 2793 Area Stations, 2795 and 2796 Data Entry Units, and 1035 Badge Reader can be found in IBM 2790 Data Communication System, Component Description, Order Number GA27-3015. The programs that support these I/O devices are discussed in IBM 1800/2790 MPX Data Communication System, Order Number GC26-3732.

Console Operations

Many functions are done at the console by the operator. They are described below.

CLEAR MAIN STORAGE

To clear main storage to zeros:

1. Set the WRITE STOR PROT BITS switch to YES.
2. Set all other console switches OFF.
3. Set the console mode switch to RUN.
4. Press and hold CLEAR STOR.
5. While holding CLEAR STOR, press console START.
6. Release these keys and allow time for main storage to completely cycle.
7. Press console STOP.
8. Press RESET.

To clear storage protect bits only:

1. Set the WRITE STOR PROT BITS switch ON.
2. Set the other console switches OFF.
3. Set the console mode switch to DISPLAY.
4. Press and hold CLEAR STOR.
5. While holding CLEAR STOR, press console START.
6. Release these keys and allow time for main storage to completely cycle.
7. Press console STOP.
8. Return the mode switch to RUN.

Note: If a console check light is on after clearing main storage, reset and repeat the operation.

CONSOLE PROGRAM LOAD (NOT UNDER MPX)

1. Press consoles STOP and RESET.
2. Set the console mode switch to RUN.
3. Place the deck to be loaded into the card read punch hopper and press reader START
4. Press console PROGRAM LOAD.

For a means of loading a program to a location other than zero, see "1442 Relocatable Card Dump Program."

ALTERING OR DISPLAYING THE CONTENTS OF MAIN-STORAGE LOCATIONS

1. With the system stopped, set the console mode switch to LOAD.
2. Set the data switches to the desired four-character hexadecimal main-storage address. Switches 0-3 constitute the first hexadecimal character, 4-7 the second, etc.
3. Press LOAD I (the selected address is displayed in the I-register).

To display the contents of the address:

1. Set the console mode switch to DISPLAY.
2. Press console START.

The contents of the selected location is displayed in the B-register. Successive pressing on the console START key displays consecutive main-storage locations.

To alter the contents of the address:

1. Set the new data word in the console data switches.
2. Set the console mode switch to LOAD.
3. Press console START.

To return to system operation:

1. Set the console mode switch to RUN.
2. Press console START.

OPERATIONS MONITOR

The Operations Monitor comprises an internal program resettable timer and manual controls on the 1800 console. An ALARM indicator on the console will light if the Operations Monitor is in operation and the timer times out. The timer is reset by a CALL OPMON within the prescribed time interval. Once the alarm is activated, it must be manually reset. An audible alarm may also be attached to the Operations Monitor.

PROCEDURE

1. Set the selector switch located on the CE panel below the console to the desired time interval. (Ensure that the call to the Operations Monitor subroutine in the MPX system occurs often enough to prevent the timer from timing out.)
2. With the system to be monitored in operation, turn on the operations Monitor toggle switch on the console. This action initiates the first timer cycle and the system is now being monitored by the Operations Monitor.

If the alarm is activated (program in loop, power failure, etc.), it cannot be reset by the program. You must manually reset it by turning off the Operations Monitor toggle switch on the console.

MPX System Generation

Introduction

System generation is the process of setting up any one of a wide variety of possible MPX systems from materials supplied by IBM Program Information Department (PID).

A system generation must be done whenever a system is to be set up for the first time or whenever the capabilities of a system are to be changed, for example, when a system is to be expanded to support new input/output devices.

Generating Your System

The process of generating your MPX system is accomplished in five major phases:

During phase 1 you will begin the generation by performing a cold start to the disk-resident System Generation Monitor (a small IBM-supplied version of BOM). Cold Start is an MPX disk-resident program whose major functions are to load disk-resident executive programs into main storage and to pass control to them (see "Cold Start"). Once in control, the System-Generation Monitor will cause approximately 2500 blank cards to be punched and passed through the 1442 card read punch. The resultant deck will contain control statements and blank cards necessary to continue the system generation.

During phase 2 you will run the cards punched and passed during phase 1 back through the 1442 card read punch. Sets of cards, called equate cards, will be punched for BOM, the Executive Director, and various Subroutine Library programs. The equate cards are used to define the configuration of your MPX system to be generated. For details, a programmer should refer to the IBM 1800 Multiprogramming Executive Operating System Programmer's Guide, Order Number GC26-3720. Information specifying the various configuration options must be punched into these cards. You should know the values you wish punched into these cards before you begin generating your system. After the equate cards are completed, BOM, the Exeuctive Director, and certain library programs are assembled by the Macro Assembler. Finally, you will define your disk packs and delete certain programs from the PID system generation pack.

During phase 3 you will transfer control from the System Generation Monitor to your BOM which was assembled and punched to cards during phase 2. Under control of your BOM, the System Loader program (punched during phase 2) will be read into main storage. One of the main functions of the System Loader is to process assignment cards, cards used to specify the I/O device and machine function assignments to interrupt levels and bit positions on those levels. For a description of the assignment cards, see Appendix A. During this phase your assignment cards will be processed by the System Loader.

During phase 4 the following batch-processing operations are performed under the control of the Batch-Processing Monitor Supervisor, a disk-resident system program called into execution by your BOM when a // JOB control statement is encountered (see "Background-Processing Operations").

1. Definition of the number of disk drives on your system
2. Labeling of cartridges
3. Definition of main-storage layout
4. Definition of 1810 disk layout

During phase 5, system generation is completed. The previous phase is finished with the Batch-Processing Monitor Supervisor in control. You will now go on to generate a batch-processing system (build BOM) or to generate a real-time system (build the Executive and coreloads).

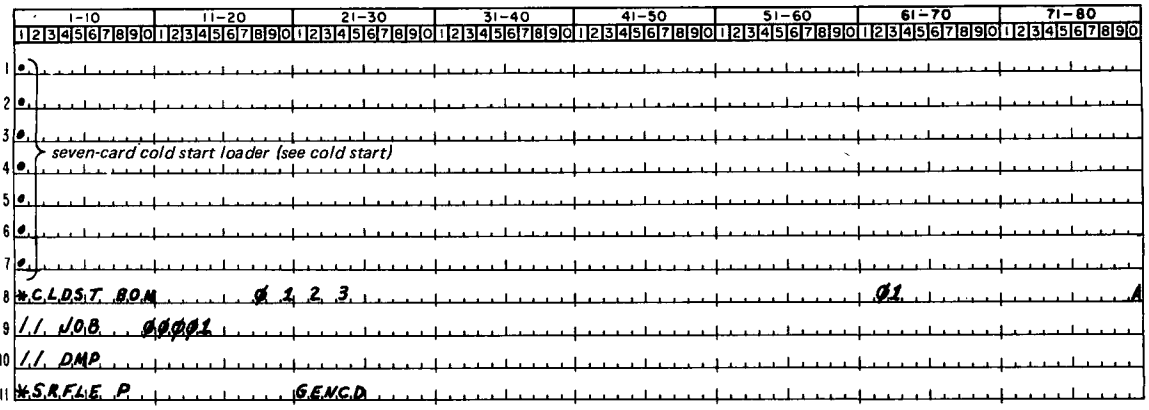
Note that during all phases of system generation you will be aided by system generation messages and system control messages printed on the list and system printers, respectively.

Materials

In order that you may perform a system generation, the IBM Program Information Department provides the following materials:

CARDS

- An 11-card starter deck. This deck consists of seven cold start loader cards used to load the disk-resident Cold Start program to main storage, a cold start name card used by Cold Start to load and pass control to the System-Generation Monitor (see Cold Start) and three Batch-Processing Monitor control cards used to perform the first phase of system generation.

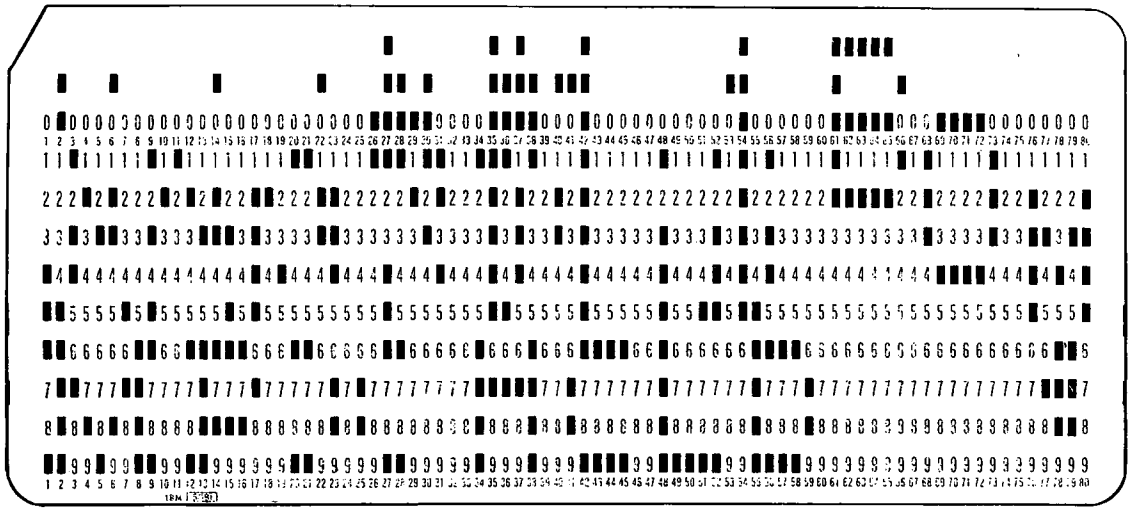


- A 4-card Basic Operating Monitor (BOM) high core loader. These cards are placed in front of your assembled BOM object deck (phase 3 of system generation). They cause that object deck to be read into main storage and control to be passed to it (see Figure 1).
- An 8-card 1442 relocatable main-storage dump to cards program. It is used as a debugging aid (see "MPX Debugging Aids").

DISKS

- One 1316 disk pack, if your system programs are to reside on a 2311 disk storage drive. The label of your 1316 disk storage pack, sent to you from PID, is PID001.
- Two 2315 disk cartridges, if your system programs are to reside on an 1810 disk storage unit and your system does not include a communications adapter.
- Three 2315 disk cartridges, if your system programs are to reside on an 1810 disk storage unit and your system is to contain one or more communications adapters.

The disk packs and cartridges issued by PID contain all the system programs and data files necessary to generate any MPX system.



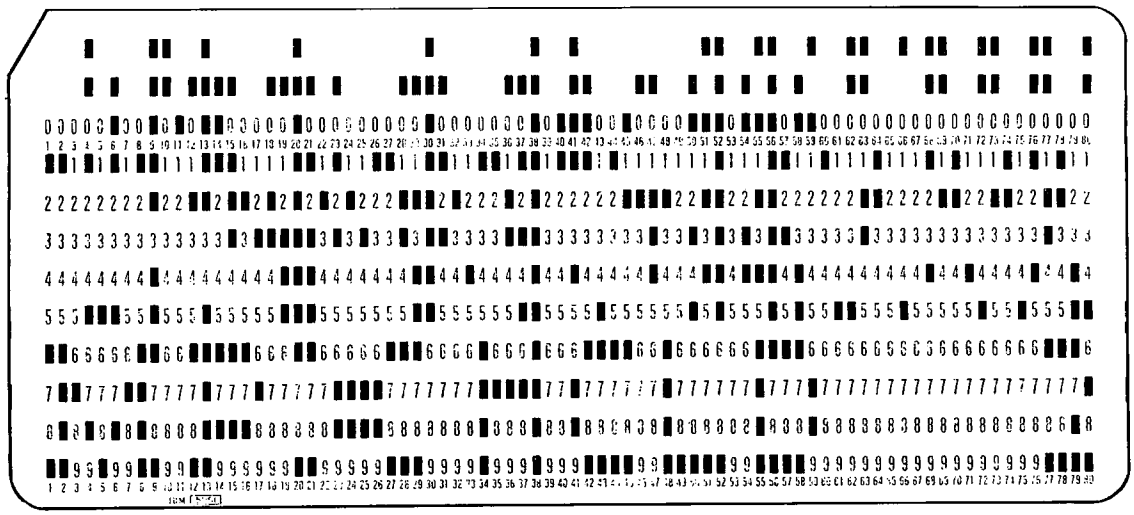
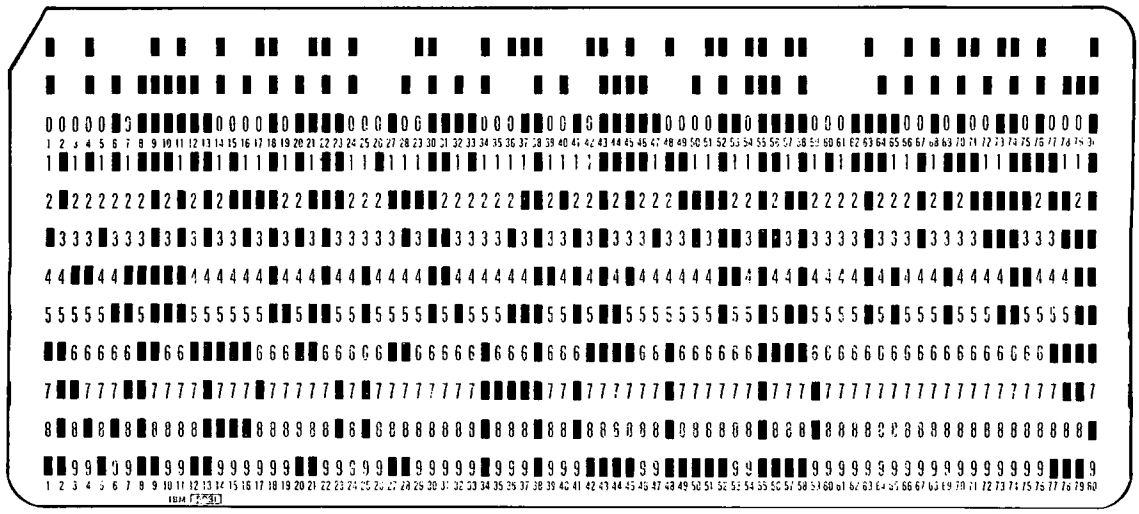


Figure 1. BOM High Core Loader Cards



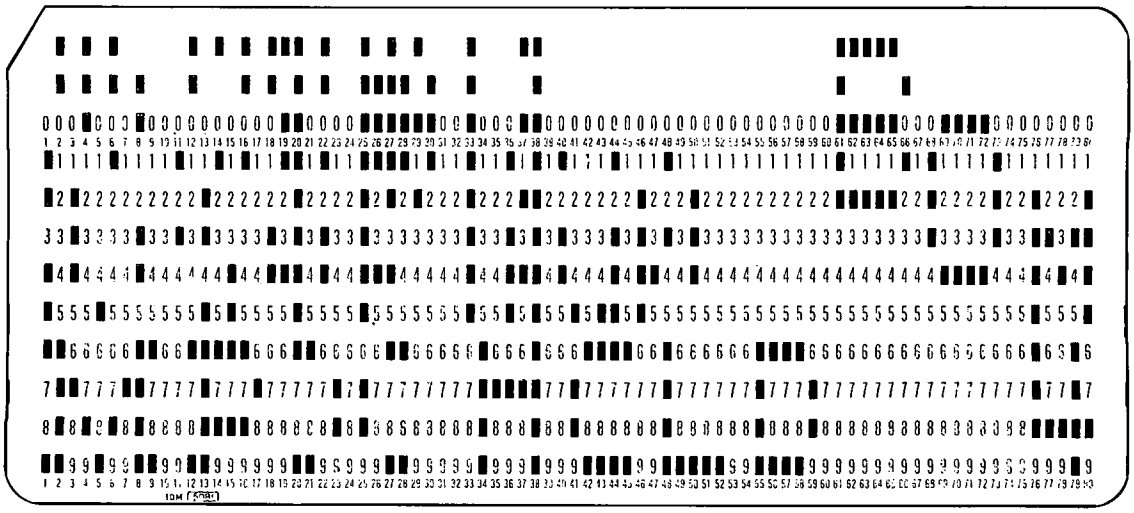


Figure 1. BOM High Core Loader Cards (Cont.)

Materials other than those provided by PID are required to generate a system. You must provide these:

CARDS

- 2,600 blank cards (more if you have a single-1810 drive system). These cards will be processed during phase 1 and will become the system-generation input for the other phases. Note that the actual number of cards used will vary between 2,500 and 2,600 depending on which system-generation options are selected.

- Note also that in phases 2 and 3 you will be required to complete equate cards and assignment cards. It is suggested that you know the values you wish punched in these cards prior to beginning the system-generation process.

SCRATCH DISKS

One scratch disk is required for each disk received from PID.

Operating Procedures

These are two sets of operating procedures for generating your system:

1. Those for generating a system using 2311(s) or two or more 1810 drives, and
2. Those for generating a system using a single 1810 drive.

For generating a system using a single 1810 drive, there is no phase 1 operation, so read the "System Generation Notes" below and skip to the section headed "Phase 2 Operations for Single 1810 Drive System."

For generating a system using 2311(s) or two or more 1810 drives, begin at the section headed "Phase 1 Operations."

SYSTEM GENERATION NOTES

1. The successful completion of each DMP operation is indicated by the printing of the message DMP FUNCTION COMPLETED. In the event the operation is unsuccessful, one or more DMP (DXX) error messages may be printed followed by the message DMP FUNCTION ABORTED. See IBM 1800 Multiprogramming Executive Operating System Error Message and Recovery Procedures, Order Number GC26-3727, for the meaning of the message and the procedure to correct the error. When the error has been corrected you may have to revert to a previous step to continue the generation process.

2. Setting program switch 6 ON causes the system to come to a WAIT on any EAC error. See the Error Messages manual for the meaning of the errors and the recovery procedures.

3. Do not stop the system-generating process by pressing the 1442 card punch STOP key. The System Generation Monitor does not contain no-response check subroutines. Pressing the STOP key may cause an interrupt to be lost, thereby placing the system in an interminable loop.

4. System generation may be stopped by pressing console STOP, and later restarted, at any point in the process. After stopping, make certain that all remaining input is saved. Cold start to PID pack 1 or load BOM from cards when restarting the system.

PHASE 1 OPERATIONS

1. Place PID disk 1 (2315 or 1316) on drive 0.
2. Ready drive 0.
3. Set the WRITE STOR PROT BITS switch ON (all other console switches should be OFF).
4. Set the console mode switch to RUN.
5. Clear main storage to zeros (refer to "I/O Device and Console Operations").
6. Place the 11-card starter deck received from PID in the 1442 card read punch.

   Note: The cold-start name card must be changed if your 2841 address is not A.
7. Fill the remaining space in the hopper with blank cards (approximately 2,600 cards will be needed during this phase).
8. Ready the card read punch.
9. Set the data switches as follows:

   If this system generation uses a 2311 disk storage drive, set:
   switch 0 : ON
   switches 1-3 : OFF
   switches 4-7 : OFF
   switches 8-11 : selector channel address (hexadecimal)
   switches 12-15 : OFF

   If this system generation does not use a 2311, ensure that all data switches are OFF.
10. Press console PROGRAM LOAD.
11. Ignore any cold start warning messages. Press console START to continue.

    Note: Cold Start can be performed only on PID disk 1.
12. The following messages are printed on the system printer:

```
IBM 1800 DACS  MPX/SYS GEN MON   00.000
SEN SW 0 ON ABSOLUTE LOADER
SEN SW 1 ON LOAD BP MONITOR
```

13. Set sense switch 1 ON.
14. Press console START.
15. The following messages are printed on the system printer and the blank cards are processed through the 1442 card read punch.

```
// JOB   00001   00 JAN 00 00.000 HRS
// DMP   00 JAN 00 00.000 HRS
*SRFLE P            GENCD              PUNCHING SYS GEN CONTROL CARDS
```

The control cards required to generate MPX are punched. These cards will be the input for phase two of the system-generation process. Note that some cards are processed without being punched. Keep removing cards from the stacker, placing them into a container in the order in which they are punched. When the process is complete, the number of cards processed is indicated by a message.

```
002520 CARDS PROCESSED
DMP FUNCTION COMPLETED
```

Blank cards remaining in the hopper are passed to the second stacker without processing. Ignore any card reader not ready message.

16. Remove the 11-card starter deck from the front of the processed output.

This completes phase 1. If you are generating a system using 2311 drive(s) or two or more 1810 drives, skip to the section headed "Phase 2 Operations for 2311 or Two or More 1810 Drive Systems."

PHASE 2 OPERATIONS FOR SINGLE 1810 DRIVE SYSTEM

The following is the operating procedure for generating a system using a single 1810 drive:

1. Prepare the following deck (comments cards are optional):

```
// JOB   00001
// *
// * PRESS START IF SYSTEM HAS A 1443 PRINTER
// *
// * PRESS CONSOLE INTERRUPT WITH SENSE SWITCH 7 UP IF SYSTEM
// * DOES NOT HAVE A 1443 PRINTER
// *
// PAUSE
// *
// * ALL PRINTING WILL BE DONE ON 1443 FROM NOW ON
// *
// SET A 0 0
// JOB   00001
// *
// *  THIS PASS GENERATES THE MPX SYSTEM. FOLLOW ALL PRINTED
// *  INSTRUCTIONS.
// *
// *  A DUMP OF BOM, EXDIR, AND LIBRARY EQUATE CARDS FOLLOWS.
// *  THE USER MUST PUNCH
// *  AN ENTRY IN THE OPERAND FIELD BEFORE RELOADING THESE CARDS.
// *  A SRFLE END CARD MUST BE INCLUDED WITH EQU'S.
// ********************************************************************
// DMP
*SRFLE P            EQUEX              PUNCHING EXDIR EQU CARDS *******
```

| (insert 130 blank cards)

```
// PAUSE       REMOVE AND DEFINE 'EXDIR' EQU CARDS
// DMP
*SRFLE P            EQUBM            PUNCHING BOM EQU CARDS************
```

| (insert 220 blank cards)

```
 / PAUSE       REMOVE AND DEFINE 'BOM' EOU CARDS
// DMP
*SRFLE P            EQULB              PUNCHING LIBRARY EQU CARDS****
```

(insert 75 blank cards)

```
// *  THE EQUATE CARDS FOR VARIOUS SUBROUTINES ARE SEPARATED FROM EACH OTHER
// *  BY CARDS CONTAINING ONLY ASTERISKS, SEE OPERATING PROCEDURES, SYSTEM
// *  GENERATION , EQUATE CARDS.
// *
// PAUSE       REMOVE AND DEFINE 'LIBR'EQU CARDS
// JOB   00001
// *
// *  A DUMP OF SYSTEM LOADER AND BOM CARD UTILITIES FOLLOWS.  AT EACH
// *  PAUSE, REMOVE CONTROL CARDS FROM PUNCHED OUTPUT AND LABEL DECK.
// *  PRESS CONSOLE START TO CONTINUE.
// *
// DMP
*DUMP       UAO PN  SYSLD IS BEING PUNCHED*******************************
```

(insert 120 blank cards)

```
// PAUSE     REMOVE AND LABEL 'SYSTEM LOADER'
// DMP
*DUMP       UAO PN  BDWAP IS BEING PUNCHED*******************************
```

| (insert 45 blank cards)

```
// PAUSE      REMOVE AND LABEL 'BOM DISK WRITE ADDRESS PROGRAM'
// DMP
*DUMP        UAO PN  BDUPL IS BEING PUNCHED******************************
```

| (insert 50 blank cards)

```
// PAUSE      REMOVE AND LABEL 'BOM DISK DUPLICATION PROGRAM'
// DMP
*DUMP        UAO PN  BDPAT IS BEING PUNCHED******************************
```

| (insert 35 blank cards)

```
// PAUSE      REMOVE AND LABEL 'BOM DISK PATCH PROGRAM'
// DMP
*DUMP        UAO PN  BLIST IS BEING PUNCHED******************************
```

| (insert 30 blank cards)

```
// PAUSE      REMOVE AND LABEL 'BOM 80-80 LIST PROGRAM'
// DMP
*DUMP        UAO PN  BDCRL IS BEING PUNCHED******************************
```

| (insert 40 blank cards)

```
// PAUSE      REMOVE AND LABEL 'BOM DISK DUMP AND RELOAD PROGRAM'
// DMP
*DUMP        UAO PN  BDPIP IS BEING PUNCHED******************************
```

| (insert 90 blank cards)

```
// PAUSE      REMOVE AND LABEL 'BOM 2311 DISK PACK INITIALIZATION PROGRAM'
// DMP
*DUMP        UAO PN  BDUMP IS BEING PUNCHED ************************************
```

| (insert 25 blank cards)

```
// PAUSE      REMOVE AND LABEL ' BOM 1810 DISK DUMP TO CARDS '
// DMP
*DUMP        UAO PN  BRELD IS BEING PUNCHED ************************************
```

| (insert 35 blank cards)

```
// PAUSE      REMOVE AND LABEL ' BOM 1810 DISK RELOAD FROM CARDS'
// JOB   00001
// DMP
*SRFLE P             SAMP1 IS BEING PUNCHED**********************
```

| (insert 310 blank cards)

```
// PAUSE      REMOVE AND LABEL 'THE MPX SAMPLE PROGRAM'
// *
// *  NOW DUMP PID PACK 1 TO CARDS FOR BACKUP
// *
// DMP
*DUMP        DKO PN
```

(insert 8,500 blank cards)

```
// PAUSE  SAVE FOR BACKUP
// *
// *
// *  AT PAUSE PLACE USER COMPLETED LIBRARY EQUATE CARDS BEHIND RESPECTIVE
// *  *SRFLE M CONTROL CARDS FOR EXDCD,CSPCD,LSPCD,MDFCD,MFICD,ADRCD,DMPSC.
// *  PRESS READER AND CONSOLE START TO CONTINUE.
// *
// *   BE SURE TO INCLUDE A SRFLE END CARD AT THE END OF EACH
// *   SET OF EQUATES.
// *
// PAUSE
// JOB   00001
// DMP
*SRFLE M            EXDCD              EXECUTIVE EQUATE CARDS MUST FOLLOW
*SRFLE M            CSPCD              CSPAR EQUATE CARDS MUST FOLLOW
*SRFLE M            LSPCD              LSPCL EQUATE CARDS MUST FOLLOW
*SRFLE M            MDFCD              MDFIO EQUATE CARDS MUST FOLLOW
*SRFLE M            MFICD              MFIO  EQUATE CARDS MUST FOLLOW
*SRFLE M            ADRCD              ADRCK EQUATE CARDS MUST FOLLOW
*SRFLE M            DMPSC              DMPS  EQUATE CARDS MUST FOLLOW
// JOB   00001
// *  ASSEMBLE EXECUTIVE DIRECTOR, PUNCH THEN STORE.
// ASM EXDIR EXDCD
```

(insert 250 blank cards)

```
*STORE              EXDIR
// *
// *  THE FOLLOWING CONTROL CARDS ASSEMBLE AND STORE THE SUBROUTINES
// *  AFFECTED BY THE JUST-INSERTED LIBRARY EQUATE CARDS.
// *
// ASM ADRCK ADRCD
*STORE            0 ADRCK
// ASM DMPS  DMPSC
*STORE            0 DMPS
// ASM CSPAR CSPCD
*STORE            0 CSPAR
// ASM LSPCL LSPCD
*STORE            0 LSPCL
// ASM MDFIO MDFCD
*STORE            0 MDFIO
// ASM MFIO  MFICD
*STORE            0 MFIO
// JOB   00001
// *  THE FOLLOWING CONTROL CARDS WILL DELETE THE ENTIRE COMMUNICATION
// *  ADAPTER OBJECT LIBRARY.  IF THIS IS DESIRED, PRESS START.
// *  IF IT IS NOT DESIRED TO DELETE THE ENTIRE COMMUNICATION
// *  LIBRARY, THEN SET PROGRAM SWITCH 7, PRESS CONSOLE INTERRUPT
// *  TO CONTINUE.
// *
// PAUSE
// DMP
*DELET    R         BSCTR              CA TRACE ROUTINE
*DELET              BSCTR              CA TRACE ROUTINE
*DELET              BSCCK              I/O AREA CHECK ROUTINE
*DELET    R         ZIPCO              CONVERSION ROUTINE
*DELET              ZIPCO              CONVERSION ROUTINE
*DELET              USHOL              USASCII CODE TO HOLLERITH TBL
*DELET              HOLUS              HOLLERITH TO USASCII CODE TBL
*DELET              USEBC              USASCII TO EBCDIC TABLE
*DELET              EBCUS              EBCDIC TO USASCII TABLE
*DELET              USPRT              USASCII TO 1443 PRINTER TABLE
*DELET              USTYP              USASCII TO TYPEWRITER TABLE
*DELET    R         PEFIB              PACKED EBCDIC TO FIXED BINARY
*DELET              PEFIB              PACKED EBCDIC TO FIXED BINARY
*DELET    R         FLBPE              FLOATING BINARY TO PACKED EBCDIC
*DELET              FLBPE              FLOATING BINARY TO PACKED EBCDIC
// JOB   00001
// *
// * IF THE LIBRARY SUBROUTINES TO SUPPORT THE USE OF 2311'S ARE NOT
// * NEEDED PRESS START.  OTHERWISE, PRESS CONSOLE INTERRUPT WITH
// * SENSE SWITCH 7 ON.
```

This page intentionally left blank.

```
// *
// PAUSE
// DMP
*DELET    R          DSOR
*DELET    R          DSCR
*DELET    R          MDAIO
*DELET               DPIP
*DELET    R          RESER
*DELET    R          DSPRO
// JOB    00001
// *
// *  THE FOLLOWING SOURCE FILES AND UTILITIES ARE DELETED
// * DELETE-- EQUBM, EQUEX, EQULB, GENCD, CSPCD, LSPCD, MDFCD, MFICD, EXDCD
// * BLIST, BDWAP, SYSLD, BDUPL, BDPAT, BDPIP, BDCRL, ADRCD, DMPSC, SAMP1
// * IF ALL PRECEDING OPERATIONS COMPLETED SUCCESSFULLY,
// * PRESS CONSOLE START TO CONTINUE.
// *
// PAUSE      PRESS START TO CONTINUE
// DMP
*DELET    D          EQUBM
*DELET    D          EQUEX
*DELET    D          EQULB              LIBRARY EQUATES
*DELET    D          GENCD
*DELET    D          CSPCD              CSPAR SOURCE
*DELET    D          LSPCD              LSPCL SOURCE
*DELET    D          MDFCD              MDFIO SOURCE
*DELET    D          MFICD              MFIO SOURCE
*DELET    D          EXDCD              EXECUTIVE DIRECTOR SOURCE
*DELET               BLIST
*DELET               BDWAP
*DELET               SYSLD
*DELET               BDUPL
*DELET               BDPAT
*DELET               BDPIP
*DELET               BDCRL
*DELET               BDUMP
*DELET               BRELD
*DELET    D          ADRCD              ADRCK SOURCE
*DELET    D          DMPSC              DMPS SOURCE
*DELET    D          SAMP1              MPX SAMPLE PGM SOURCE
// JOB
// *
// * THE FOLLOWING CONTROL CARDS DELETE THE 2790 OBJECT LIBRARY
// *
// DMP
*DELET               CLIP
*DELET               CLTP
*DELET               FILES
*DELET    R          LACCN
*DELET               LINKF
*DELET    R          LOOP1
*DELET    R          LOOP2
*DELET    R          PULSE
*DELET               WRTLN
// JOB
// DMP
*DEFINE PAKDK 0
// JOB    00001
// *
// *  MOUNT PID PACK 2 ON DRIVE 0
// *  DO NOT COLD START.
// *
// PAUSE
// JOB    00002
// *
// *  NOW DUMP PID PACK 2 TO CARDS FOR BACKUP
// *
// DMP
*DUMP       DKO PN
```

(insert 5,000 blank cards)

```
// PAUSE  SAVE FOR BACKUP
// *
// *
// *  AT PAUSE PLACE USER-COMPLETED EQUATE CARDS BEHIND
// *  *SRFLE M CONTROL CARD FOR BOM
// *  PRESS READER AND CONSOLE START TO CONTINUE.
// *
// PAUSE
// *
// DMP
*SRFLE M              BOMCD
// *
// * NOW ASSEMBLE AND PUNCH BOM
// *
// ASM BOM    BOMCD
```

(insert 800 blank cards)

```
// *
// *  AT PAUSE MOUNT PID PACK 1 IF SYSTEM DOES NOT REQUIRE CA.
// *  MOUNT PID PACK 3 IF SYSTEM REQUIRES CA.
// *
// PAUSE
// *
// JOB    00003
// *
// * SINCE THE ONE DRIVE SYSTEM DOES NOT SUPPORT THE 2790 ADAPTER,
// * DELETE THE FOLLOWING FILE.
// *
// DMP
*DELET     D          X2790
// *
// *  NOW DUMP PID PACK 3 TO CARDS FOR BACKUP
// *
// DMP
*DUMP        DK0 PN
```

(insert 5,000 blank cards)

```
// PAUSE  SAVE FOR BACKUP
// *
// *
// *    AT PAUSE PLACE BSCIO SYSGEN EQUATES AFTER *SRFLE CONTROL CARD
// *    PRESS READER START AND CONSOLE START TO CONTINUE
// *
// PAUSE
// *
// DMP
*SRFLE M            BSCCD
// *
// *
// *  ASSEMBLE AND PUNCH BSCIO
// *
// ASM BSCIO BSCCD
```

(insert 75 blank cards)

```
// *  THE EQUATE CARDS FOR VARIOUS SUBROUTINES ARE SEPARATED FROM EACH OTHER
// *  BY CARDS CONTAINING ONLY ASTERISKS, SEE OPERATING PROCEDURES, SYSTEM
// *  GENERATION , EQUATE CARDS.
// *
// PAUSE       REMOVE AND DEFINE 'LIBR'EQU CARDS
// JOB   00001
// *
// *  A DUMP OF SYSTEM LOADER AND BOM CARD UTILITIES FOLLOWS.  AT EACH
// *  PAUSE, REMOVE CONTROL CARDS FROM PUNCHED OUTPUT AND LABEL DECK.
// *  PRESS CONSOLE START TO CONTINUE.
// *
// DMP
*DUMP       UAO PN  SYSLD IS BEING PUNCHED*******************************
```

(insert 120 blank cards)

```
// PAUSE     REMOVE AND LABEL 'SYSTEM LOADER'
// DMP
*DUMP       UAO PN  BDWAP IS BEING PUNCHED*******************************
```

(insert 45 blank cards)

```
// PAUSE     REMOVE AND LABEL 'BOM DISK WRITE ADDRESS PROGRAM'
// DMP
*DUMP       UAO PN  BDUPL IS BEING PUNCHED*******************************
```

(insert 50 blank cards)

```
// PAUSE     REMOVE AND LABEL 'BOM DISK DUPLICATION PROGRAM'
// DMP
*DUMP       UAO PN  BDPAT IS BEING PUNCHED*******************************
```

(insert 35 blank cards)

```
// PAUSE     REMOVE AND LABEL 'BOM DISK PATCH PROGRAM'
// DMP
*DUMP       UAO PN  BLIST IS BEING PUNCHED*******************************
```

This page intentionally left blank.

```
(insert 30 blank cards)
// PAUSE      REMOVE AND LABEL 'BOM 80-80 LIST PROGRAM'
// DMP
*DUMP        UAO PN  BDCRL IS BEING PUNCHED*****************************

(insert 40 blank cards)
// PAUSE      REMOVE AND LABEL 'BOM DISK DUMP AND RELOAD PROGRAM'
// DMP
*DUMP        UAO PN  BDPIP IS BEING PUNCHED*****************************

(insert 90 blank cards)
// PAUSE      REMOVE AND LABEL 'BOM 2311 DISK PACK INITIALIZATION PROGRAM'
// DMP
*DUMP        UAO PN  BDUMP IS BEING PUNCHED ************************************

(insert 25 blank cards)
// PAUSE      REMOVE AND LABEL ' BOM 1810 DISK DUMP TO CARDS '
// DMP
*DUMP        UAO PN  BRELD IS BEING PUNCHED ************************************

(insert 20 blank cards)
// *
// *  AT PAUSE MOUNT PID PACK 1
// *
// PAUSE
// *
// JOB   00001
// *
// *  CONSOLE INTERRUPT WITH PROGRAM SWITCH 7 ON IF NO CA.
// *  OTHERWISE , PRESS CONSOLE START.
// *
// PAUSE
// *
// *  PLACE BSCIO OBJECT DECK AFTER STORE CARD.
// *
// PAUSE
// *
// DMP
*STORE       RD    0 BSCIO
// JOB   00001
// END   AT THIS POINT REFER TO OP MANUAL FOR FURTHER INSTRUCTIONS
*END SRFLE                                                        99999999
```

This entire file will be input to phase 2 of the system-generation process for a single-1810 drive system.

2. Place PID disk 1 on drive 0.
3. Ready drive 0.
4. Set the WRITE STOR PROT BITS switch ON (all other console switches should be OFF).
5. Set the console mode switch to RUN.
6. Clear main storage to zeros (refer to "I/O Device and Console Operations").
7. Place the 7-card cold start loader and the cold start name card received from PID in the 1442 card read punch.
8. Fill the remaining space in the hopper with the system-generation input file that you punched in step 1.
9. Ready the card read punch.

10. Press console PROGRAM LOAD.

11. Ignore any cold start warning messages. Press console START to continue.

    Note: Cold Start can be performed only on PID disk 1.

12. The following messages are printed:

```
IBM 1800 DACS  MPX/SYS GEN MON   00.000
SEN SW 0 ON ABSOLUTE LOADER
SEN SW 1 ON LOAD BP MONITOR
```

13. Set sense switch 1 ON.

14. Press console START.

15. Execute steps 3 through 13 under the section headed "Phase 2 Operations for 2311 or Two or More 1810 Drive Systems" and return.

16. You will now dump PID disk 1 to cards to save it for a backup. You can reload it by using BRELD. Press console START to continue.

17. Execute steps 75 through 81 under the section headed "Phase 2 Operations for 2311 or Two or More 1810 Drive Systems" and return. Note that this procedure will differ slightly in that BOM equates are not placed on disk by SRFLE nor is BOM assembled.

18. Execute steps 85 through 87 under the section headed "Phase 2 Operations for 2311 or Two or More 1810 Drive Systems" and return.

19. You will now dump PID disk 2 to cards to save it for backup. Place PID disk 2 on drive 0.

20. Ready drive 0.

21. Press console START to continue (to dump PID disk to cards).

22. Place your completed BOM equate cards behind the *SRFLE card in the 1442.

23. Ready the 1442.

24. Press console START to continue. The BOM equate cards are now read to disk and BOM is assembled.

25. Remove and label the BOM object deck (just punched).

26. If your system does not require communications adapters, skip to step 33. If your system does require one or more communications adapters, mount PID disk 3 on drive 0.

27. Ready drive 0.

28. Press console START to continue. You will not dump PID disk 3 to cards to save for backup.

29. Place your completed BSCIO equate cards behind the *SRFLE card in the 1442.

30. Ready the 1442.

31. Press console START to continue. The BSCIO equate cards are now read to disk and BSCIO is assembled.

32. Remove and label the BSCIO object deck (just punched).

33. Place PID disk 1 on drive 0.

34. Ready drive 0.

35. Press console START.

36. If your system does not contain any communications adapters, set program switch 7 ON and press CONSOLE INTERRUPT. This completes phase 2 operations for a single-1810 drive system. To continue, go to the section headed "Phase 3 Operations." If your system does contain one or more communications adapters, press console START.

37. Place the BSCIO object deck punched in step 31 behind the *STORE card in the 1442.

38. Ready the 1442.

39. Press console START to continue. The BSCIO program is now stored to disk.

This completes phase 2 operations for a single-1810 drive system. To continue, go to the section headed "Phase 3 Operations."

PHASE 2 OPERATIONS FOR 2311 OR TWO OR MORE 1810 DRIVE SYSTEMS

1. Fill the 1442 card read punch with the cards processed during phase 1 operations.

2. Ready the 1442 card read punch.

3. The following messages are printed on the system printer:

```
// JOB   00001   00 JAN 00 00.000 HRS
// *
// * PRESS START IF SYSTEM HAS A 1443 PRINTER
// *
// * PRESS CONSOLE INTERRUPT WITH SENSE SWITCH 7 UP IF SYSTEM
// * DOES NOT HAVE A 1443 PRINTER
// *
// PAUSE   00 JAN 00 00.000 HRS
```

4a. If your system has a 1443 printer, ready it and press console START. The following messages are printed on the system printer:

```
CONTINUE
// *
// * ALL PRINTING WILL BE DONE ON 1443 FROM NOW ON
// *
// SET A 0 0   00 JAN 00 00.000 HRS
    IN ST  LIST  SYS
OLD    A     1     1
NEW    A     0     0
```

and on the 1443 printer:

```
// SET A 0 0   00 JAN 00 00.000 HRS
    IN ST  LIST  SYS
OLD    A     1     1
NEW    A     0     0
```

Go to step 5.

4b. If your system does not have a 1443 printer, set sense switch 7 ON and press CONSOLE INTERRUPT.

5. The following messages are printed and the equate cards for the Executive Director are punched. Some of the following //JOB cards may be different if your system uses 1810 drives.

```
// JOB   0000100002 00 JAN 00 00.000 HRS
// *
// *  THIS PASS GENERATES THE MPX SYSTEM. FOLLOW ALL PRINTED
// *  INSTRUCTIONS.
// *
// *  A DUMP OF BOM, EXDIR, AND LIBRARY EQUATE CARDS FOLLOWS.
// *  THE USER MUST PUNCH
// *  AN ENTRY IN THE OPERAND FIELD BEFORE RELOADING THESE CARDS.
// *  A SRFLE END CARD MUST BE INCLUDED WITH EQU'S
// ***************************************************************
// DMP   00 JAN 00 00.000 HRS
*SRFLE P            EQUEX            PUNCHING EXDIR EQU CARDS *******
000122 CARDS PROCESSED
DMP FUNCTION COMPLETED
// PAUSE       REMOVE AND LABEL 'EXDIR' EQU CARDS   00 JAN 00 00.000 HRS
```

6. When the program pauses, remove and label the Executive Director equate cards. Note that control cards precede the equate cards (cards with no sequence numbers in columns 73-80) in the stacker. Disregard these cards as they are no longer needed.

7. Press console START to continue. The following messages are printed and the equate cards for BOM are punched.

```
CONTINUE
// DMP   00 JAN 00 00.000 HRS
*SRFLE P            EQUBM            PUNCHING BOM EQU CARDS************
000217 CARDS PROCESSED
DMP FUNCTION COMPLETED
// PAUSE       REMOVE AND LABEL 'BOM' EQU CARDS   00 JAN 00 00.000 HRS
```

8. When the program pauses, remove and label the BOM equate cards. Omit control cards.

9. Press console START to continue. The following messages are printed and equate cards for various Subroutine Library programs are punched.

```
CONTINUE
// DMP   00 JAN 00 00.000 HRS
*SRFLE P            EQULB            PUNCHING LIBRARY EQU CARDS****
000031 CARDS PROCESSED
DMP FUNCTION COMPLETED
// *  THE EQUATE CARDS FOR VARIOUS SUBROUTINES ARE SEPARATED FROM EACH OTHER
// *  BY CARDS CONTAINING ONLY ASTERISKS, SEE OPERATING PROCEDURES, SYSTEM
// *  GENERATION, EQUATE CARDS
// PAUSE       REMOVE AND LABEL  'LIBR'EQU CARDS   00 JAN 00 00.000 HRS
```

10. When the program pauses, remove and label the deck of LIBR equate cards. Omit the control cards which precede and follow the equate cards in the stacker. Cards containing only asterisks separate the equate cards for the various subroutines. Replace these cards with cards containing 9s in the last eight columns.

11. Press console START to continue. The following messages are printed and the System Loader is punched to cards.

```
CONTINUE
// JOB   00001      00 JAN 00 00.000 HRS
// *
// *  A DUMP OF SYSTEM LOADER AND BOM CARD UTILITIES FOLLOWS. AT EACH
// *  PAUSE, REMOVE CONTROL CARDS FROM PUNCHED OUTPUT AND LABEL DECK.
// *  PRESS CONSOLE START TO CONTINUE.
// *
// DMP   00 JAN 00 00.000 HRS
*DUMP       UAO PN  SYSLD IS BEING PUNCHED****************************
DMP FUNCTION COMPLETED
// PAUSE     REMOVE AND LABEL 'SYSTEM LOADER'   00 JAN 00 00.000 HRS
```

12. When the program pauses, remove and label the System Loader deck. Omit all control cards.

13. Press console START to continue. The following sets of messages will be printed and card decks punched. At each pause between the decks, remove control cards from the punched output and label the decks. Press console START after each to continue.

```
CONTINUE
// DMP    00 JAN 00 00.000 HRS
*DUMP        UAO PN  BDWAP IS BEING PUNCHED*****************************
DMP FUNCTION COMPLETED
// PAUSE      REMOVE AND LABEL 'BOM DISK WRITE ADDRESS PROGRAM'
CONTINUE
// DMP    00 JAN 00 00.000 HRS
*DUMP        UAO PN  BDUPL IS BEING PUNCHED*****************************
DMP FUNCTION COMPLETED
// PAUSE      REMOVE AND LABEL 'BOM DISK DUPLICATION PROGRAM'
CONTINUE
// DMP    00 JAN 00 00.000 HRS
*DUMP        UAO PN  BDPAT IS BEING PUNCHED*****************************
DMP FUNCTION COMPLETED
// PAUSE      REMOVE AND LABEL 'BOM DISK PATCH PROGRAM'   00 JAN 00 00.000 HRS
CONTINUE
// DMP    00 JAN 00 00.000 HRS
*DUMP        UAO PN  BLIST IS BEING PUNCHED*****************************
DMP FUNCTION COMPLETED
// PAUSE      REMOVE AND LABEL 'BOM 80-80 LIST PROGRAM'   00 JAN 00 00.000 HRS
CONTINUE
// DMP    00 JAN 00 00.000 HRS
*DUMP        UAO PN  BDCRL IS BEING PUNCHED*****************************
DMP FUNCTION COMPLETED
// PAUSE      REMOVE AND LABEL 'BOM DISK DUMP AND RELOAD PROGRAM'
CONTINUE
// DMP    00 JAN 00 00.000 HRS
*DUMP        UAO PN  BDPIP IS BEING PUNCHED*****************************
DMP FUNCTION COMPLETED
// PAUSE      REMOVE AND LABEL 'BOM 2311 DISK INITIALIZATION PROGRAM'
CONTINUE
// DMP    00 JAN 00 00.000 HRS
*DUMP        UAO PN  BDUMP IS BEING PUNCHED*****************************************
DMP FUNCTION COMPLETED
// PAUSE      REMOVE AND LABEL ' BOM 1810 DISK DUMP TO CARDS '
CONTINUE
// DMP    00 JAN 00 00.000 HRS
*DUMP        UAO PN  BRELD IS BEING PUNCHED*****************************************
DMP FUNCTION COMPLETED
// PAUSE      REMOVE AND LABEL ' BOM 1810 DISK RELOAD FROM CARDS'
CONTINUE
// JOB    00001      00 JAN 00 00.000 HRS
// DMP    00 JAN 00 00.000 HRS
*SRFLE P             SAMP1 IS BEING PUNCHED*****************************
000130 CARDS PROCESSED
DMP FUNCTION COMPLETED
// PAUSE      REMOVE AND LABEL 'THE MPX SAMPLE PROGRAM'   00 JAN 00 00.000 HRS
```

14. At this point the system-generation procedure differs depending on which types of disk units you have in your system.

    If your system uses physical 1810 drives for this system generation, the following messages are printed:

```
CONTINUE
// *
// *  PLACE SCRATCH CARTRIDGE ON DRIVE 1. ADDRESS SCRATCH CARTRIDGE
// *  USING JUST-PUNCHED BOM DISK WRITE ADDRESS PROGRAM BDWAP.
// *
// *  WHEN MPX/BOM MESSAGES ARE PRINTED-- NPRO READER, PLACE BDWAP DECK
// *  IN READER, FOLLOW WITH NPRO JOB CARD AND REMAINDER OF STACKED
// *  INPUT, READY READER, TURN ON SS 0, DATA SWITCH 15, AND PRESS
// *  CONSOLE START.
// *
// *  LEAVE DATA SWITCH 15 ON FOR ONE TRY. PRESS CONSOLE START.
// *
// *  LEAVE DATA SWITCH 15 ON FOR DRIVE 1. PRESS CONSOLE START.
// *
// *  AT COMPLETION OF BDWAP, TURN OFF CONSOLE SWITCHES, PRESS
// *  STOP, RESET AND START.  THEN SET PROGRAM SWITCH 7 ON, AND PRESS
// *  CONSOLE INTERRUPT TO CONTINUE.
// *
// END   00 JAN  00 00.000 HRS
```

If your system uses 2311 drives for this system generation, the following messages are printed:

```
CONTINUE
// *
// *
// *  IF THIS SYSTEM GENERATION USES A SINGLE 2311 DRIVE, IGNORE THE
// *  INSTRUCTIONS TO THE NEXT // END CARD.
// *
// *  PLACE A SCRATCH PACK ON DRIVE 1.  INITIALIZE IT USING THE JUST-PUNCHED
// *  BOM 2311 DISK PACK INITIALIZATION PROGRAM (BDPIP) AND COPY ONTO IT THE
// *  CONTENTS OF DRIVE 0 USING THE BOM DISK DUPLICATION PROGRAM (BDUPL),
// *  AS FOLLOWS
// *
// *  WHEN MPX/BOM MESSAGES ARE PRINTED-- NPRO READER, PLACE BDPIP DECK
// *  IN READER.  PLACE BDPIP CONTROL CARDS AFTER OBJECT DECK.  NEXT
// *  PLACE BDUPL , THE NPRO CARDS AND THE REMAINDER OF THE STACKED INPUT
// *  IN THE READER.  READY THE READER.  TURN ON SS 0, AND DATA SWITCH 15.
// *  PRESS CONSOLE START.  AS MESSAGES ARE PRINTED, YOU MAY
// *  PRESS START TO CONTINUE.
// *
// *  AT COMPLETION OF BDPIP, PRESS CONSOLE START WITH SS0 AND
// *  DATA SWITCH 15 ON.
// *
// *  TO COPY DRIVE 0 TO DRIVE 1, REFER TO OPERATING PROCEDURES MANUAL
// *  FORM C26-3725, BOM CARD UTILITIES, FOR DETAILS.
// *
// *  SAVE COPY OF PID PACK
// *
// *
// *  AT COMPLETION OF BDUPL, TURN OFF CONSOLE SWITCHES, PRESS
// *  STOP, RESET AND START.  THEN SET PROGRAM SWITCH 7 UP, AND PRESS
// *  CONSOLE INTERRUPT TO CONTINUE.
// *
// END   00 JAN 00 00.000 HRS
```

The following messages are then printed (in both cases) on the system printer:

```
IBM 1800 DACS  MPX/SYS GEN MON   00.000
SEN SW 0 ON ABSOLUTE LOADER
SEN SW 1 ON LOAD BP MONITOR
```

If your system generation uses 2315 disk cartridges, go to step 15 to continue.

If your system generation uses a 1316 disk pack and your system has more than one 2311 disk drive, skip to step 58 to continue.

If your system generation uses a 1316 disk pack and your system has only one 2311 disk drive, skip to step 46 to continue.

SYSTEM GENERATION USING 2315 DISK CARTRIDGES

15. Load your scratch cartridge to drive 1.

16. Ready drive 1.

17. Remove all cards from the 1442 hopper.

18. Nonprocess out all the cards inside the 1442. Omit all these cards except the last which is a // JOB card.

19. Place the BOM Disk Write Addresses Program, BDWAP (one of the BOM utilities punched in step 13), into the 1442 hopper.

20. Follow it with the // JOB card and the remainder of the system-generation input deck from steps 17 and 18.

21. Ready the 1442 card read punch.

22. Set sense switch 0 ON and data switch 15 ON. Press console START.

23. The following messages are printed:

```
BOM DISK WRITE ADDRESSES PROGRAM
ENTER NO  TRIES ON DATA SW MAX 001F
```

    Enter the number of tries in the data switches (/00 xx) (described under "BDWAP - BOM Disk Write Addresses Program") and press console START.

24. The following messages are printed:

```
 DATA SWITCHES EQUAL LOGICAL DRIVE
DRIVE CODES - HEX 0000 0001 0002
```

    Press console START.

25. The following message is printed:

```
DRIVE SELECTED IS  1.IF CORRECT, PRESS START WITH SENSE SWITCH 0 ON
```

    Press console START.

26. The following messages should be printed:

```
 THERE ARE NO DEFECTIVE CYLINDERS
BOM DISK WRITE ADDRESSES PROGRAM
ENTER NO. TRIES ON DATA SW MAX 001F
```

    If a message indicating that there are defective cylinders is printed, refer to "BOM Card Utilities," BOM Disk Write Address Program (BDWAP). If the cartridge is defective, replace the cartridge and return to step 15.

27. Set all data switches OFF.

28. Press console STOP, RESET and START.

29. Set program switch 7 ON and press CONSOLE INTERRUPT. The following messages will be printed:

```
// JOB   00001   00 JAN 00 00.000 HRS
// DMP   00 JAN 00 00.000 HRS
*DCOPY 0 1
DMP FUNCTION COMPLETED
// *
// * SAVE PID PACK 1, REMOVE COPY OF PID PACK1 FROM DRIVE 1 AND SAVE.
// * MOUNT PID PACK2 ON DRIVE 0 AND SCRATCH PACK ON DRIVE 1.
// * DO NOT COLD START.
// *
// *
// *
// *  WHEN MPX/BOM MESSAGES ARE PRINTED-- NPRO READER, PLACE BDWAP DECK
// *  IN READER, FOLLOW WITH NPRO JOB CARD AND REMAINDER OF STACKED
// *  INPUT, READY READER, TURN ON SS 0, DATA SWITCH 15, AND PRESS
// *  CONSOLE START.
// *
// *  LEAVE DATA SWITCH 15 ON FOR ONE TRY. PRESS CONSOLE START.
// *
// *  LEAVE DATA SWITCH 15 ON FOR DRIVE 1. PRESS CONSOLE START.
// *
// *  AT COMPLETION OF BDWAP, TURN OFF CONSOLE SWITCHES, PRESS
// *  STOP, RESET AND START.  THEN SET PROGRAM SWITCH 7, AND PRESS
// *  CONSOLE INTERRUPT TO CONTINUE.
// *
// END   00 JAN 00 00.000 HRS
```

The following is then printed on the system printer:

```
IBM 1800 DACS  MPX/SYS GEN MON   00.000
SEN SW 0 ON ABSOLUTE LOADER
SEN SW 1 ON LOAD BP MONITOR
```

30. Unload and save cartridges from drives 0 and 1. (PID pack 1 master and its copy.)

31. Mount PID pack 2 on drive 0 and a scratch pack on drive 1.

32. Write disk addresses on scratch pack (drive 1) by repeating steps 15 through 29. The BDWAP card deck should be removed from the 1442 and saved for future use.

33. When you have redone those steps, the following messages will be printed:

```
// JOB   00002   00 JAN 00 00.000 HRS
// DMP   00 JAN 00 00.000 HRS
*DLABL  1  11112
DMP FUNCTION COMPLETED
```

This labels drive 1 with the label 11112.

```
// JOB   0000211112   00 JAN 00 00.000 HRS
// DMP   00 JAN 00 00.000 HRS
*DEFINE CONFG C1              163
DMP FUNCTION COMPLETED
```

This defines a Core Image Area on drive 1.

```
// JOB   0000211112   00 JAN 00 00.000 HRS
// *
// *  AT PAUSE PLACE USER-COMPLETED EQUATE CARDS BEHIND
// *  *SRFLE M CONTROL CARD FOR BOM
// *  PRESS READER AND CONSOLE START TO CONTINUE.
// *
// DMP   00 JAN 00 00.000 HRS
*DFILE    1         BOM   1304
WILL RESERVE AT SCTR ADDR   1128
DMP FUNCTION COMPLETED
//  PAUSE   00 JAN 00 00.000 HRS
```

34. Remove all cards from the 1442 hopper.

35. Nonprocess out all the cards inside the 1442. Omit all these cards except the // DMP and *SRFLE control cards.

36. At this point you should complete your BOM, Executive Director and various Subroutine Library program equate cards (if this has not already been done).

INSERTING EQUATE CARDS

37. Place your completed BOM equate cards behind the *SRFLE card from step 35. Place these cards in the 1442 hopper. Fill the hopper with the remaining system-generation input deck from step 34.

38. Press console START. The following messages will be printed:

```
CONTINUE
// DMP    00 JAN 00 00.000 HRS
*SRFLE M              BOMCD BOM          BOM EQUATE CARDS MUST FOLLOW
016548 CARDS ON FILE
001199 SECTORS USED
000105 SECTORS AVAIL
000217 CARDS READ
DMP FUNCTION COMPLETED
// *
// *  REMOVE PID CARTRIDGE 2 FROM DRIVE 0 AND SAVE.
// *  MOUNT COPY OF PID CARTRIDGE 1 ON DRIVE 0.
// *
// * NEXT PUSH IMMEDIATE STOP AND START TO RELOAD SGMON.
// *
// * THEN WITH SENSE SWITCH 1 ON PUSH START TO LOAD BPMON.
// PAUSE   00 JAN 00 00.000 HRS
```

39. Unload the PID cartridge 2 from drive 0.

40. Load drive 0 with a copy of PID cartridge 1.

41. Press console IMMEDIATE STOP and START. The following messages are printed on the system printer:

```
IBM 1800 DACS  MPX/SYS GEN MON   00.000
SEN SW 0 ON ABSOLUTE LOADER
SEN SW 1 ON LOAD BP MONITOR
```

42. Set sense switch 1 ON. Press console START. The following messages are printed:

```
// JOB   00001   00 JAN 00 00.000 HRS
// *
// *  AT PAUSE PLACE USER COMPLETED LIBRARY EQUATE CARDS BEHIND RESPECTIVE
// *  *SRFLE M CONTROL CARDS FOR CSPCD,LSPCD,MDFCD,MFICD,ADRCD,DMPSC,EXDCD.
// *  PRESS READER AND CONSOLE START TO CONTINUE.
// *
// *   BE SURE TO INCLUDE A SRFLE END CARD AT THE END OF EACH
// *   SET OF EQUATES.
// *
// PAUSE   00 JAN 00 00.000 HRS
```

43. Place your completed Executive Director and Subroutine Library equate cards behind their respective *SRFLE control cards. The *SRFLE control cards are in the system-generation input file in the 1442 card read punch. Be sure that you have included cards containing 9s in the last eight columns behind each set of equates (refer to step 10).

    The last set of equates in the library equate deck are for BSCIO and will be placed behind the proper *SRFLE control card in a later step. Mark these equates as BSCIO equates and set them aside.

44. Ready the 1442.

45. Press console START. The following messages are printed indicating the processing of the equate cards by the SRFLE function:

```
CONTINUE
// JOB    00001    00 JAN 00 00.000 HRS
// DMP    00 JAN 00 00.000 HRS
*SRFLE M              EXDCD              EXECUTIVE EQUATE CARDS MUST FOLLOW
000122 CARDS READ
000005 SECTORS AVAIL
DMP FUNCTION COMPLETED
*SRFLE M              CSPCD              CSPAR EQUATE CARDS MUST FOLLOW
000003 CARDS READ
000003 SECTORS AVAIL
DMP FUNCTION COMPLETED
*SRFLE M              LSPCD              LSPCL EQUATE CARDS MUST FOLLOW
000003 CARDS READ
000003 SECTORS AVAIL
DMP FUNCTION COMPLETED
*SRFLE M              MDFCD              MDFIO EQUATE CARDS MUST FOLLOW
000005 CARDS READ
000008 SECTORS AVAIL
DMP FUNCTION COMPLETED
*SRFLE M              MFICD              MFIO  EQUATE CARDS MUST FOLLOW
000011 CARDS READ
000006 SECTORS AVAIL
DMP FUNCTION COMPLETED
*SRFLE M              ADRCD              ADRCK EQUATE CARDS MUST FOLLOW
000002 CARDS READ
000006 SECTORS AVAIL
DMP FUNCTION COMPLETED
*SRFLE M              DMPSC              DMPS  EQUATE CARDS MUST FOLLOW
000002 CARDS READ
000004 SECTORS AVAIL
DMP FUNCTION COMPLETED
```

Go to "Assembly of System Programs for a System with 2315 Disk Cartridges."

SYSTEM GENERATION USING A 1316 DISK PACK

Because your system generation uses a 1316 disk pack and your system has only one 2311 disk drive, you will now produce a copy of your PID pack on cards.

DUMPING THE PID PACK TO CARDS

46. Remove all cards from the 1442 hopper.

47. Nonprocess out all the cards inside the 1442. Omit all those cards except the last, which is a // JOB card.

48. Place the BOM Disk Dump and Reload program, BDCRL (one of the BOM utilities punched in step 13), into the 1442 hopper.

49. Dump cylinder 0 to cards (refer to "Bom Card Utilities").

50. Remove the BDCRL card deck from the 1442 stacker and save it.

51. Remove the cards containing the dump of the PID pack and save it.

52. Punch the following statements into cards and place those cards in the 1442 hopper:

```
// JOB    0000100002
// DMP
*DUMP         DK0 PN
*DUMP         DK1 PN
```

53. Follow each of the *DUMP statements by approximately 12,000 cards. You will now dump the mapped 1810 drives to cards.

54. Set sense switch 7 ON.

55. Ready the 1442 card read punch.

56. Press CONSOLE INTERRUPT.

57. When the dump operations have been completed, save the cards removed from the 1442. Skip to step 71.

DUPLICATING THE PID DISK PACK

Because your system has two or more 2311 drives, you will now copy the PID pack to a scratch pack.

58. Load your scratch pack on drive 1.

59. Remove all cards from the 1442 hopper.

60. Nonprocess out all cards inside the 1442. Omit all those cards except the last which is a // JOB card.

61. Place the BOM 2311 Disk Pack Initialization Program, BDPIP (one of the BOM utilities punched in step 13), into the 1442 hopper.

62. Follow BDPIP in the 1442 hopper by control cards necessary to initialize the disk pack on drive 1 (refer to "BOM Card Utilities").

63. Follow the BDPIP control cards by the BOM Disk Duplication Program, BDUPL (one of the BOM utilities punched in step 13).

64. Follow BDUPL by the remainder of the system-generation input deck from steps 59 and 60.

65. Ready the 1442.

66. Set sense switch 0 and data switch 15 ON. Press console START.

67. During the execution of BDPIP, refer to "BOM Card Utilities" for BDPIP operating procedures.

68. When execution of BDPIP has been successfully completed, set sense switch 0 and data switch 15 ON and press console START.

69. During execution of BDUPL, refer to "BOM Card Utilities" for BDUPL operating procedures.

70. Unload and save the pack on drive 1 (it is now a copy of the PID pack).

71. Set all sense switches, data switches and program switches OFF; press console STOP, RESET and START.

72. When the following messages are printed on the system printer, set program switch 7 ON and press CONSOLE INTERRUPT:

```
IBM 1800 DACS  MPX/SYS GEN MON   00.000
SEN SW 0 ON ABSOLUTE LOADER
SEN SW 1 ON LOAD BP MONITOR
```

73. The following messages are printed:

```
// JOB   00001   00 JAN 00 00.000 HRS
// *
// *  AT PAUSE, INSERT YOUR COMPLETED BOM, EXECUTIVE DIRECTOR AND SUBROUTINE
// *  EQUATE CARDS, FOLLOWED BY SRFLE END CARDS, BEHIND THEIR RESPECTIVE
// *  *SRFLE M CONTROL CARDS.
// *  PRESS READER AND CONSOLE START TO CONTINUE.
// *
// *   BE SURE TO INCLUDE A SRFLE END CARD AT THE END OF EACH
// *   SET OF EQUATES.
// *
// PAUSE   00 JAN 00 00.000 HRS
```

74. At the pause, remove all cards from the 1442 stacker, saving the BDUPL and BDPIP card decks and omitting the control cards.

75. At this point you should complete your BOM, Executive Director and Subroutine Library program equate cards (if this has not already been done).

INSERTING EQUATE CARDS

76. Place your completed BOM, Executive Director, and Subroutine Library equates behind their respective *SRFLE control cards. The *SRFLE control cards are in the system-generation input file in the 1442 card read punch. Be sure that you have included cards containing 9s in the last eight columns behind each set of equates for the library programs (see step 10).

    The last set of equates in the library equate deck are for BSCIO and will be placed behind the proper *SRFLE control card in a later step. Mark these equates as BSCIO equates and set them aside. Place the remainder of the system-generated input (from steps 46 and 47) into the 1442 hopper.

77. Ready the 1442.

78. Press console START. The following messages are printed, indicating the processing of the equate cards:

```
CONTINUE
// JOB   0000100002   00 JAN 00 00.000 HRS
// DMP   00 JAN 00 00.000 HRS
*SRFLE M            BOMCD              BOM EQUATE CARDS MUST FOLLOW
000217 CARDS READ
000105 SECTORS AVAIL
DMP FUNCTION COMPLETED
*SRFLE M            EXDCD              EXECUTIVE EQUATE CARDS MUST FOLLOW
000122 CARDS READ
000005 SECTORS AVAIL
DMP FUNCTION COMPLETED
*SRFLE M            CSPCD              CSPAR EQUATE CARDS MUST FOLLOW
000003 CARDS READ
000003 SECTORS AVAIL
DMP FUNCTION COMPLETED
*SRFLE M            LSPCD              LSPCL EQUATE CARDS MUST FOLLOW
000003 CARDS READ
000003 SECTORS AVAIL
DMP FUNCTION COMPLETED
*SRFLE M            MDFCD              MDFIO EQUATE CARDS MUST FOLLOW
000005 CARDS READ
000008 SECTORS AVAIL
DMP FUNCTION COMPLETED
*SRFLE M            MFICD              MFIO  EQUATE CARDS MUST FOLLOW
000011 CARDS READ
000006 SECTORS AVAIL
DMP FUNCTION COMPLETED
*SRFLE M            ADRCD              ADRCK EQUATE CARDS MUST FOLLOW
000002 CARDS READ
000006 SECTORS AVAIL
DMP FUNCTION COMPLETED
*SRFLE M            DMPSC              DMPS  EQUATE CARDS MUST FOLLOW
000002 CARDS READ
000004 SECTORS AVAIL
DMP FUNCTION COMPLETED
```


ASSEMBLY OF SYSTEM PROGRAMS FOR A SYSTEM WITH ONE OR MORE 1316 DISK PACKS

79. The printout continues:

```
// JOB   00001   00 JAN 00 00.000 HRS
// *  THE FOLLOWING CONTROL CARDS WILL DELETE THE ENTIRE COMMUNICATION
// *  ADAPTER OBJECT LIBRARY.  IF THIS IS DESIRED, PRESS START.
// *  IF IT IS NOT DESIRED SET PROGRAM SWITCH 7 UP AND PRESS CONSOLE INTERRUPT.
// *
// PAUSE   00 JAN 00 00.000 HRS
```

79a. If your system is to support one or more communications adapters, set program switch 7 ON and press CONSOLE INTERRUPT. Go to step 80 to continue.

79b. If your system does not include communications adapters, press console START and the following programs will be deleted:

```
CONTINUE
// DMP   00 JAN 00 00.000 HRS
*DELET              BSCRT                    CA ON-LINE TEST ROUTINE
DMP FUNCTION COMPLETED
*DELET    R         BSCTR                    CA TRACE ROUTINE
DMP FUNCTION COMPLETED
*DELET              BSCTR                    CA TRACE ROUTINE
DMP FUNCTION COMPLETED
*DELET              BSCCK                    I/O AREA CHECK ROUTINE
DMP FUNCTION COMPLETED
*DELET    R         ZIPCO                    CONVERSION ROUTINE
DMP FUNCTION COMPLETED
*DELET              ZIPCO                    CONVERSION ROUTINE
CMP FUNCTION COMPLETED
*DELET              USHOL                    USASCII CODE TO HOLLERITH TBL
DMP FUNCTION COMPLETED
*DELET              HOLUS                    HOLLERITH TO USASCII CODE TBL
DMP FUNCTION COMPLETED
*DELET              USEBC                    USASCII TO EBCDIC TABLE
DMP FUNCTION COMPLETED
*DELET              EBCUS                    EBCDIC TO USASCII TABLE
DMP FUNCTION COMPLETED
*DELET              USPRT                    USASCII TO 1443 PRINTER TABLE
DMP FUNCTION COMPLETED
*DELET              USTYP                    USASCII TO TYPEWRITER TABLE
DMP FUNCTION COMPLETED
*DELET    R         PEFIB                    PACKED EBCDIC TO FIXED BINARY
DMP FUNCTION COMPLETED
*DELET              PEFIB                    PACKED EBCDIC TO FIXED BINARY
DMP FUNCTION COMPLETED
*DELET    R         FLBPE                    FLOATING BINARY TO PACKED EBCDIC
DMP FUNCTION COMPLETED
*DELET              FLBPE                    FLOATING BINARY TO PACKED EBCDIC
DMP FUNCTION COMPLETED
```

80. The printout continues:

```
// JOB   00001   00 JAN 00 00.000 HRS
// *
// * IF THE LIBRARY SUBROUTINES TO SUPPORT THE USE OF 2311'S ARE NOT
// * NEEDED PRESS START.  OTHERWISE, PRESS CONSOLE INTERRUPT WITH
// * SENSE SWITCH 7 ON.
// *
// PAUSE   00 JAN 00 00.000 HRS
```

80a. If your system is to support one or more 2311s, set program switch 7 ON and press CONSOLE INTERRUPT. Go to step 81. If your system is not to support 2311s, go to step 80b.

80b. If your system does not support 2311s, press console START and the following programs will be deleted:

```
CONTINUE
// DMP   00 JAN 00 00.000 HRS
*DELET    R         DSDR
DMP FUNCTION COMPLETED
*DELET    R         DSCR
DMP FUNCTION COMPLETED
*DELET    R         MDAIO
DMP FUNCTION COMPLETED
*DELET              DPIP
DMP FUNCTION COMPLETED
*DELET    R         RESER
DMP FUNCTION COMPLETED
*DELET    R       0 DSPRO
DMP FUNCTION COMPLETED
// JOB   0000100002   00 JAN 00 00.000 HRS
// *  THE FOLLOWING CONTROL CARDS WILL DELETE THE 2790 OBJECT AND SOURCE
// *  LIBRARIES.  IF THIS IS DESIRED, PRESS START, OTHERWISE SET SENSE
// *  SWITCH 7 UP AND PRESS CONSOLE INTERRUPT.
// *
// PAUSE   00 JAN 00 00.000 HRS
```

81. If your system is to support the 2790 system, set program switch 7 ON and press CONSOLE INTERRUPT. Go to step 82. If your system is not to support the 2790 system, go to step 81a.

81a. If your system does not support the 2790 system, press console START and the following programs will be deleted.


```
CONTINUE
// DMP   00 JAN 00 00.000 HRS
*DELET              CLIP
DMP FUNCTION COMPLETED
*DELET              CLTP
DMP FUNCTION COMPLETED
*DELET              FILES
DMP FUNCTION COMPLETED
*DELET              LINKF
DMP FUNCTION COMPLETED
*DELET    R         LOOP1
DMP FUNCTION COMPLETED
*DELET    R         LOOP2
DMP FUNCTION COMPLETED
*DELET    R         WRTLN
DMP FUNCTION COMPLETED
*DELET    D         A2790
DMP FUNCTION COMPLETED
// JOB   00 JAN 00 00.000 HRS.
// DMP   00 JAN 00 00.000 HRS.
*DEFINE PAKDK 0
DMP FUNCTION COMPLETED
```

82. Now assemble the Executive Director.

```
// JOB   0000100002   00 JAN 00 00.000 HRS.
// *  ASSEMBLE EXECUTIVE DIRECTOR, PUNCH THEN STORE.
// ASM EXDIR EXDCD   00 JAN 00 00.000 HRS
*OVERFLOW SECTORS 32,0,0                                    EXD00005
*RE-ENTRANT                                                FEXD00010
*PUNCH                                                     FEXD00015

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.


EXDIR SPECL BACK  QLEVL
DMP FUNCTION COMPLETED
*STORE              EXDIR
EXDIR SPECL BACK  QLEVL
DMP FUNCTION COMPLETED
// *
// * NOW ASSEMBLE AND PUNCH BOM
// *
// ASM BOM   BOMCD   00 JAN 00 00.000 HRS
*OVERFLOW SECTORS 32,0,0                                    BOM00005
*PUNCH                                                     FBOM00010

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.


BOM
DMP FUNCTION COMPLETED
```

83. **Remove and label the Executive Director and BOM object decks, omitting control cards. Without operator intervention, the system-generation procedure will continue to assemble additional system subroutines.**

```
// *
// *  THE FOLLOWING CONTROL CARDS ASSEMBLE AND STORE THE SUBROUTINES
// *  AFFECTED BY THE JUST-INSERTED LIBRARY EQUATE CARDS
// *
// ASM ADRCK ADRCD   00 JAN 00 00.000 HRS
*RE-ENTRANT                                                    ADK00010

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.


ADRCK
DMP FUNCTION COMPLETED
*STORE            0 ADRCK
ADRCK
DMP FUNCTION COMPLETED
// ASM DMPS  DMPSC   00 JAN 00 00.000 HRS

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.


DMPS  DUMPS DMPST DMP   DUMP  DMPHX DMPDC
DMP FUNCTION COMPLETED
*STORE            0 DMPS
DMPS  DUMPS DMPST DMP   DUMP  DMPHX DMPDC
DMP FUNCTION COMPLETED
// ASM CSPAR CSPCD   00 JAN 00 00.000 HRS
*RE-ENTRANT                                                    CSP00010

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.




CSPAR CSPLS
DMP FUNCTION COMPLETED
*STORE            0 CSPAR
CSPAR CSPLS
DMP FUNCTION COMPLETED
// ASM LSPCL LSPCD   00 JAN 00 00.000 HRS
*RE-ENTRANT                                                    SLP00010

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.


LSPCL LSPPP
DMP FUNCTION COMPLETED
*STORE            0 LSPCL
LSPCL LSPPP
DMP FUNCTION COMPLETED
// ASM MDFIO MDFCD   00 JAN 00 00.000 HRS
*RE-ENTRANT                                                    MDI00030

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.


MDFIO MDAF  MDAI  MDCOM MDF   MDFX  MDI   MDIX  MDRED MDWRT
DMP FUNCTION COMPLETED
*STORE            0 MDFIO
MDFIO MDAF  MDAI  MDCOM MDF   MDFX  MDI   MDIX  MDRED MDWRT
DMP FUNCTION COMPLETED
// ASM MFIO  MFICD   00 JAN 00 00.000 HRS
*RE-ENTRANT                                                    MFI00010

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.


MFIO  MRED  MWRT  MCOMP MIOAF MIOAI MIOFX MIOIX MIOF  MIOI
DMP FUNCTION COMPLETED
*STORE            0 MFIO
MFIO  MRED  MWRT  MCOMP MIOAF MIOAI MIOFX MIOIX MIOF  MIOI
DMP FUNCTION COMPLETED
```

84. At the completion of step 83, the following is printed:

```
// JOB   0000100002   00 JAN 00 00.000 HRS
// *
// *   THE FOLLOWING FUNCTIONS ARE CONCERNED WITH THE ASSEMBLY OF THE
// *   COMMUNICATIONS IOCR (BSCIO).  IF THIS SYSTEM DOES NOT SUPPORT
// *   COMMUNICATIONS PRESS CONSOLE INTERRUPT WITH SENSE SWITCH 7 UP,
// *   OTHERWISE PRESS START.
// *
// PAUSE   00 JAN 00 00.000 HRS
```


85. If your system is to support no communications adapters, set program switch 7 ON and press CONSOLE INTERRUPT. N11 error messages will be printed and are to be ignored. Go to step 86 to continue.

85a. If your system is to support one or more communications adapters, press START and the following messages are printed:

```
CONTINUE
// *
// *   AT PAUSE PLACE BSCIO SYSGEN EQUATES AFTER *SRFLE CONTROL CARD
// *   INCLUDE A SRFLE END CARD AT THE END OF THE EQU'S
// *   PRESS READER START AND CONSOLE START TO CONTINUE
// *
// PAUSE   00 JAN 00 00.000 HRS
```

85b. Insert the BSCIO equate cards behind the *SRFLE card in the 1442. Ready the 1442 and press console START. The following is printed:

```
CONTINUE
// DMP   00 JAN 00 00.000 HRS
*SRFLE M           BSCCD
000005 CARDS READ
000011 SECTORS AVAILABLE
DMP FUNCTION COMPLETED
// *
// *
// *  ASSEMBLE AND PUNCH BSCIO
// *
// ASM BSCIO BSCCD   00 JAN 00 00.000 HRS
*RE-ENTRANT                                                    BSC00010
*PUNCH                                                         BSC00020


000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.
BSCIO
DMP FUNCTION COMPLETED
*STORE            0 BSCIO
BSCIO
DMP FUNCTION COMPLETED
```

85c. Remove and label the BSCIO object deck, omitting the control cards.

86. The printout continues:

```
// JOB   0000100002   00 JAN 00 00.000 HRS
// *
// *  THE FOLLOWING SOURCE FILES AND UTILITIES ARE DELETED
// * DELETE-- EQUBM, EQUEX, EOULB, GENCD, CSPCD, LSPCD, MDFCD, MFICD, EXDCD
// * BLIST, BDWAP, SYSLD, BDUPL, BDPAT, BDPIP, BDCRL, ADRCD, DMPSC, SAMP1
// * BOMCD, BSCCD
// * IF ALL PRECEDING OPERATIONS COMPLETED SUCCESSFULLY,
// * PRESS CONSOLE START TO CONTINUE.
// *
// PAUSE      PRESS START TO CONTINUE   00 JAN 00 00.000 HRS
```

If your system generation has not proceeded successfully to this point, don 't continue. Correct the error and go back to the appropriate step to continue the generation process.

If your system generation has proceeded successfully to this point, press console START. This will cause the deletion of system-generation source files on the disk and bring you to the end of phase 2.

```
CONTINUE
// DMP   00 JAN 00 00.000 HRS
*DELET    D         EQUBM
DMP FUNCTION COMPLETED
*DELET    D         EQUEX
DMP FUNCTION COMPLETED
*DELET    D         EQULB                    LIBRARY EQUATES
DMP FUNCTION COMPLETED
*DELET    D         GENCD
DMP FUNCTION COMPLETED
*DELET    D         CSPCD                    CSPAR SOURCE
DMP FUNCTION COMPLETED
*DELET    D         LSPCD                    LSPCL SOURCE
DMP FUNCTION COMPLETED
*DELET    D         MDFCD                    MDFIO SOURCE
DMP FUNCTION COMPLETED
*DELET    D         MFICD                    MFIO SOURCE
DMP FUNCTION COMPLETED
*DELET    D         EXDCD                    EXECUTIVE DIRECTOR SOURCE
DMP FUNCTION COMPLETED
*DELET              BLIST
DMP FUNCTION COMPLETED
*DELET              BDWAP
DMP FUNCTION COMPLETED
*DELET              SYSLD
DMP FUNCTION COMPLETED
*DELET              BDUPL
DMP FUNCTION COMPLETED
*DELET              BDPAT
DMP FUNCTION COMPLETED
*DELET              BDPIP
DMP FUNCTION COMPLETED
*DELET              BDCRL
DMP FUNCTION COMPLETED
*DELET              BDUMP
DMP FUNCTION COMPLETED
*DELET              BRELD
DMP FUNCTION COMPLETED
*DELET    D         ADRCD                    ADRCK SOURCE
DMP FUNCTION COMPLETED
*DELET    D         DMPSC                    DMPS SOURCE
DMP FUNCTION COMPLETED
*DELET    D         SAMP1                    MPX SAMPLE PGM SOURCE
DMP FUNCTION COMPLETED
*DELET    D         BOMCD
DMP FUNCTION COMPLETED
*DELET    D         BSCCD
DMP FUNCTION COMPLETED
```

```
// JOB    00 JAN 00 00.000 HRS
// DMP    00 JAN 00 00.000 HRS
*DEFINE PAKDK 0
DMP FUNCTION COMPLETED


// JOB    00 JAN 00 00.000 HRS
// *  AT THIS POINT REFER TO OP MANUAL FOR FURTHER INSTRUCTIONS.
// END    2790 USERS SHOULD REFER TO 2790 SRL FOR BUILDING CORELOADS.
```


2790 INSTALLATION

If you are installing the 2790 system, it is necessary to store to disk drive 0 the On Line 2790 Area Station Exerciser/Customer Engineer Diagnostic Program. The object card deck for this program can be obtained from your customer engineer and stored in core image format using the following job steps:

```
1-10      11-20     21-30     31-40     41-50     51-60
1234567890 1234567890 1234567890 1234567890 1234567890 1234567890
1 //JOB
2 //DMP
3 *STORE    T RD    0 @CEX@
4   .
5   .  } Object deck of 2790 Exerciser/Diagnostic
6   .
7 *STORECIX          0 @CEX@ @CEX@
8 *LOCALPCDIG,ASDIG,ERRST,DRECT,(RQDI,RQD2,CONSL)
9 *CCEND
10
```

These cards are included with the object card deck.

This completes phase 2.

ASSEMBLY OF SYSTEM PROGRAMS FOR A SYSTEM WITH 2315 DISK CARTRIDGES

87. The following is printed:

```
// JOB    00001   00 JAN 00 00.000 HRS
// *  THE FOLLOWING CONTROL CARDS WILL DELETE THE ENTIRE COMMUNICATION
// *  ADAPTER OBJECT LIBRARY.  IF THIS IS DESIRED, PRESS START.
// *  IF IT IS NOT DESIRED SET PROGRAM SWITCH 7 UP AND PRESS CONSOLE INTERRUPT.
// *
// PAUSE   00 JAN 00 00.000 HRS
```

88. If your system is to support one or more communications adapters, set program switch 7 ON and press CONSOLE INTERRUPT. Go to step 89 to continue.

88a. If your system does not include communications adapters, press console START and the following programs will be deleted:

```
CONTINUE
// DMP    00 JAN 00 00.000 HRS
*DELET               BSCRT                    CA ON-LINE TEST ROUTINE
DMP FUNCTION COMPLETED
*DELET    R          BSCTR                    CA TRACE ROUTINE
DMP FUNCTION COMPLETED
*DELET               BSCTR                    CA TRACE ROUTINE
DMP FUNCTION COMPLETED
*DELET               BSCCK                    I/O AREA CHECK ROUTINE
DMP FUNCTION COMPLETED
*DELET    R          ZIPCO                    CONVERSION ROUTINE
DMP FUNCTION COMPLETED
*DELET               ZIPCO                    CONVERSION ROUTINE
DMP FUNCTION COMPLETED
*DELET               USHOL                    USASCII CODE TO HOLLERITH TBL
DMP FUNCTION COMPLETED
*DELET               HOLUS                    HOLLERITH TO USASCII CODE TBL
DMP FUNCTION COMPLETED
*DELET               USEBC                    USASCII TO EBCDIC TABLE
DMP FUNCTION COMPLETED
*DELET               EBCUS                    EBCDIC TO USASCII TABLE
DMP FUNCTION COMPLETED
*DELET               USPRT                    USASCII TO 1443 PRINTER TABLE
DMP FUNCTION COMPLETED
*DELET               USTYP                    USASCII TO TYPEWRITER TABLE
DMP FUNCTION COMPLETED
*DELET    R          PEFIB                    PACKED EBCDIC TO FIXED BINARY
DMP FUNCTION COMPLETED
*DELET               PEFIB                    PACKED EBCDIC TO FIXED BINARY
DMP FUNCTION COMPLETED
*DELET    R          FLBPE                    FLOATING BINARY TO PACKED EBCDIC
DMP FUNCTION COMPLETED
*DELET               FLBPE                    FLOATING BINARY TO PACKED EBCDIC
DMP FUNCTION COMPLETED
```

89. The printout continues:

```
// JOB    00001    00 JAN 00 00.000 HRS
// *
// * IF THE LIBRARY SUBROUTINES TO SUPPORT THE USE OF 2311'S ARE NOT
// * NEEDED PRESS START.  OTHERWISE, PRESS CONSOLE INTERRUPT WITH
// * SENSE SWITCH 7 ON .
// *
// PAUSE    00 JAN 00 00.000 HRS
```

89a. If your system is to support one or more 2311s, set program switch 7 ON and press CONSOLE INTERRUPT. Go to step 90.

89b. If your system does not support 2311s, press console START and the following programs will be deleted:

```
CONTINUE
// DMP    00 JAN 00 00.000 HRS
*DELET    R          DSOR
DMP FUNCTION COMPLETED
*DELET    R          DSCR
DMP FUNCTION COMPLETED
*DELET    R          MDAIO
DMP FUNCTION COMPLETED
*DELET               DPIP
DMP FUNCTION COMPLETED
*DELET    R          RESER
DMP FUNCTION COMPLETED
*DELET    R        0 DSPRO
DMP FUNCTION COMPLETED


// JOB    00001    00 JAN 00 00.000 HRS
// *  THE FOLLOWING CONTROL CARDS WILL DELETE THE 2790 OBJECT LIBRARY.
// *  IF THIS IS DESIRED, PRESS START, OTHERWISE SET SENSE SWITCH 7 UP
// *  AND PRESS CONSOLE INTERRUPT.
// *
// PAUSE    00 JAN 00 00.000 HRS
```

90. If your system is to support the 2790 system, set program switch 7 ON and press CONSOLE INTERRUPT. Go to step 91.

90a. If your system does not support the 2790 system, press console START and the following programs will be deleted:

```
CONTINUE
// DMP   00 JAN 00 00.000 HRS
*DELET               CLIP
DMP FUNCTION COMPLETED
*DELET               CLTP
DMP FUNCTION COMPLETED
*DELET               FILES
DMP FUNCTION COMPLETED
*DELET               LINKF
DMP FUNCTION COMPLETED
*DELET    R          LOOP1
DMP FUNCTION COMPLETED
*DELET    R          LOOP2
DMP FUNCTION COMPLETED
*DELET    R          WRTLN
DMP FUNCTION COMPLETED


// JOB   00 JAN 00 00.000 HRS.
// DMP   00 JAN 00 00.000 HRS.
*DEFINE PAKDK 0
DMP FUNCTION COMPLETED
```

91. The Executive Director and BOM are now assembled. These operations may take as long as two hours, depending on the main-storage size of your system. At the completion of the previous step, the following messages are printed, indicating that the Executive Director and BOM have been assembled.

```
// JOB   0000111112   00 JAN 00 00.000 HRS
// *  ASSEMBLE EXECUTIVE DIRECTOR, PUNCH THEN STORE.
// ASM EXDIR EXDCD   00 JAN 00 00.000 HRS
*OVERFLOW SECTORS 32,0,0                                          EXD00005
*RE-ENTRANT                                                       FEXD00010
*PUNCH                                                            FEXD00015

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.

EXDIR SPECL BACK  QLEVL
DMP FUNCTION COMPLETED
*STORE              EXDIR
EXDIR SPECL BACK  OLEVL
DMP FUNCTION COMPLETED
*STORE              EXDIR
DMP FUNCTION COMPLETED
// *
// * NOW ASSEMBLE AND PUNCH BOM
// *  IF YOU GET A D11 NEED BLANK CARDS ERROR MESSAGE
// *  NPRO THE CARDS IN THE CARD READER AND INSERT BLANK CARDS.
// *
// ASM BOM   BOM   00 JAN 00 00.000 HRS
*OVERFLOW SECTORS 32,0,0                                           BOM00005
*PUNCH                                                            FBOM00010

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.


BOM
DMP FUNCTION COMPLETED
```

92. **Remove and label the Executive Director and BOM object decks, omitting control cards. Without operator intervention, the system-generation procedure will continue to assemble additional system subroutines.**

```
// *
// *  THE FOLLOWING CONTROL CARDS ASSEMBLE AND STORE THE SUBROUTINES
// *  AFFECTED BY THE JUST-INSERTED LIBRARY EQUATE CARDS
// *
// ASM ADRCK ADRCD   00 JAN 00 00.000 HRS
*RE-ENTRANT                                                    ADK00010

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.


ADRCK
DMP FUNCTION COMPLETED
*STORE            0 ADRCK
ADRCK
DMP FUNCTION COMPLETED
// ASM DMPS  DMPSC   00 JAN 00 00.000 HRS

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.


DMPS  DUMPS DMPST DMP   DUMP  DMPHX DMPDC
DMP FUNCTION COMPLETED
*STORE            0 DMPS
DMPS  DUMPS DMPST DMP   DUMP  DMPHX DMPDC
DMP FUNCTION COMPLETED
// ASM CSPAR CSPCD   00 JAN 00 00.391 HRS
*RE-ENTRANT                                                    CSP00010

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.


CSPAR CSPLS
DMP FUNCTION COMPLETED
*STORE            0 CSPAR
CSPAR CSPLS
DMP FUNCTION COMPLETED
// ASM LSPCL LSPCD   00 JAN 00 00.423 HRS
*RE-ENTRANT                                                    SLP00010

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.


LSPCL LSPPP
DMP FUNCTION COMPLETED
*STORE            0 LSPCL
LSPCL LSPPP
DMP FUNCTION COMPLETED
// ASM MDFIO MDFCD   00 JAN 00 00.467 HRS
*RE-ENTRANT                                                    MDI00030

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.


MDFIO MDAF  MDAI  MDCOM MDF   MDFX  MDI   MDIX  MDRED MDWRT
DMP FUNCTION COMPLETED
*STORE            0 MDFIO
MDFIO MDAF  MDAI  MDCOM MDF   MDFX  MDI   MDIX  MDRED MDWRT
DMP FUNCTION COMPLETED
// ASM MFIO  MFICD   00 JAN 00 00.542 HRS
*RE-ENTRANT                                                    MFI00010

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.


MFIO  MRED  MWRT  MCOMP MIOAF MIOAI MIOFX MIOIX MIOF  MIOI
DMP FUNCTION COMPLETED
*STORE            0 MFIO
MFIO  MRED  MWRT  MCOMP MIOAF MIOAI MIOFX MIOIX MIOF  MIOI
DMP FUNCTION COMPLETED
```

93. At the completion of step 92, the following is printed:

```
// JOB   00001   00 JAN 00 00.000 HRS
// *
// *  THE FOLLOWING FUNCTIONS ARE CONCERNED WITH THE ASSEMBLY OF THE COMM-
// *  UNICATIONS ADAPTER IOCA (BSCIO) AND BUILDING OF THE 2790 FILE ON DR 1.
// *  IF THIS SYSTEM DOES NOT SUPPORT COMMUNICATIONS ADAPTER AND 2790,
// *  PRESS CONSOLE INTERRUPT WITH SENSE SWITCH 7 UP, OTHERWISE PRESS START.
// *
// *  IF CONSOLE INTERRUPT IS USED IGNORE ALL ERRORS OF THE FORM
// *
// *  N11 LABEL ERR DR N
// *
// PAUSE   00 JAN 00 00.000 HRS
```

93a. If your system does not support the communications adapter or the 2790 system, set program switch 7 ON and press CONSOLE INTERRUPT. If this option is taken, N11 LABEL ERR DR N will be printed until the next JOB 00001 card is encountered. Go to step 102.

94. If your system is to support one or more communications adapters or the 2790 system, press START and the following messages will be printed:

```
CONTINUE
// *
// *  AT PAUSE PRESS STOP RESET AND START
// *
// *  MOUNT PID PACK 3 ON DRIVE 0 AND A SCRATCH PACK ON DRIVE 1.
// *
// *  WHEN MPX/BOM MESSAGES ARE PRINTED, NPRO READER, PLACE BDWAP DECK IN
// *  READER, FOLLOW WITH NPRO JOB CARD AND REMAINDER OF STACKED INPUT,
// *  READY READER, TURN ON SS 0, DATA SWITCH 15, AND PRESS CONSOLE START.
// *
// *  LEAVE DATA SWITCH 15 ON FOR ONE TRY.  PRESS CONSOLE START.
// *
// *  LEAVE DATA SWITCH 15 ON FOR DRIVE 1.  PRESS CONSOLE START.
// *
// *  AT COMPLETION OF BDWAP, TURN OFF CONSOLE SWITCHES, PRESS STOP, RESET
// *  AND START. THEN SET PROGRAM SW 7, AND PRESS CONSOLE INTERRUPT TO CONTINUE
// *
// PAUSE   00 JAN 00 00.000 HRS


// JOB   00003   00 JAN 00 00.000 HRS
// DMP   00 JAN 00 00.000 HRS
*DLABL  1  11113
DMP FUNCTION COMPLETED


// JOB   0000311113   00 JAN 00 00.000 HRS
// DMP   00 JAN 00 00.000 HRS
*DEFINE CONFG C1              0150
DMP FUNCTION COMPLETED


// JOB   0000311113   00 JAN 00 00.000 HRS
// *
// *  THE FOLLOWING FUNCTIONS SETUP THE 2790 LIBRARY ON DRIVE 1.
// *  IF THIS SYSTEM DOES NOT SUPPORT 2790, PRESS CONSOLE INTERRUPT WITH
// *  SENSE SWITCH 7 UP, OTHERWISE PRESS START.
// *
// PAUSE   00 JAN 00 00.000 HRS
```

95. If your system does not support the 2790 system, set sense switch 7 ON and press CONSOLE INTERRUPT. Go to step 97.

96. If your system is to support the 2790 system, the A2790 file will be defined on drive 1. Press START to continue. The following is printed:

```
CONTINUE
// DMP    00 JAN 00 00.000 HRS
*DFILE    1          A2790 950
WILL RESERVE AT SCTR ADDR    1190
DMP FUNCTION COMPLETED
*MACRO UPDATE
SELECT
BUILD 'A2790'
JOIN  'X2790'                UPDATE COMPLETED
ENDUP


// JOB    0000311113    00 JAN 00 00.000 HRS
// *  THE FOLLOWING FUNCTIONS SETUP THE BSCIO FILE ON DRIVE 1.
// *  IF THIS SYSTEM DOES NOT SUPPORT COMM ADAPTER PRESS CONSOLE INTERRUPT
// *  WITH SENSE SWITCH 7 UP, OTHERWISE PRESS START.
// *
// PAUSE    00 JAN 00 00.000 HRS
```

97. If your system does not support any communications adapters, set sense switch 7 ON and press CONSOLE INTERRUPT. Go to step 100.

98. If your system supports any communications adapters, the BSCIO file will be set up on drive 1. Press START to continue. The following is printed:

```
CONTINUE
// DMP    00 JAN 00 00.000 HRS
*DFILE    1          BSCIO 250
WILL RESERVE AT SCTR ADDR    1546
DMP FUNCTION COMPLETED
// *
// *  AT PAUSE PLACE BSCIO SYSGEN EQUATES AFTER *SRFLE CARD, INCLUDE A SRFLE
// *  END CARD AT THE END OF TH EQUATES.  PRESS READER AND CONSOLE START.
// *
// PAUSE    00 JAN 00 00.000 HRS
```

99. Insert the BSCIO equate cards behind the *SRFLE card in the 1442. Ready the 1442 and press console START. The following is printed:

```
CONTINUE
// DMP    00 JAN 00 00.000 HRS
*SRFLE M             BSCCD BSCIO
003299 CARDS ON FILE
000239 SECTORS USED
000011 SECTORS AVAILABLE
000005 CARDS READ
DMP FUNCTION COMPLETED

// JOB    0000311113    00 JAN 00 00.000 HRS
// *  REMOVE PID PACK 3 FROM DRIVE 0 AND SAVE.
// *  MOUNT COPY OF PID PACK 1 ON DRIVE 0.
// *
// *  WHEN THE DRIVE IS READY PRESS STOP, RESET AND START.
// *  THEN WITH SENSE SWITCH 1 UP, PRESS START.
// *
// PAUSE    00 JAN 00 00.000 HRS
```

100. When drive 0 is READY, press STOP, RESET, and START. Set sense switch 1 ON and press START. The following is printed:

```
// JOB   0000111113   00 JAN 00 00.000 HRS
// *  THE FOLLOWING FUNCTIONS ASSEMBLE AND STORE BSCIO.
// *  IF THIS SYSTEM DOES NOT SUPPORT COMM ADAPTER PRESS CONSOLE INTERRUPT
// *  WITH SENSE SWITCH 7 UP, OTHERWISE PRESS START.
// *
// PAUSE   00 JAN 00 00.000 HRS
```


101. If your system supports any communications adapters, press START to assemble and store BSCIO.

101a. If your system does not support communications adapters, set sense switch 7 ON, press CONSOLE INTERRUPT, and go to step 102.

```
CONTINUE
// ASM BSCIO BSCIO   00 JAN 00 00.000 HRS
*RE-ENTRANT                                                    BSC00010
*PUNCH                                                         BSC00020

000 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.


BSCIO
DMP FUNCTION COMPLETED
*STORE            0 BSCIO
BSCIO
DMP FUNCTION COMPLETED
```

102. The printout continues:

```
// JOB   00001   00 JAN 00 00.000 HRS
// *
// *  THE FOLLOWING SOURCE FILES AND UTILITIES ARE DELETED
// * DELETE-- EQUBM, EQUEX, EQULB, GENCD, CSPCD, LSPCD, MDFCD, MFICD, EXDCD
// * BLIST, BDWAP, SYSLD, BDUPL, BDPAT, BDPIP, BDCRL, ADRCD, DMPSC, SAMP1
// * IF ALL PRECEDING OPERATIONS COMPLETED SUCCESSFULLY,
// * PRESS CONSOLE START TO CONTINUE.
// *
// PAUSE      PRESS START TO CONTINUE   00 JAN 00 00.000 HRS
```

If your system generation has not proceeded successfully to this point, don't continue. Correct the error and go back to the appropriate step to continue the generation process.

If your system generation has proceeded successfully to this point, press console START. This will cause the deletion of system-generation source files on the disk and bring you to the end of phase 2.

```
CONTINUE
// DMP    00 JAN 00 00.000 HRS
*DELET     D          EQUBM
DMP FUNCTION COMPLETED
*DELET     D          EQUEX
DMP FUNCTION COMPLETED
*DELET     D          EQULB                    LIBRARY EQUATES
DMP FUNCTION COMPLETED
*DELET     D          GENCD
DMP FUNCTION COMPLETED
*DELET     D          CSPCD                    CSPAR SOURCE
DMP FUNCTION COMPLETED
*DELET     D          LSPCD                    LSPCL SOURCE
DMP FUNCTION COMPLETED
*DELET     D          MDFCD                    MDFIO SOURCE
DMP FUNCTION COMPLETED
*DELET     D          MFICD                    MFIO SOURCE
DMP FUNCTION COMPLETED
*DELET     D          EXDCD                    EXECUTIVE DIRECTOR SOURCE
DMP FUNCTION COMPLETED
*DELET                BLIST
DMP FUNCTION COMPLETED
*DELET                BDWAP
DMP FUNCTION COMPLETED
*DELET                SYSLD
DMP FUNCTION COMPLETED
*DELET                BDUPL
DMP FUNCTION COMPLETED
*DELET                BDPAT
DMP FUNCTION COMPLETED
*DELET                BDPIP
DMP FUNCTION COMPLETED
*DELET                BDCRL
DMP FUNCTION COMPLETED
*DELET                BDUMP
DMP FUNCTION COMPLETED
*DELET                BRELD
DMP FUNCTION COMPLETED
*DELET     D          ADRCD                    ADRCK SOURCE
DMP FUNCTION COMPLETED
*DELET     D          DMPSC                    DMPS SOURCE
DMP FUNCTION COMPLETED
*DELET     D          SAMP1                    MPX SAMPLE PGM SOURCE
DMP FUNCTION COMPLETED


// JOB    00 JAN 00 00.000 HRS
// DMP    00 JAN 00 00.000 HRS
*DEFINE PAKDK 0
DMP FUNCTION COMPLETED


// JOB    00 JAN 00 00.000 HRS
// *  AT THIS POINT REFER TO OP MANUAL FOR FURTHER INSTRUCTIONS.
// END    2790 USERS SHOULD REFER TO 2790 SRL FOR BUILDING CORELOADS.
```

This completes phase 2.

Information formerly on this page has been moved to page 49.

PHASE 3 OPERATIONS

LOADING BOM

1. Remove any cards that may be in the 1442.

2. Clear main storage to zeros (refer to "I/O Device and Console Operations").

3. Place the 4-card BOM high core loader (received from PID) followed by the BOM deck (punched during phase 2, step 82) into the 1442 hopper.

4. *Ready the* 1442 and the 2311 drives.

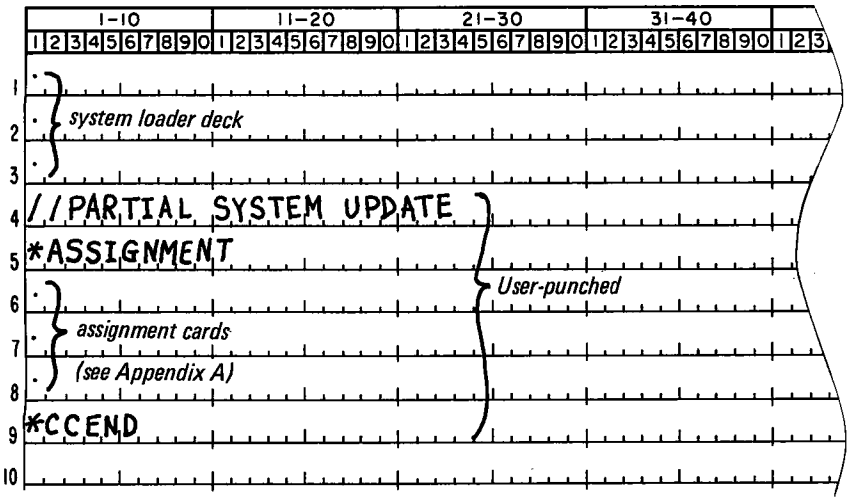
5. Press PROGRAM LOAD on the console. The following messages are printed:

```
IBM 1800 MPX/BOM XX. XXX      00 JAN 00
SEN SW 0 ON ABSOLUTE LOADER
SEN SW 1 ON LOAD BP MONITOR
SEN SW 2 ON SET CLOCK VIA DATA SWS
SEN SW 3 ON SET DATE VIA DATA SWS
```

LOADING THE ASSIGNMENT CARDS

6. Punch your assignment cards (see Appendix A) and the System Loader control cards shown below.

7. Place the System Loader deck (punched during phase 2), the system loader control cards and your completed assignment cards into the 1442 hopper in the following order:



8. Set sense switch 0 ON, data switch 15 ON and press console START. The I/O device and interrupt assignments you have selected will be printed by the System Loader.

   To insure that valid FORTRAN logical unit numbers are inserted into the IOU subroutine, IOU must be on drive 0.

This completes phase 3.

PHASE 4 OPERATIONS

In this phase you will define the number of drives in your system, label those drives and define your main-storage layout. The sequences of control cards used to carry out phase 4 operations are illustrated in Figures 2 and 3. Note that each job for each drive requires a separate // JOB and // DMP card; the // JOB statements don't contain drive labels, because the drives haven't been initialized yet. Refer to IBM 1800 Multiprogramming Executive Operating System Programmer's Guide, Order Number GC26-3720, for details on how to complete and use each of these control statements.

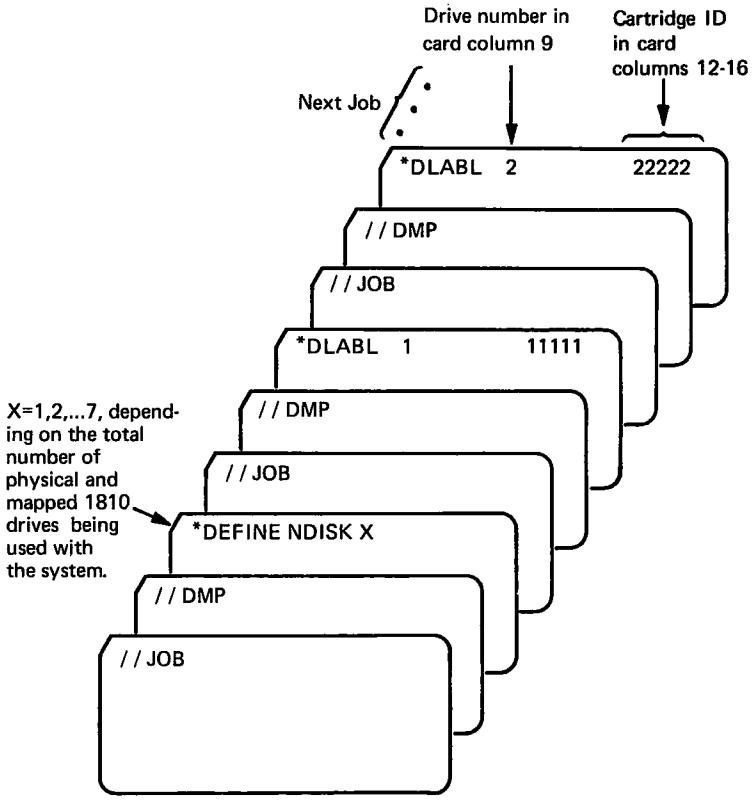


Figure 2. Define Number of Drives and Label Disk Cartridges

OPERATING PROCEDURES

1. If applicable, place the scratch packs to be labeled on their respective drives and ready those drives.

2. Place the deck of control cards to define and label disks and to define your main-storage layout in the 1442 hopper in the sequence illustrated in Figures 2 and 3.

3. Press START on the 1442. This will cause the cards to be read and processed.

This completes phase 4.

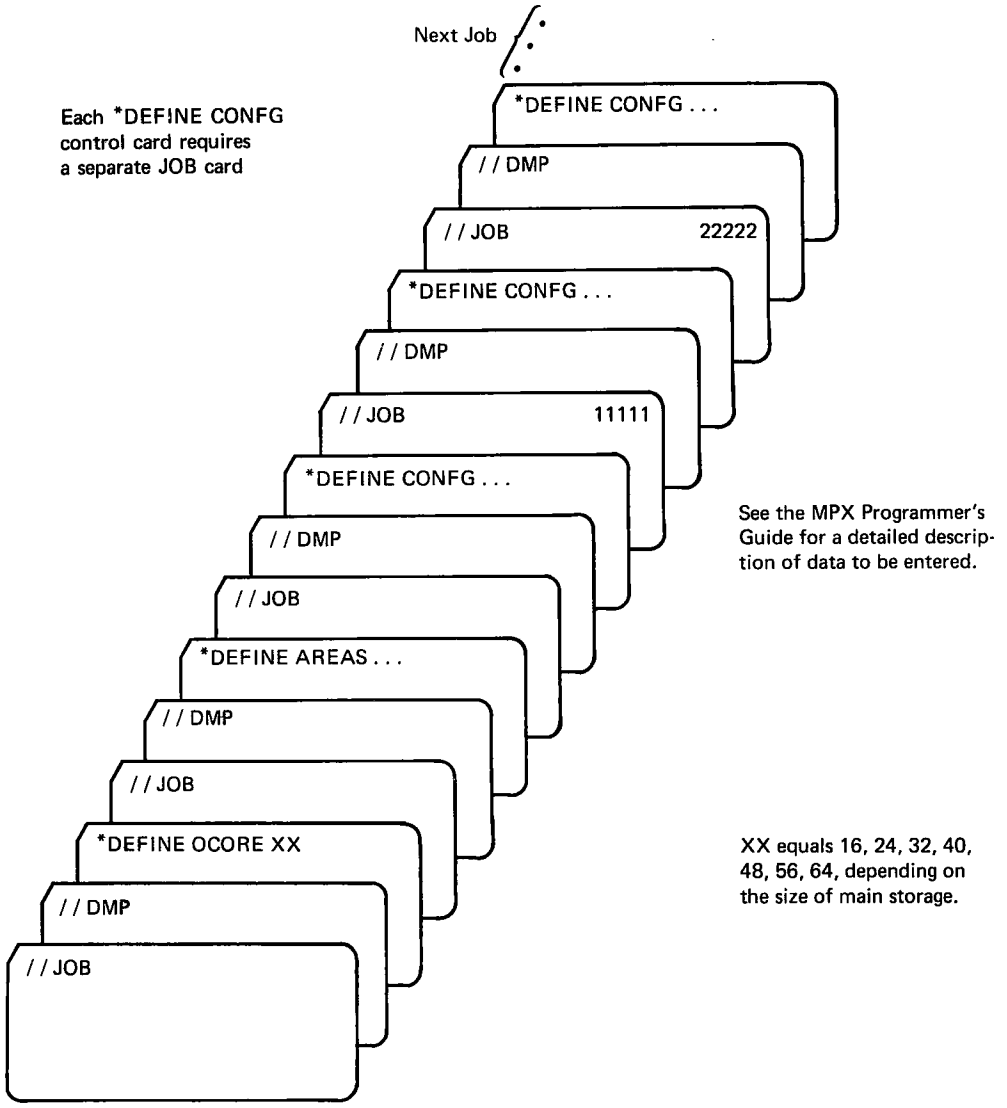


Figure 3. Define Main-Storage and Disk Layout

PHASE 5 OPERATIONS

Phase 4 operations finished with the Batch-Processing Monitor Supervisor in control. Batch-processing operations continue, but are divergent at this point, with one procedure for generating a batch-processing system and another for generating a real-time system.

BATCH-PROCESSING SYSTEM

To complete the generation of your batch-processing system, you build BOM, using the control statements and BOM object deck illustrated in Figure 4.

1. Place the control cards to build BOM and the BOM object deck in the 1442 hopper in the sequence illustrated in Figure 4.

2. Press the START on the 1442. This will cause the cards to read and processed.

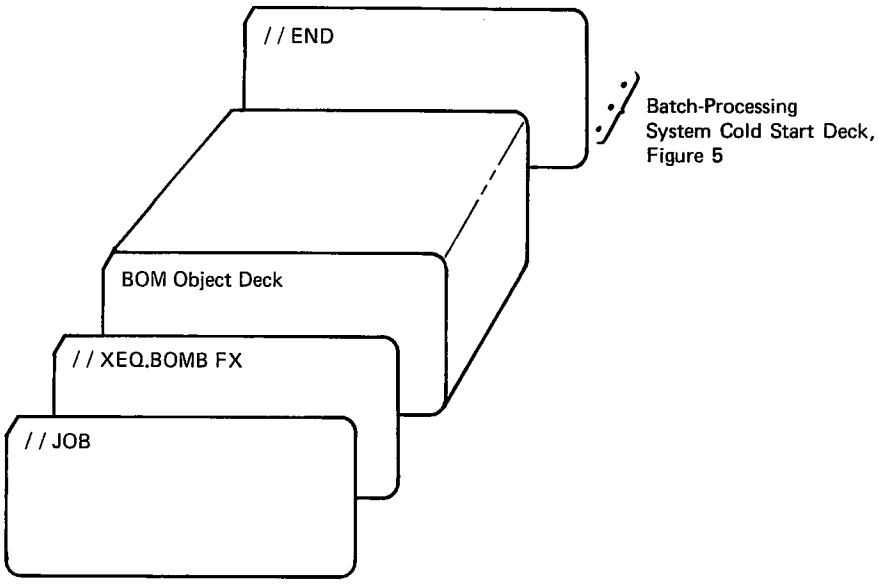


Figure 4. BOM Build Job

This completes the system-generation process for a batch-processing system. To start your system you will use the seven cold start loader cards and a cold start name card (Figure 5). For details on cold starting a system, see "Cold Start."

Before you begin batch processing, you may want to use the CE Coreload (described later under "CE Coreload Programs") to initialize system error counters.

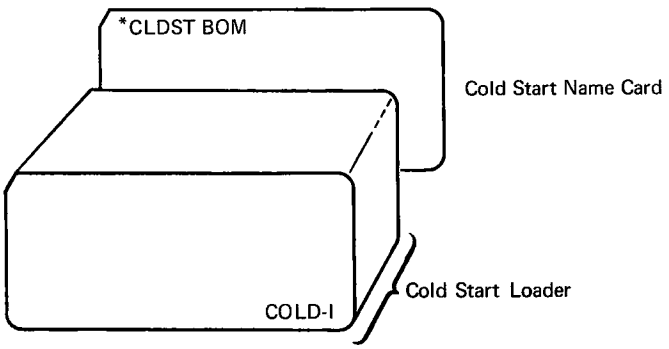


Figure 5. Batch-Processing System Cold Start Deck

REAL-TIME SYSTEM

To complete the generation of your real-time system, you assemble or compile and store any subroutines to be included in the Executive, build your Executive, and assemble or compile and build your process coreloads. The sequences of control statements necessary to complete phase 5 are illustrated in Figures 6 and 7. Refer to IBM 1800 Multiprogramming Executive Operating System Programmer's Guide, Order Number GC26-3720, for details on how to complete and use each of these control statements.

1. Place the completed control statements and source decks in the 1442 hopper in the order illustrated in Figure 6.

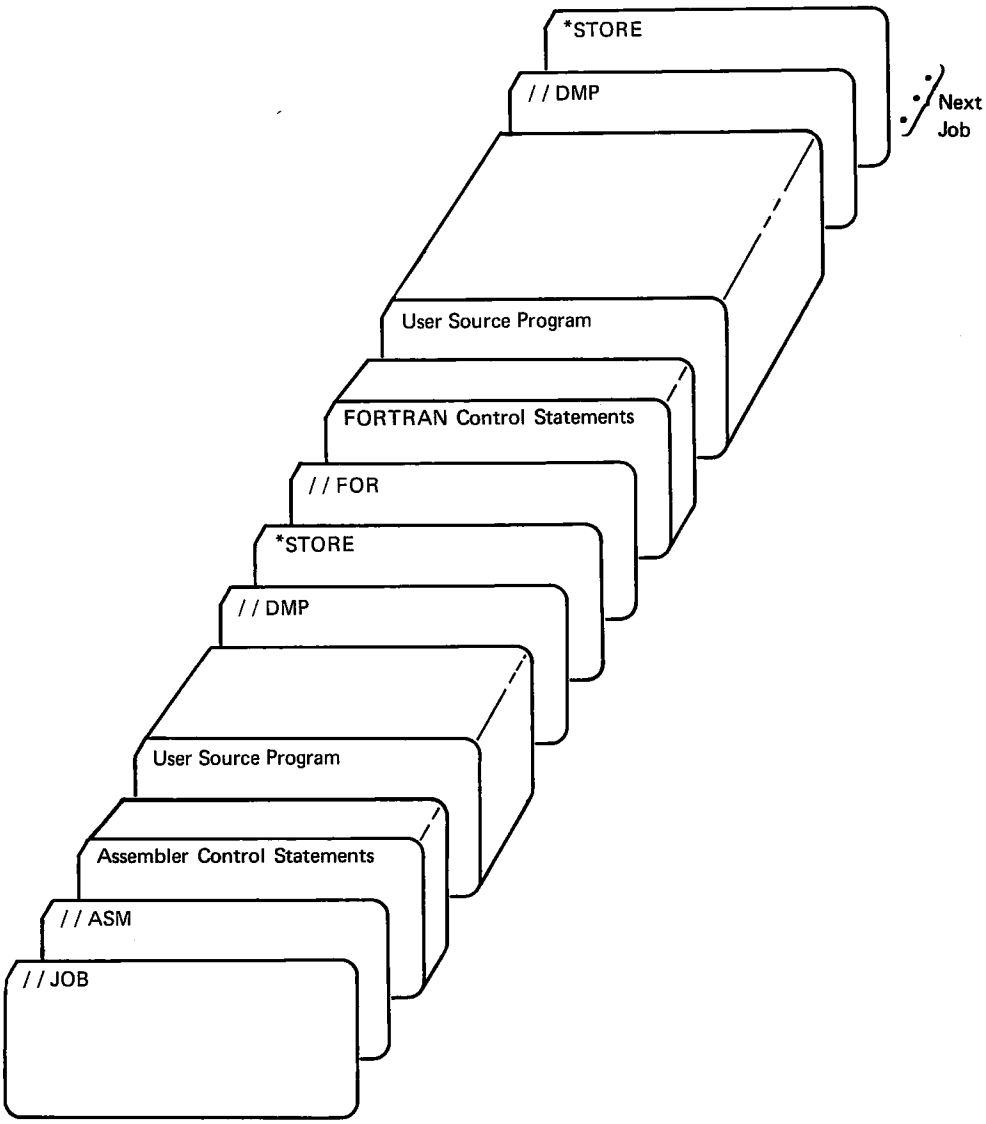


Figure 6. Assemble, Compile, and Store Executive Subroutines on Disk

2. Place the completed control statements and BOM object deck in the 1442 hopper in the order illustrated in Figure 7.

3. Press START on the 1442. This will cause the cards to be read and processed.

This completes the system-generation process for a real-time system. Remaining is the assembling or compiling and building of your process coreloads. At this point you may wish to start your real-time system or you may continue under the Batch-Processing Monitor. If you wish to continue under the Batch-Processing Monitor, skip to step 4.

To start your real-time system you must perform a cold start to the Batch-Processing Monitor (see "Cold Start"). When the cold start is complete, continue with step 4.

4. Complete the control statements necessary to assemble or compile and build your process coreloads as illustrated in Figure 8.

5. Place the completed control statements and source decks in the 1442 hopper.

6. Press START on the 1442. This will cause the cards to be read and processed.

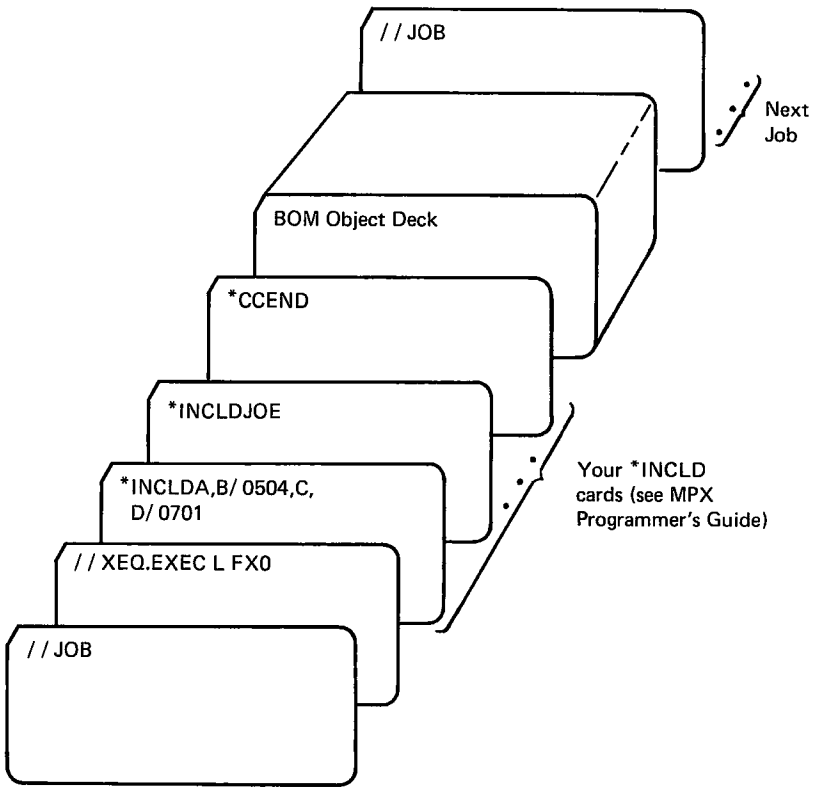


Figure 7. Building the Executive

2790 INSTALLATION

If you are installing the 2790 system, it is necessary to store to disk drive 0 the On Line 2790 Area Station Exerciser/Customer Engineer Diagnostic Program. The object card deck for this program can be obtained from your customer engineer and stored in core image format using the following job steps:

```
1-10      11-20     21-30     31-40     41-50     51-60
1234567890 1234567890 1234567890 1234567890 1234567890 1234567890
1 // JOB
2 // OMP
3 *STORE  T RD  0 @CEX@
4 .
5 .  } Object deck of 2790 Exerciser/Diagnostic
6 .
7 *STORECIX  0 @CEX@ @CEX@
8 *LOCALPCDIG,ASDIG,ERRST,DRECT,(RQD1,RQD2,CONSL)
9 *CCEND
10
```

These cards are included with the object card deck.

This completes phase 5. To start your real-time system you must cold start to either the Batch-Processing Monitor or a process coreload (see "Cold Start").

Before you begin real-time processing, you might want to use the CE Coreload (described later under "CE Coreload Programs") to initialize system error counters.

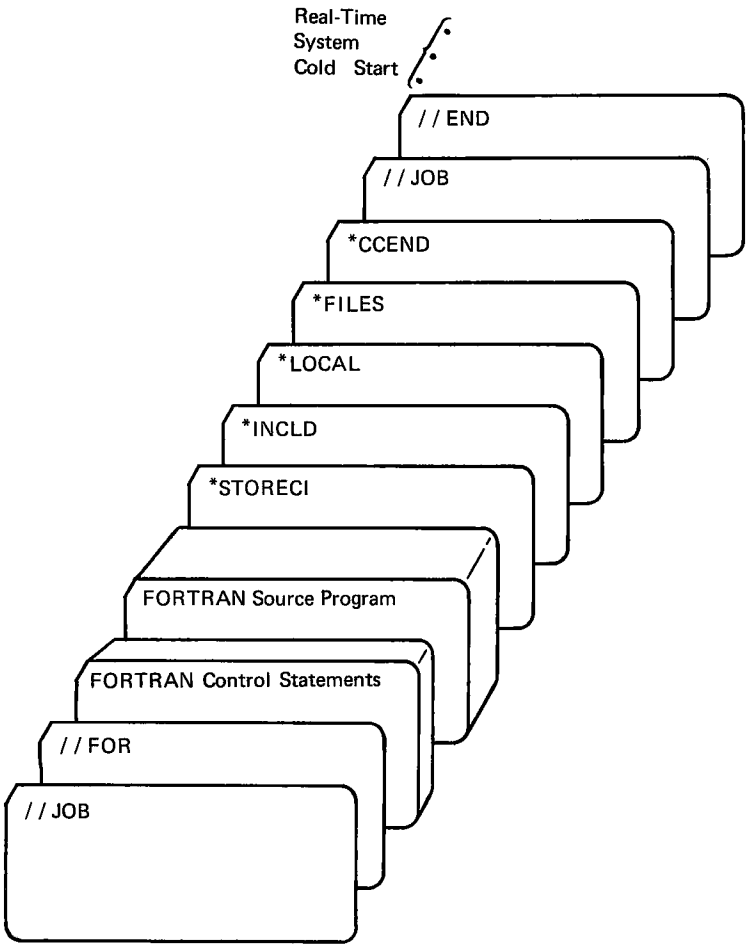


Figure 8. Compile and/or Assemble User-Written Programs and Build Process Coreloads for Execution

REBUILDING THE SYSTEM EXECUTIVE

SYSTEM EXECUTIVE REBUILD CONDITIONS

While relocatable programs can be deleted and replaced on-line by the Batch Processing Monitor, it is not possible to modify any features of the System Executive on-line. Changes in the Executive area (including Executive interrupt routines) thus require an off-line Executive rebuild.

The MPX System Executive may be rebuilt at any time by following the detailed operating procedures specified for an initial Executive build, which is discussed elsewhere in the MPX Operating Procedures manual.

Since INSKEL COMMON is not open-ended, the user may face the difficulty of adding to it after it is defined. It is recommended that an extra area be reserved in INSKEL COMMON to allow for programming contingencies. See Core Load Rebuild Conditions.

The Executive Branch Table (EBT) has already been described. This table enables the user to rebuild the System Executive when modifying subroutines, changing the logic flow or adding patches to the Executive Director and BOM. After such rebuilding, addresses in core loads will still reference a fixed address in the EBT. An ability to shift the entry points of subroutines within the Executive is thus available without the necessity of rebuilding the referencing core loads.

If, however, the entry points within these tables no longer pointed to the same subroutine, all core loads must be rebuilt (see Core Load Rebuild Conditions).

When the Executive is initially built, the entry points to the in-core-with-Executive (ICI) routines that are to be serviced by a level and bit (LLBB) designation are placed in the Interrupt Core Load Table (ICLT), and all other entry spaces are zeroed out. Subsequently, when a core load is built and queued to an LLBB designation (out-of-core interrupt servicing), an entry is made into the ICLT in the corresponding LLBB position which indicates the location on disk where the out-of-core interrupt servicing core load resides.

However, if it is desirable to rebuild the Executive, the information reflected in the ICLT for out-of-core interrupts can be preserved. This is accomplished by rebuilding the Executive with the SAVE ICLT OPTION (by placing an S in column 20 of the //XEQ . EXEC control card). This effectively copies the word count and sector address for each out-of-core interrupt from the old ICLT into a corresponding LLBB position of the new (fresh) ICLT, providing a new ICI was not included in the new Executive for this LLBB position.

CORE LOAD REBUILD CONDITIONS

In an Executive rebuild, it may not be necessary to rebuild those core loads built under the previous (that is, old) Executive if the following conditions are met:

1. No previously included CALL, LIBF, ICI, or ISS type subroutines may be removed from the Executive. However, additional CALL or ISS type subroutines may be added as dictated under condition 5.
2. All previously included subroutines may be reassembled or recompiled in preparation for an Executive rebuild, provided the names and the number of entry points for these subroutines are not modified between inclusions.
3. No additional LIBF, ICI, or ISS type subroutines (types 3, 4, and 5 respectively) may be included in the new Executive.

   CALL or ISS type subroutines (types 4 and 6) may, however, be added provided that Patch Area in the Executive is large enough to contain these additions.
4. If BOM or the Executive Director is reassembled, the number of interrupt levels used, the length of INSKEL COMMON, and the length of the System Executive Area should not be altered.

   Note also that a change in the size of INSKEL COMMON implies a reassembly of BOM, and a resulting alteration in the high-address origin of INSKEL COMMON. Any new core load built at this time will therefore be incompatible with previously-built core loads using INSKEL COMMON. In addition, if INSKEL COMMON were increased in size, old core loads would remain compatible with each other in their use of this common area. However, if the size of INSKEL COMMON were decreased, previously built core loads using INSKEL COMMON would be invalidated.

5. If there are to be additions to user-included subroutines in the Executive as permitted under condition 3, the order of *INCLD control cards in the Executive rebuild process is critical.

   In the rebuild operation, three sets of *INCLD control cards A, B, and C are required in that order.

   Set A must correspond to the *INCLD control cards used for the previous Executive build.

   Set B must be made up by the user and contain the names of the Builder-included subroutines pertinent to the previous Executive build. These names appear in the Executive Core Map (of the previous Executive build), and must be specified on *INCLD control cards in this exact order.

   Set C must contain the names of the new user-included subroutines.

   See Example of Executive Build and Rebuild.

6. Following the rebuild process, a visual comparison of the previous Executive and the new Executive core maps must show identical entry points for those LIBF and CALL map entries common to both executives.

EXAMPLE OF INITIAL EXECUTIVE BUILD AND EXECUTIVE REBUILD

Program Listing 16 illustrates the general sequence of control cards, the Executive Core Map, the Interrupt Core Load Table (ICLT), and the Executive Tables for a typical initial executive build and executive rebuild. A separate interpretation of the Executive Core Map and the ICLT is given at the end of this section.

In the initial Executive:

FADD is an IBM MPX Library subroutine included in the Executive for the purpose of being shared by various core areas.

CALLA and CALLB are two subroutines included in the Executive for the purpose of servicing process interrupts on level 10 bit 2 and level 10 bit 10 respectively.

Note that these two assignments are printed in the ICLT map.

CALLC is a special user-written CALL-type subroutine included in the Executive for the purpose of being shared by various core load areas.

In the rebuilt Executive:

No major modifications are implemented. Neither the Executive I/O nor INSKEL COMMON are altered.

The interrupt servicing subroutines CALLA and CALLB have been reassigned to service level 14 bit 1 and level 2 bit 2 respectively. These pertain to set A *INCLD control cards.

Set B *INCLD control cards contains those Builder-included subroutines which were implicitly added in the previous build.

Set C *INCLD control cards gives the additional subroutines included by the user. These are:

ISSA - a user-included subroutine for the express purpose of servicing an RPQ device interrupt.
CALLD - a user-included CALL-type subroutine to be used by the various core load areas.

You should be aware that in rebuilding the Executive, the set A *INCLD control cards must be in exact order-correspondence with the previous Executive build, so that these routines will be loaded in the same order-sequence, and entry points in the Executive Branch Table and Executive Transfer Vector remain valid.

Program Listing No. 16

Initial Executive Build

```
// JOB          A 00.000 HRS


// XEQ .EXEC L FXO
*INCLDFADD
*INCLDCALLA/1002
*INCLDCALLB/1010
*INCLDCALLC
*CCEND


MPX, BUILD .EXEC

EXEC CORE  MAP
TYPE NAME  ARG1  ARG2

CALL BULKN 0644  007C
CALL TYPEN 0AA4  00B9
CALL WRTYN 0AA4  00B9
CALL LINK  380C  008E
CALL EXIT  35D8  00B6
CALL PRNTN 1143  00BA
CALL RSAVE 14F5  00C7
CALL ABORT 17FF  0039
CALL EACRL 17CE  00C6
CALL EACPT 1C92  0099
CALL GETQ  1797  00FC
CALL PUTQ  1739  00FB
CALL IOTST 14D1  0062
CALL IOSET 148B  0063
CALL IOEXT 1359  0076
CALL IOSAV 138C  0075
CALL QZEXT 1414  009B
CALL QZSAV 13F5  009A
CALL TVEXT 1303  00AD
CALL TVSAV 13B3  00AC
COMM       01AE  0275
ORG  EXDIR 26D6
CLNT  EXEC       3E3E
ICI  CALLA 3C44  0A02
ICI  CALLB 3C46  0A0A
LIBF FADD  3C68  3E5E  R
CALL CALLC 3CE6  3E76
CALL SPECL 3820  3E75  R
CALL BACK  3847  3E74  R
CALL QLEVL 39A7  3E73  R
LIBF FSBR  3C48  3E61  R
LIBF FSBRX 3C4D  3E64  R
LIBF FSUB  3C5C  3E67  R
LIBF FADDX 3C62  3E6A  R
LIBF FSUBX 3C57  3E6D  R
LIBR FARC  3CE8  3E70  R
PTCH       3D1C  3E3B


  ICL TABLE MAP
LLBB  WC/EP  SA   ICLT

0A02  3C44        28E6
0A0A  3C46        28F6
```

User Included

Builder Included

Executive Tables

```
3E00 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

3E30 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 059C 4259 0004 0000  <-CLNT
3E40 0000 0000 4480 00B9 0000 0000 0000 0000 0000 0000 4480 00BA 0000 0000 0000 0000  <-FIO
3E50 0000 0000 0000 0000 7005 0000 4480 00B9 0000 0000 0000 0000 7005 0000 0000 4480  <-ETV
3E60 3E7F 0000 4480 3E7E 0000 4480 3E7D 0000 4480 3E7C 0000 4480 3E7B 0000 4480 3E7A
3E70 0000 4480 3E79 39A7 3847 3820 3CE6 3C46 3C44 3CE8 3C57 3C62 3C5C 3C4D 3C48 3C68  <-EBT
3E80 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

                     CALL                ICI                 LIBF
```

Executive Rebuild

```
// JOB      A 00.099 HRS

// XEQ .EXEC L FXO
*INCLDFADD
*INCLDCALLA/1401
*INCLDCALLB/0202
*INCLDCALLC
*INCLDSPECL,BACK,QLEVEL,FSBR,FSBRX,FSUB
*INCLDFADDX,FSUBX,FARC
*INCLDISSA
*INCLDCALLD
*CCEND

MPX, BUILD .EXEC

EXEC CORE  MAP
TYPE NAME  ARG1  ARG2

CALL BULKN 0644  007C
CALL TYPEN 0AA4  00B9
CALL WRTYN 0AA4  00B9
CALL LINK  380C  008E
CALL EXIT  35DB  00B6
CALL PRNTN 1143  00BA
CALL RSAVE 14F5  00C7
CALL ABORT 17FF  0039
CALL EACRL 17CE  00C6
CALL EACPT 1C92  0099
CALL GETQ  1797  00FC
CALL PUTQ  1739  00FB
CALL IOTST 14D1  0062
CALL IOSET 148B  0063
CALL IOEXT 1359  0076
CALL IOSAV 138C  0075
CALL QZEXT 1414  009B
CALL QZSAV 13F5  009A
CALL TVEXT 13D3  00AD
CALL TVSAV 138E  00AC
COMM       01AE  0275
ORG  EXDIR 26D6
CLNT .EXEC       3E3C
ICI  CALLA 3C44  0E01
ICI  CALLB 3C46  0202
LIBF FADD  3C68  3E5C  R
CALL CALLC 3CE6  3E76  R
CALL SPECL 3820  3E75  R
CALL BACK  3847  3E74  R
CALL QLEVL 39A7  3E73  R
LIBF FSBR  3C48  3E5F  R
LIBF FSBRX 3C4D  3E62  R
LIBF FSUB  3C5C  3E65  R
LIBF FADDX 3C62  3E68  R
LIBF FSUBX 3C57  3E6B  R
LIBF FARC  3CE8  3E6E  R
CALL ISSA  3C1E  3E72
CALL CALLD 3D24  3E71
PTCH       3D26  3E39

  ICL TABLE MAP
LLBB  WC/EP  SA    ICLT

0202  3C46        282E
0E01  3C44        2954
```

Set A, Set B, Set C, User Included, Builder Included, Additional User Included

Executive Tables

```
3E00 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
                                                                                  CLNT
3E30 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 059C 4259 0004 0000 0000 0000
3E40 4480 00B9 0000 0000 0000 0000 0000 0000 4480 00BA 0000 0000 0000 0000 0000 0000  FIO
3E50 0000 0000 7005 0000 4480 00B9 0000 0000 0000 0000 7005 0000 0000 4480 3E7F 0000
3E60 4480 3E7E 0000 4480 3E7D 0000 4480 3E7C 0000 4480 3E7B 0000 4480 3E7A 0000 4480  ETV
3E70 3E79 3D24 3D1E 39A7 3847 3820 3CE6 3C46 3C44 3CE8 3C57 3C62 3C5C 3C4D 3C48 3C68  EBT
3E80 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
          ISS & CALL                ICI                LIBF
```

CLNT, FIO, ETV, EBT

INTERPRETATION OF THE EXECUTIVE CORE MAP AND THE INTERRUPT CORE LOAD TABLE (ICLT) MAP

EXECUTIVE CORE MAP

The Executive Build function always prints a map of the assembled Executive in the following format:


```
EXEC   CORE   MAP            (Page Heading)
TYPE   NAME   ARG1   ARG2    (Column Heading)
```

Type indicates the map entry type (for example, LIBF, CALL, and CLNT). Up to five alphameric characters are permitted under NAME to describe a subroutine, control program, etc. (for example, BULKN, EXDIR).

ARG1 and ARG2 may contain either a four-digit hexadecimal number or a blank field.

Core Load Name Table (CLNT)

```
CLNT   NNNNN   XXXX   YYYY
```

The word count and disk address of the core load named NNNNN, which is referenced within the Executive, are assigned to locations YYYY and YYYY+1 of the Executive Core Load Name Table. The XXXX field is blank for all CLNT entires.

Executive Director Origin (ORG)

```
ORG    EXDIR   XXXX   YYYY
```

XXXX is the hexadecimal address of the Executive Director: More precisely, it is the beginning of the ICLT pointer block. YYYY is always blank.

In-Core-With-Executive Interrupt Routines (ICI)

```
ICI    NNNNN   XXXX   LLBB
```

The entry point to the in-core-with-Executive interrupt servicing routine named NNNNN is at absolute location XXXX. LL designates the interrupt level (ILSW) while BB designates the interrupt level (ILSW) while BB designates the bit position within the PISW for that associated level.

Library Function Subroutines (LIBF)

```
LIBF   NNNNN   XXXX   YYYY
```

The LIBF-type subroutine entry point named NNNNN is at absolute location XXXX of the Executive. The corresponding three-word transfer vector entry point will be at the location YYYY in the ETV Table in the System Executive.

Call-type Subroutines (CALL)

```
CALL   NNNNN   XXXX   YYYY
```

The Call-type subroutine entry point named NNNNN is at absolute location XXXX of the Executive. The indirect entry point is at location YYYY of the System Executive Branch Table (EBT).

INSKEL COMMON (COMM)

```
COMM   XXXX   YYYY
```

The low core storage boundary of INSKEL COMMON is at absolute location XXXX of the Executive. The high boundary is at location YYYY.

Patch Area (PTCH)

```
PTCH   XXXX   YYYY
```

The Patch Area (that is, unused core locations) extends from XXXX, the absolute location of the Executive, through location YYYY.

INTERRUPT CORE LOAD TABLE (ICLT) MAP

The ICL Table map is printed to reveal any interrupt assignments made in the Executive ICLT. Its format is as follows:

```
ICL     TABLE   MAP
LLBB    WC/EP   SA      ICLT (Column Heading)
```

The interrupt level and bit assignments are indicated by a four-digit hexadecimal number under LLBB. The two high-order digits contain the level; the two low-order digits represent the bit assignment.

If the entry is an in-core-with-Executive routine, the WC/EP column will contain the hexadecimal entry point to this routine. This address corresponds to the ARG1 address (that is, XXXX) of ICI entries. The SA field will be blank. The ICL Table absolute core location in which the entry point is placed is indicated in the ICLT column.

When rebuilding the Executive with the SAVE ICL TABLE option, word counts and sector addresses of any interrupt core loads are retained from the old ICLT. Their interrupt assignments are indicated in the LLBB column, while the WC/EP and SA columns will contain their word counts and disk addresses. The corresponding ICLT absolute core location is found in the ICLT column.

Cold Start

Cold Start is an MPX system program. Its main function is to start your system running. There are three ways you can use the Cold Start Program: to give control to BOM, to give control to the Batch-Processing Monitor, or to give control to one of your coreloads.

Cold Starting a Batch-Processing System

Once the system-generation procedure has been completed and your batch-processing MPX system has been built, cold start can be used to set up and start that system. First Cold Start loads BOM from disk to main storage. It then establishes the logical numbers of 1810s, 2311s, and communications adapters specified on the cold start name card (described later in this chapter). Cold Start then branches to an entry point in BOM for the completion of the initialization process. BOM storage protects certain areas of itself, brings specified disks on line, starts Timer C, and prints the following Batch-Processing Monitor initialization messages:

```
IBM 1800 DACS  MPX/BOM   00.000  00 JAN 00
SEN SW 0 ON ABSOLUTE LOADER
SEN SW 1 ON LOAD BP MONITOR
SEN SW 2 ON SET CLOCK VIA DATA SWS
SEN SW 3 ON SET DATE VIA DATA SWS
```

Once the above messages have been printed, the cold start operation is complete and the Batch-Processing Monitor is in control.

Cold Starting a Real-Time System

Once the system-generation procedure has been completed and your real-time system has been built, Cold Start is used to set up and start that system. Cold Start begins by loading the Executive from disk into main storage. It then establishes the logical numbers for 1810s, 2311s, and communications adapters specified on the cold start name card (described later in this chapter). Certain areas of the Executive are then storage protected, specified disks are brought on line, and reload information (if specified) is set up in the Executive. Cold Start then requests time, date and data switch settings by the following messages:

```
ENTER TIME THROUGH DATA SWITCHES

TIME ENTERED WAS 00.533 HOURS

ENTER DATE THROUGH DATA SWITCHES
DATE ENTERED WAS   FEB   04  1970
SET VALUE IN DATA SWITCHES IF REQUIRED BY COLD START CORE LOAD
```

The information you set is placed in the Executive by Cold Start. Timer C is started. The cold start coreload is initiated and control is passed to the Executive. If no cold start coreload is named, background processing begins.

USING A COLD START CORELOAD

Cold starting by giving control to one of your coreloads can be performed only under a real-time system. The functions of Cold Start then are identical to those for other cold starts of a real-time system except that before calling EXIT, Cold Start queues the coreload (called the Cold Start coreload). The first operation of the real-time system will be to execute that coreload.

Cold Start Card Formats

COLD START LOADER CARDS

The seven cold start loader cards are provided by IBM along with the system-generation PID disk(s). The Loader cards constitute a program which will load the Cold Start program from disk (the last two cylinders of a physical 1810 drive or the last four tracks of a mapped 1810 drive) into main storage and branch to its entry point. (See Figure 9 for the formats of the seven cold start loader cards.)

COLD START NAME CARD

You must punch the cold start name card. Its format is shown below. It is from the cold start name card that Cold Start gets the information necessary to start a system.

Cold Start Name Card Format

Card Column	Description of Contents
1-6	*CLDST
7	Blank
8-12	One of the following (left-justified): • BOM, for cold starting a batch-processing system • BPMON, for cold starting a real-time system doing background processing • NAME, the name (up to five characters long) of the cold start coreload
13,15,16,17,19, 21,23,25,27,29, 31,33	Blank
18,20,22,24,26, 28,30,32	Numbers that specify the assignment of logical drive numbers to 1810 drives. The physical drive numbers, if any, entered in these columns become logical 1810 drives 0-7, respectively. If the drive is a mapped 1810, then enter the identifier, x, in the name of the mapped drive (BULKx). Only the numbers 0-7 may be used. Note that this assignment specifies the drive to be on the system; you cannot use the drive if you do not specify it here.

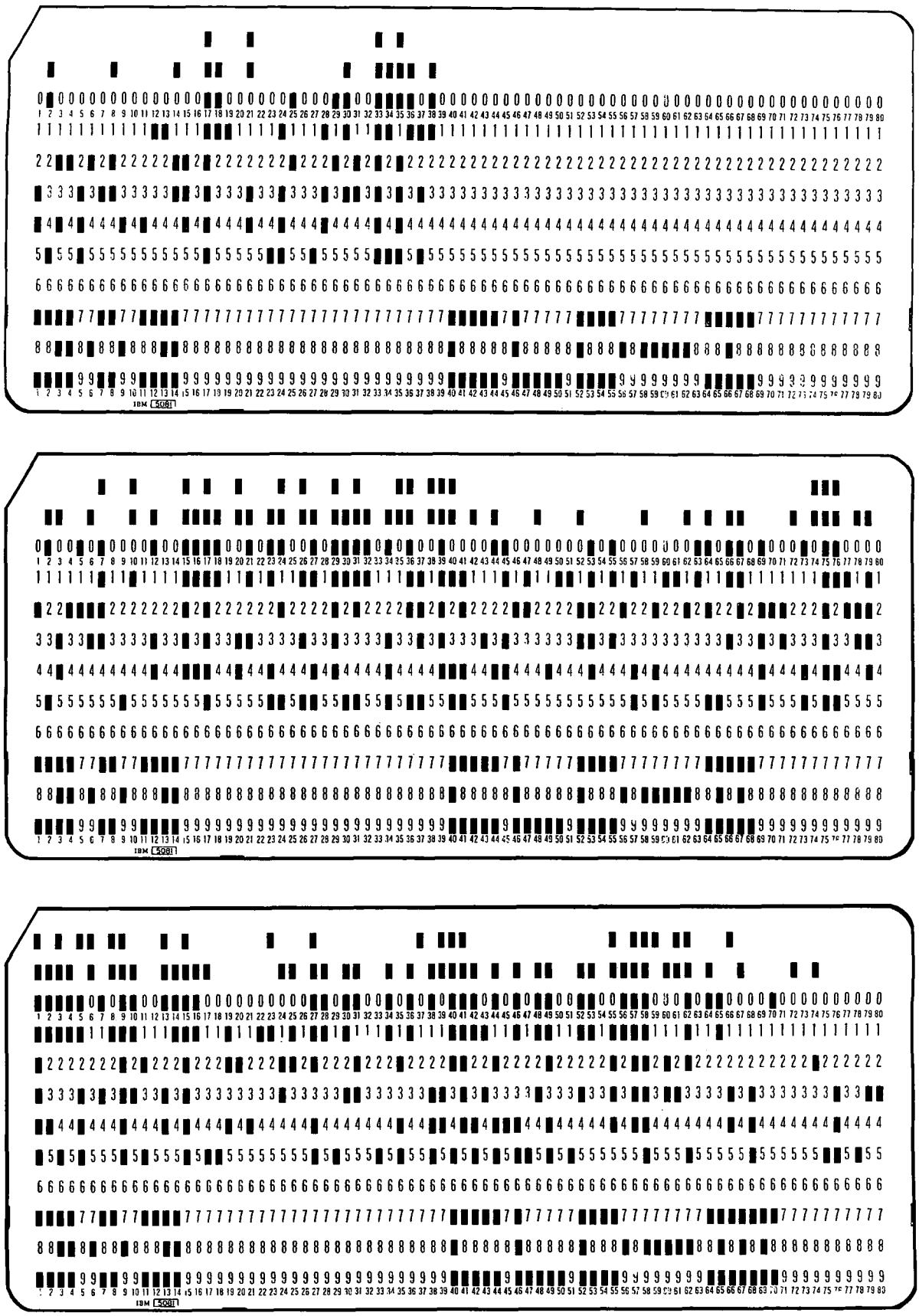


Figure 9. Cold Start Loader Cards

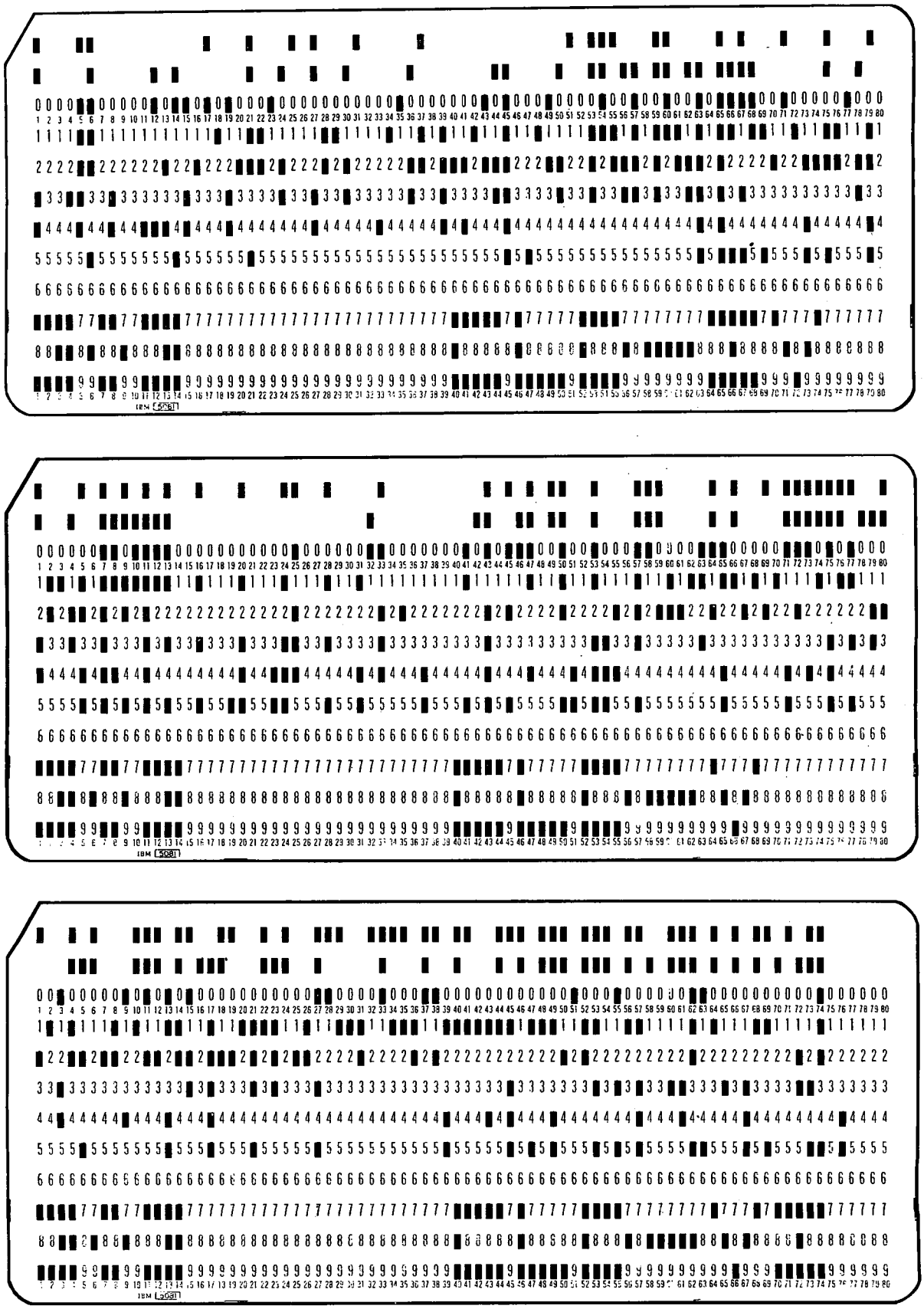


Figure 9. Cold Start Loader Cards (Cont.)

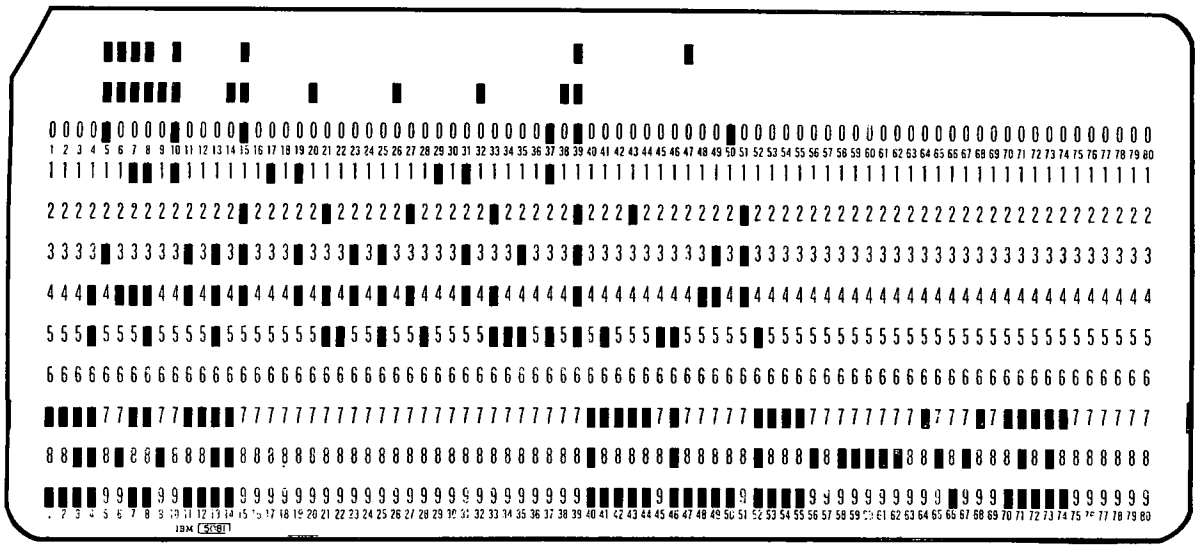


Figure 9. Cold Start Loader Cards (Cont.)

This page intentionally left blank.

Card Column	Description of Contents
34-38	The name of the coreload to be called in the event of a system reload (left-justified). The reload coreload must be built to execute in VCORE. If these columns are blank, the system programs an interrupt to the level and bit named in the LDLEV and LDBIT equates.
39	Blank
40-43	Level and partition numbers of the cold start coreload (LLAA), in decimal. Blanks are assumed to be zeros; zeros denote the basic level and VCORE.
44-47	A four-digit hexadecimal number used, bit-by-bit, to indicate the mask status of interrupt levels 0-13 for the cold start coreload (the first hexadecimal digit corresponds to levels 0-3, the second to levels 4-7, and so on; a 1 bit indicates a level is masked). If this entry is blank, all levels are unmasked. For example, an entry of /A014 would mean levels 0, 2, 11 and 13 were to be masked, and all others unmasked. The disk interrupt level must be unmasked.
48-51	Same as columns 44-47 except that this field indicates the mask status for interrupt levels 14-23 for the cold start coreload. These columns are meaningful only if the NULEV equate indicates 14 or more levels.
52	Blank
53-56	Same as columns 44-47 except that this field indicates the mask status of interrupt levels 0-13 for the reload coreload.
57-60	Same as columns 44-47 except that this field indicates the mask status of interrupt levels 14-23 for the reload coreload. These columns are meaningful only if the NULEV equate indicates 14 or more levels.
61	Blank
62, 63, 64, 65, 66, 67, 68, 69	Numbers that specify the assignment of logical drive numbers to your 2311 drives. The physical drive numbers, if any, entered in these columns become logical 2311 drives 0-7, respectively. The physical drive number entered must have been previously defined (by BOM equate cards) as part of the system.
70	Blank
71, 72, 73, 74 75, 76, 77, 78	Numbers that specify the assignment of logical line numbers for your communications adapters. The physical line numbers entered in these columns become logical lines 0-7, respectively. Each referenced physical line number must have been previously defined (by BOM equate cards) as part of the system. If physical lines were defined in BOM and they are to be used, these columns must be set to define the line numbers.
79	Blank
80	If any 2311 disk drives are used, this column must contain the hexadecimal address of the 2841 control unit, in hexadecimal. Valid entries are 0-F.

Operating Procedures

To cold start the MPX system, perform the following steps:

1. Turn ON the WRITE STG PRT BITS switch, turn OFF the CHECK STOP switch and set the console mode switch to RUN.

2. Clear main storage to zeros.

3. If the Cold Start Program resides on a physical 1810 drive, set data switch 0 OFF and enter the physical 1810 drive number in data switches 1-15 as follows:

   All data switches OFF means drive 0
   Data switch 15 ON means drive 1
   Data switch 14 ON means drive 2

   If the Cold Start Program resides on a mapped 1810 (on a 2311 drive),

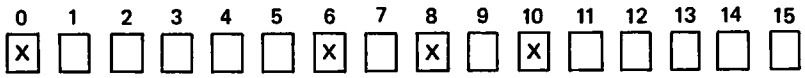
   a. Set data switch 0 ON.

   b. Enter the 1810 drive identifier in data switches 4-7 (the identifier is the 1810 number x in BULKx, the name used when the mapped 1810 drive was created).

   c. Enter the 2841 control unit address in data switches 8-11.

   d. Enter the physical drive number of the 2311 in data switches 13-15.

For example, data switches set to



(where x means ON) specifies that:

Cold Start resides on a 2311 disk drive
The mapped 1810 drive number is 2
The 2841 Control Unit address is A
The physical drive number of the 2311 is 0

4. **Place the seven cold start loader cards followed by the cold start name card and a blank card (or the following job) in the 1442 card read punch hopper and ready the reader.** The 2311 drives must be ready before pressing program load.

5. Press console RESET and PROGRAM LOAD

   a. If this is a batch-processing system cold start, BOM is read into main storage from disk and the operation is complete when the BOM initialization messages are printed on the 1053.

   b. If this is a real-time system cold start, the Executive is read into main storage from disk and the following message is printed on the 1053:

```
ENTER TIME THROUGH DATA SWITCHES
```

6. Enter the time in hours and minutes (HHMM) in the data switches and press START. After Cold Start reads the time from the data switches, it prints the time in hours and thousandths of an hour.

```
TIME ENTERED WAS 00.533 HOURS
```

   Then the following is printed:

```
ENTER DATE THROUGH DATA SWITCHES
```

7. Enter the date in month, day and year (MDDY) in the data switches as follows:

   Month is a hexadecimal value 1-C entered in data switches 0-3
   Day is a hexadecimal value 1-1F entered in data switches 7-11
   Year is a hexadecimal value 0-E entered in data switches 12-15
   which represents 1970-1984

When the date has been entered, press START and the following is printed:

```
DATE ENTERED WAS  FEB   04  1970
```

8. Cold Start then prints the following message:

```
SET VALUE IN DATA SWITCHES IF REQUIRED BY COLD START CORE LOAD
```

   If the cold start coreload requires that the data switches be set to a certain configuration, set the switches to that value.

9. Press START

   This starts a real-time system as follows:

   a. If this is a cold start to BPMON, background processing begins,

   b. If this is a cold start to a coreload, the coreload is queued for execution.

Cold Start Messages

Each cold start message, whether error, informational, or instructive, has an associated indicator which is loaded into the accumulator if the 1053 specified in the BOM equate SLORG is not ready. The system then comes to a wait. Thus you can interpret certain problems that arise, correct them, and complete the cold start even though the 1053 is inoperative.

To ignore a message and continue, turn ON sense switch 0 and press console START.

To retry the cold start operation, ready the 1053, leave sense switch 0 OFF and press console START. If the reattempt is unsuccessful, the system will again wait with the indicator value in the accumulator.

When the indicator in the accumulator reflects a message requesting that you enter data in the data switches, you must enter that data at that wait; the normal data entry wait is bypassed.

The following are the informational and instructive messages printed by Cold Start. For the error messages see the IBM 1800 Multiprogramming Executive Operating System Error Messages and Recovery Procedures, Order Number GC26-3727. The hexadecimal value beside the cold start message is the indicator that will be placed in the accumulator.

```
9971:  WARNING PUSH START TO CONTINUE
```

This message is simply a warning of a possible error. If you decide this message indicates that you need a new cold start card, then you must cold start again. To bypass the warning, press START.

```
998E:  DATE ENTERED WAS MMM DD 19YY
```

You have entered the date shown.

```
998F:  ENTER DATE THROUGH DATA SWITCHES
```

Enter the date in month, day, and year in the data switches and press START.

```
9991:  NO RELOAD CORELOAD ON NAME CARD.  CALL EXIT EXECUTED AT RELOAD.
```

There is no reload coreload name specified on the name card. The word count and sector address of the reload coreload in the reload information table are set to zero, indicating that a CALL EXIT is to be done during a reload.

9992: ENTER TIME THROUGH DATA SWITCHES

Enter the time in hours and minutes (HH, MM) in the data switches and press START.

9993: TIME ENTERED WAS XX.XXX HOURS

You have entered the time shown.

9994: SET VALUE IN DATA SWITCHES IF REQUIRED BY COLD START CORELOAD

Set up the data switches as required by the cold start coreload.

BOM/Executive Reload

The reloading of a batch-processing system or a real-time system is essentially the restarting of that system after a severe error occurs. A reload differs from a cold start or a loading of BOM from cards in that BOM or the Executive is already in main storage and many of the system indicators have already been set. The reload process merely reinitializes indicators in the system and starts the system running.

The reload function is automatically performed by both the batch-processing system and the real-time system when a severe error condition is encountered while executing. An example of such an error condition is a storage protect violation. See IBM 1800 Multiprogramming Executive Operating System: Error Messages and Recovery Procedures, Order Number GC26-3727, for additional errors that would cause an automatic reload.

If you wish to stop the current execution of your system and restart it, you can do so without cold starting or loading BOM by performing a manual reload. To perform a manual reload, turn off sense switches 0 and 1, and press console STOP, RESET and START.

BOM Reload

When an automatic or manual reload of a batch-processing system is performed, the following BOM initialization messages are printed:

```
IBM 1800 DACS  MPX/BOM   XX.XXX   DD   MMM   YY
SEN SW 0 ON ABSOLUTE LOADER
SEN SW 1 ON LOAD BP MONITOR
SEN SW 2 ON SET CLOCK VIA DATA SWS
SEN SW 3 ON SET DATE VIA DATA SWS
```

where XX.XXX is the time in hours and thousandths of an hour and DD MMM YY is the current date (day, month and year, respectively).

By setting the appropriate sense switch and pressing console START, you can select one of the following batch-processing options:

ABSOLUTE LOADER

Setting sense switch 0 ON and pressing START causes absolute programs in card form to be loaded by the BOM Absolute Loader from the 1442 card read punch into main storage. BOM card utilities and the System Loader are loaded in this way. BOM loads the absolute program and then branches to the entry point of the absolute program to begin its execution.

BATCH-PROCESSING MONITOR

Setting sense switch 1 ON and pressing START causes BOM to read the batch-processing Monitor Supervisor from disk into main storage. The Supervisor then begins searching the input stream for a // JOB statement.

SET CLOCK

To set the clock, set sense switch 2 ON and set data switches 0 through 15 to the time, in hours and minutes, as follows:

Hours:	First digit:	Switches 0-3
	Second digit:	Switches 4-7
Minutes:	First digit:	Switches 8-11
	Second digit:	Switches 12-15

For example, to set the time to 6:17 pm, enter 1817 in the data switches as follows:

```
0001       1000       0001       0111

 1          8          1          7
```

Press START. This causes BOM to reset the clock to the value in the data switches and to print the BOM initialization messages again with the new time:

```
IBM 1800 DACS  MPX/BOM   18.283  00 JAN 00
SEN SW 0 ON ABSOLUTE LOADER
SEN SW 1 ON LOAD BP MONITOR
SEN SW 2 ON SET CLOCK VIA DATA SWS
SEN SW 3 ON SET DATE VIA DATA SWS
```

SET DATE

To set the date, set sense switch 3 ON and set the data switches as follows:

Month	(Hexadecimal 1-C):	Switches 0-3
Day	(Hexadecimal 1-1F):	Switches 7-11
Year	(Hexadecimal 0-E):	Switches 12-15

Note that year values of 0-E correspond to 1970 - 1984, respectively.

For example, to set a date of December 22, 1970, enter C160 in the data switches as follows:

```
1100       0001       0110       0000

 C          1          6          0
```

Press START. This causes BOM to reset the date to the value in the data switches and to print the BOM initialization messages reflecting the new date.

```
IBM 1800 DACS  MPX/BOM   18.392  22 DEC 70
SEN SW 0 ON ABSOLUTE LOADER
SEN SW 1 ON LOAD BP MONITOR
SEN SW 2 ON SET CLOCK VIA DATA SWS
SEN SW 3 ON SET DATE VIA DATA SWS
```

Executive Reload

When an automatic or manual reload of a real-time system is done (Executive reload), the following is performed:

1. Reread from disk the user-included subroutines in the Executive if this option was elected at BOM assembly (BOM equate SYEXR set to 1).

2. Reread from disk the SPAR coreload(s) currently in main storage if this option was elected at BOM assembly (BOM equate RLSPR set to 1).

3. Unmask the system to the mask condition specified in the cold start name card. If a reload coreload was specified at cold start, load this coreload from disk to VCORE and branch to it (BOM equate NOREL must be set to 1). If no reload was specified, program an interrupt to the level and bit named in the LDLEV and LDBIT equates.

This page intentionally left blank.

Background-Processing Operations

Background-processing operations consist of assemblies, compilations, Disk Management Program functions, Builder functions, and background program executions. They run under the control of the Batch-Processing Monitor Supervisor (SUP). SUP reads and analyzes all Supervisor control statements, such as // JOB, // ASM, and // FOR, initializes the Disk Communications area (DCOM) when a // JOB statement is encountered, loads requested programs, and passes control to them.

For a detailed description of the Batch-Processing Monitor Supervisor, refer to the IBM 1800 Multiprogramming Executive Operating System Programmer's Guide, Order Number GC26-3720.

Loading the Supervisor

The Supervisor can run in a batch-processing system (under BOM) or in a real-time system (under the Executive).

To load and pass control to the Supervisor under a batch-processing system, BOM must first be initialized. BOM can be initialized in any of three ways:

- Loading the BOM object card deck to main storage with the BOM high core loader (see "System Generation").
- Cold Starting to BOM (see "Cold Start").
- Reloading BOM (see "BOM/Executive Reload").

When BOM is initialized, the following messages are printed:

```
IBM 1800 DACS  MPX/BOM    XX.XXX   YY MTH ZZ
SEN SW 0 ON ABSOLUTE LOADER
SEN SW 1 ON LOAD BP MONITOR
SEN SW 2 ON SET CLOCK VIA DATA SWS
SEN SW 3 ON SET DATE VIA DATA SWS
```

Setting sense switch 1 ON and pressing console START causes BOM to load and pass control to the Supervisor.

Note that setting sense switch 7 ON and pressing CONSOLE INTERRUPT at any time during execution of any program will cause that program to be aborted, the Supervisor to be loaded, and control to be passed to the Supervisor.

To load and pass control to the Supervisor under a real-time system, do a cold start to BPMON (see Cold Start). Background processing will then be carried out until a // END statement is encountered. Such a statement signals the Supervisor that there is no more background processing. To start background processing again after a // END statement has been encountered, set sense switch 7 ON and press CONSOLE INTERRUPT. This will cause the Supervisor to be loaded and given control.

Input to the Supervisor

Once SUP has control, it begins searching the input stream for a // JOB statement. A // JOB statement causes SUP to terminate any previous background-processing job and to begin a new one.

Other control statements must immediately follow a // JOB statement in the input stream to direct the Supervisor to load the proper program (Macro Assembler, FORTRAN Compiler, etc.). For a detailed description of the MPX control statements, see IBM 1800 Multiprogramming Executive Operating System Programmer's Guide, Order Number GC26-3720. For a summarized description of the control statements, refer to MPX Control Statements, Order Number GX26-1594.

The input stream to the Supervisor may be from the 1442 card read punch or the 1816 keyboard. The card reader is the assumed input source unless otherwise specified.

To specify the 1816 keyboard as the input device, the // SET control statement may be used (see IBM 1800 Multiprogramming Executive Operating System Programmer's Guide, Order Number GC26-3720, or MPX Control Statements, Order Number GX26-1594). The other way to change the input stream device is to press the 1816 keyboard request key with sense switch 7 ON. This will cause execution of the current background-processing operation to be terminated.

Background processing initiated by keyboard request will be terminated after a time lapse of five minutes if no input from the 1816 occurs. Note that this time lapse has the same effect as a // END statement.

When the keyboard is being used as the input stream device, you can go back to using the 1442 in two ways:

- Press CONSOLE INTERRUPT with sense switch 7 ON. This will cause the current background-processing job to be aborted and background processing to continue through input from the card reader.

- Using the // SET statement to transfer the input stream device assignment without terminating the current job.

KEYBOARD INPUT

Input through the 1816 keyboard is accepted by the Disk Management Program as well as the Batch-Processing Monitor Supervisor. That input can consist of any control statements acceptable to these programs with the exception of the *CCEND statement required to end a store operation. That statement must be entered by the card reader. Non-binary input following *MON and *SRFLE control statements may also be entered through the keyboard.

Once the input stream device has been established to be the 1816 keyboard, the green PROCEED light on that keyboard comes on. The Supervisor is then waiting for your input. Your input control statements should be typed in print locations corresponding to card columns. When you have typed the control statement, press the EOF key to send the statement to the Supervisor.

CORRECTION OF ERRORS

If you make a mistake before pressing the EOF key, you may use the ER CHR key to backspace and re-enter a character. Alternatively, you may press the ER FLD key and re-key the entire statement. When the Supervisor or Disk Management Program encounters an error in the control statement, an error message will be typed on the system printer. Other system messages such as DMP FUNCTION COMPLETE will also be printed on the system printer. Note that the system printer you defined at BOM assembly (by BOM equate cards) is not necessarily the 1816 currently being used as your input device. The // SET statement should be used if you want to alter printer assignments.

CHECKING INPUT STATEMENTS

If you want to check an input statement, type two question marks (??) at the end of that statement. This must be done before the EOF key is pressed. The string just entered is typed in red so you can check it. If the statement is acceptable, press EOF. If the statement is unacceptable, cancel it by pressing the NOT sign (¬) key.

```
// FOR NAME ??          (Typed input)

// FOR NAME             (Typed output in red)
```

REQUESTING STATEMENT FORMATS

If you are unfamiliar with the format of the control statement you wish to use, enter a question mark (?) followed by a blank and the name of the control statement and press the EOF key. This will cause the standard format of the specified control statement to be typed, with question marks replacing asterisks and slashes. You can then type your input directly below the line displayed.

```
? DUMP                                  (Typed input)

?DUMP      Z XXN YYM NAMEP   NSEC       (Typed output in red)

*DUMP        UA  PR  BW15D              (Typed input)

DMP FUNCTION COMPLETED
```

You must know the correct name for the control statement. The message INCORRECT OR INSUFFICIENT INPUT. TRY AGAIN is typed if the control-statement name is not correct.

MACRO ASSEMBLER LISTING

This listing shows typical output from the Macro Assembler. The circled numerals on the listing are keyed to the notes that follow the listing.

```
// JOB   09 JUL 70 10.974 HRS
// ASM M   09 JUL 70 10.975 HRS
*LIST
*XREF
*ONE WORD INTEGERS
*COMMON  REAL,INT,ARRAY(2,2,2),IRAY(5,10)
*COMMON/INSKEL/  AREAD,B,IB
```

```
0020                    33 M  MULTI DC       3          2ND ATTEMPT TO DEFINE MULTI
A05 ERRONEOUS STMNT ON PASS ONE
0000 03 C400FFFE         1          LD    L  REAL
0002 03 C400FFFD         2          LD    L  INT
0004 03 C4000000         3 R        LD    L  REAL+INT
0006 02 C400FFFC         4          LD    L  B
0004                     5     C4   EQU      4
0008 02 C400FFF8         6          LD    L  B-C4
000A 02 C4000000         7 R        LD    L  IB+B+4
000C 02 C400FFFF         8          LD    L  IB+4
FFFE                     9     AA   EQU      REAL
000E 03 C400FFFE        10          LD    L  AA
FFFD                    11     S    SET      INT
0010 03 C400FFFD        12          LD    L  S
0012 0  C000            13 R        LD       S
                        14     *
                        15     *
0013 0  62F0            16          LDX   2  -16
0014 03 CE00FFFC        17     LOOP LDD   L2 ARRAY+2    LOAD EXTENTED PREC WORD
0016 0  7202            18          MDX   2  2
0017 0  70FC            19          MDX      LOOP
                        20     *
                        21     *
                        22     *
0018 0  61CE            23          LDX   1  -50
0019 03 C500FFEC        24     LOP  LD    L1 IRAY+1     LOAD INTEGER ELEMENT
001B 0  7101            25          MDX   1  1
001C 0  70FC            26          MDX      LOP
                        27     *
001D 0  C007            28          LD       =REAL     LOAD ADDR OF COMMON ELEMENT
001E 0  C007            29          LD       =12       CONSTANT OF 12
                        30     *
                        31     *
001F 0  0001            32     MULTI DC       1         FIRST DEFINATION OF MULTI
0020 0  0003            33     MULTI DC       3         2ND ATTEMPT TO DEFINE MULTI
                        34     *
0021 01 C400001F        35          LD    L  MULTI     NOTICE NO M FLAG
                        36     *
0023 00 C4000000        37 U        LD    L  UNDEF     REF TO UNDEFINED LABEL
0025                    38     GO   EQU      *
                        39          LORG               DEFINE LITERAL TABLE
0025 3  FFFE            40     +    DC       REAL
0026 0  000C            41     +    DC       12
0027 01 4C000025        42          BSC   L  GO
002A    0025            43          END      GO

005 ERROR(S) AND 000 WARNING(S) IN ABOVE ASSEMBLY.
```

```
SYMBOL  VALUE REL  DEFN    REFERENCES-
AA       FFFE   C-COMMON     10R
AREAD    FFFE   I-COMMON
ARRAY    FFFA   C-COMMON     17R
B        FFFC   I-COMMON      4R      6R      7R
C4       0004   0       5     6R
GO       0025   1      38     42R     43R
IB       FFFB   I-COMMON      7R      8R
INT      FFFD   C-COMMON      2R      3R     11R
IRAY     FFEB   C-COMMON     24R
LOOP     0014   1      17     19M
LOP      0019   1      24     26M
MULTI MULT DFN/SET   32      35R
REAL     FFFE   C-COMMON      1R      3R      9R     40R
S        FFFD   C-COMMON     12R     13R
UNDEF **UNDEFINED-SYMBOL**   37R
```

Notes:

1. All statements that are found to be in error by the Macro Assembler during pass one are printed. These statements may or may not be flagged during pass two.

2. At the end of pass one, the A05 message is printed on the list and system printers if any errors were detected during that pass.

3. A 3 is printed in column 7 if an instruction references a symbol defined in coreload COMMON area.

4. An R flag is printed in column 24 if an illegal reference is made to a COMMON element.

5. A 2 is printed in column 7 if an instruction references a symbol defined in INSKEL COMMON.

6. The SET statement can be used to assign a COMMON element a new name.

7. A literal can be used to obtain the address of an element defined in a COMMON area.

8. This statement shows normal constant definition using a literal.

9. This statement is erroneous but is flagged only on pass one of the assembly.

10. The symbol MULTI is assigned the value of the first definition (line 32). Any reference to a multiply-defined symbol is not flagged with an M.

11. This statement shows normal reference to an undefined symbol. Also see the cross-reference symbol table.

12. The LORG statement is used to instruct the Macro Assembler to define the literals at this point.

13. This DC is generated to define the address of REAL.

14. This DC is generated to define the constant 12.

15. The number printed in the error message signoff includes errors diagnosed during both pass one and pass two.

16. Columns 18-23 show the line number of each statement that the Macro Assembler processes. Even statements such as EJCT, LIST and HDNG which are not printed are assigned a line number.

17. A + is always printed in this column for a Macro Assembler-generated statement.

18. The cross-reference heading is printed at the top of each page of the XREF table. The column under SYMBOL shows each symbol defined in an assembly; under VALUE is shown the value assigned to the symbol during the assembly; under REL is shown a 0 or 1 to indicate whether the symbol is absolute or relocatable; under DEFN is shown the number of the line where the symbol is defined in the assembly (or see 19 and 20); under REFERENCES- is shown the number of the line where each symbol is referenced. The only references that will not be found in this table are from AIF and AGO statements assembled in one-pass mode. (See also 23.)

19. A symbol defined in the coreload COMMON area is assigned the code C-COMMON in the column under the heading DEFN.

20. A symbol defined in INSKEL COMMON is assigned the code I-COMMON in the column under the heading DEFN.

21. A symbol which is either multiply-defined or defined more than once by the SET statement is assigned the code MULT DFN/SET in the column under the heading VALUE REL.

BKGPR

22. A symbol that is referenced but not defined in an assembly is assigned the code **UNDEFINED-SYMBOL**.

23. With each reference line number printed under the heading column REFERENCES-, one of the three letters B, M, and R is printed to indicate that the referenced instruction is of the type branch, modify, or reference. For example, a BSC is a branch, an MDX or STO is a modify, and an LD or A is a reference. A short MDX reference will show an M even though it is a branch.

DMON-System Maintenance Program

This program does real-time maintenance on the MPX system. It operates under DMP and is called by a *MON statement.

There are instances when IBM requires that you do a DMON operation (for example, to install a PTF, change the Version or Modification number of the system, or load a new version of a system program), and there are instances when you may choose to do one yourself (for example, to patch a particular disk word, delete the Macro Assembler and/or FORTRAN Compiler from disk, or modify a system program).

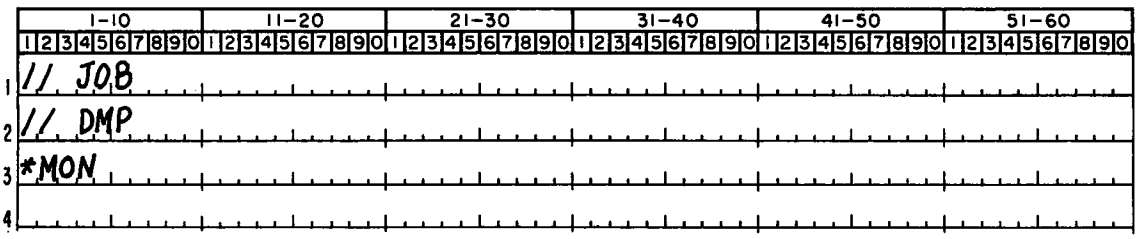
DMON can do the following:

- Patch any word in a disk sector.
- Patch system programs (patching outside current program disk boundaries is permitted).
- Install Program Temporary Fixed (PTFs) to system programs and record them in a log.
- Print the PTF log.
- Update the Version and Modification level of the system.
- Delete the FORTRAN Compiler and/or Macro Assembler from disk and pack the disk.
- Load a new version of a system program already on disk. (New subroutines are loaded by *STOREMD. *MON is used to update the Version/Modification number and the PTF log.)

In summary, DMON permits system updating, keeps a record of the status of the system, prints this status on demand, and prints out a complete picture of all patches installed.

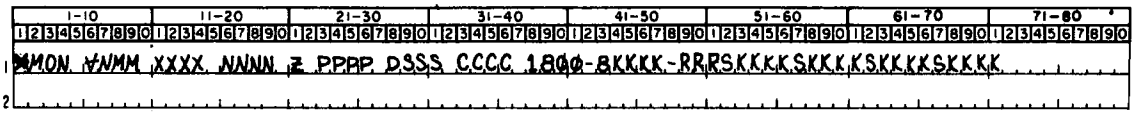
A card deck containing corrections to update the MPX system to its latest Version and Modification level is supplied by IBM. Each time you receive one of these decks you must install the update by running the deck, even if the program affected has been deleted from your MPX system. This is necessary in order to update your system version and modification level.

The calling sequence for DMON is:



The entries in the *MON control statement fields determine the function to be done by DMON. Any number of *MON control statements can be used in a DMON operation.

The following is the general format of the *MON control statement:



where:

VNMM — (columns 6-9) indicates that this PTF is applicable to MPX systems of Version V and Modification levels from MM through MM+N. The numbers must be specified in hexadecimal format. If column 7 contains a U, the system Version and Modification level numbers are updated to the values specified by V and MM.

XXXX — (columns 11-14) is the program name (left-justified) as it appears in LET. (If this field is blank, an absolute patch (see columns 28-31 and 33-36) is being done.)

NNNN — (columns 16-19) specifies the number of *MON statements including this statement, in decimal. This field is used only on the first *MON statement of each DMON function. Leading zeros must be entered.

Z — (column 21) specifies the type of patch records that will follow the *MON statement. It must be either H, which indicates hexadecimal patches, or B, which indicates binary patches.

PPPP — (columns 23-26) specifies the number of patch records following this MON statement, in decimal. Leading zeros must be entered.

DSSS — (columns 28-31) specifies the drive number and sector address, in hexadecimal. This field is used only if the LET entry field (columns 11-14) is blank. The DSSS field is used for patching a relative word or a program on the specified sector.

CCCC — (columns 33-36) specifies the address that would be occupied by the first word of the sector specified in columns 28-31 if it were loaded into main storage. This field is used only if columns 28-31 are used. If this field is blank and columns 28-31 are used, a relative word in the sector is being patched.

The remaining fields of the *MON statement are used to modify, maintain, and print the PTF log.

1800-8 — (columns 38-43) is the PTF sign-on indicator to indicate that a PTF is to be installed. This entry is also used when a printout of the PTF log is desired.

KKKK — (columns 44-47) specifies a number unique to the PTF being installed. If this field is 0000, a PTF is being removed from the PTF log (see columns 52-71).

-RRR — (columns 48-51) is reserved for IBM program support code. The first column in this field must contain a minus sign.

SKKKK

(columns 52-71) these fields are used to provide control information for PTF installation and recording. They serve three purposes:

1. They indicate whether prior PTFs are prerequisite to the PTF currently being installed or removed. A prerequisite PTF is indicated when the S part of a field is blank. The KKKK part is the PTF number, in decimal.

2. They indicate whether prior PTFs are made obsolete by the PTF currently being installed. A PTF made obsolete is indicated when the S part of the field contains a minus sign. The KKKK part contains the PTF number, in decimal.

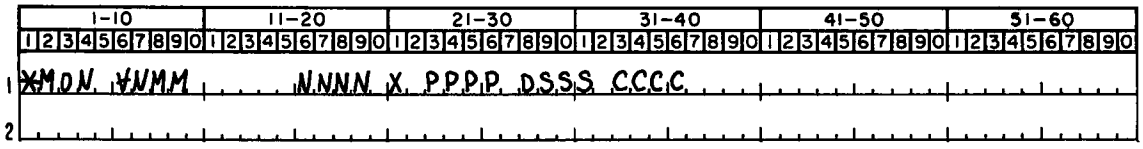
3. If the S part of a field contains a minus sign and columns 44-47 contain zeros, the PTF number specified in the field will be removed from the log.

Any errors in the *MON control statement cause the system to abort the DMON function and return to DMP. However, an error in a patch record terminates the processing of that record only.

Each time a system is regenerated from the PID disks, all patches must be run to bring the system up to operational level. You should keep all program updates and patches in one deck so that they can be loaded by DMON.

Patching Any Word in a Disk Sector

If a patch function is being done, you must use an *MON statement followed by patch records in hexadecimal format or patch cards in binary format. The control statement format is as follows:



Where:

VNMM

(columns 6-9) indicates that this PTF is applicable to MPX systems of Version V and Modification levels from MM through MM+N. The numbers must be specified in hexadecimal.

NNNN

(columns 16-19) indicates the number of *MON statements, including this statement, in decimal. This field is used only on the first *MON statement of each DMON function. Leading zeros must be entered.

X

(column 21) indicates the type of patch cards that follow. This column can contain H for hexadecimal or B for binary.

PPPP

(columns 23-26) indicates the number of patch records following this *MON statement, in decimal. Leading zeros must be entered.

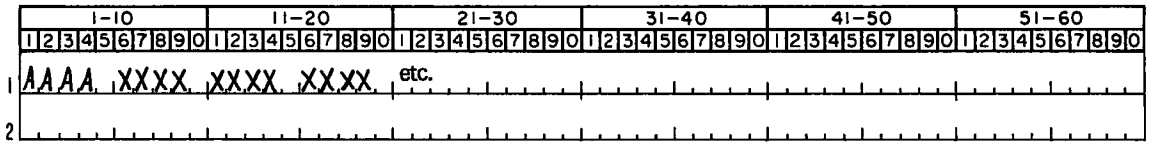
DSSS

(columns 28-31) indicates the drive number and sector address to which the patch will be made.

CCCC — (columns 33-36) indicates the address that would be occupied by the first word of the sector specified in the DSSS field (columns 28-31) if it were loaded into main storage. This field can only be used if columns 28-31 are used. If this field is blank and columns 28-31 are used, a relative word in the sector is patched.

HEXADECIMAL PATCH RECORD FORMAT

The format of a hexadecimal patch record is:



AAAA — specifies, in hexadecimal, the main-storage address of the first word to be patched. If a relative patch is specified (columns 28-31 contain a sector address and columns 33-36 are blank on the *MON control statement), the relative word number is specified. For example, 001F in this field would cause a patch of relative word 31 in the sector specified in columns 28-31 of the *MON control statement.

XXXX — specifies one hexadecimal word of patch data. As many as 15 consecutive words can be specified on one patch record. Each word must be followed by a blank column.

BINARY PATCH CARD FORMAT

Word	Contents
1	Main-storage address of the first word to be patched, or the number of the relative word on the sector.
3	First 8 bits — the type of patch card (1, A, or F). A is the type of a normal patch card. The 1, A, or F appears right-justified in the 8-bit field. Last 8 bits — number of patches on the card.
10-54	Data words 1 through 45.

See Appendix C for a description of card types and an illustration of a binary card.

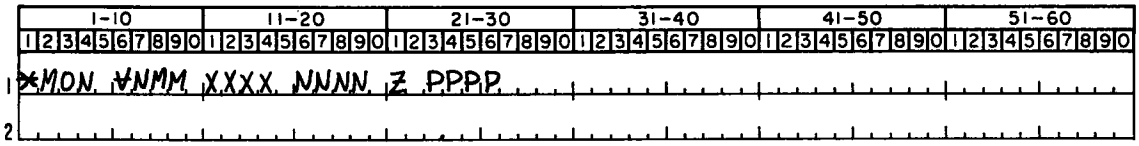
When an absolute patch is made, the patch is printed on the list printer. If data switch 15 is OFF, the system will execute a WAIT after printing each patch. If the patch is correct, push START with data switch 0 OFF. If data switch 15 is ON, the patch will be made and the system will not execute a WAIT.

Patching System Programs

DMON may be used to patch any system program or program phase. It is used to install Program Temporary Fixes (PTFs); a PTF is in the form of a patch or a new program.

A new *MON statement is required for each program or phase patched. The program sets no limits to the number of patch data records that can be used following the *MON statement.

The control statement format is:



VNMM — (columns 6-9) indicates that this PTF is applicable to MPX systems of Version V and Modification levels from MM through MM+N. The numbers must be specified in hexadecimal.

XXXX — (columns 11-14) specifies the program name (left-justified) as it appears in LET.

NNNN — (columns 16-19) indicates the number of *MON statements, including this statement, in decimal. This field is used only on the first *MON statement of each DMON operation. Leading zeros must be entered.

Z — (column 21) indicates the type of patch cards that follow. This column can contain H for hexadecimal or B for binary.

PPPP — (columns 23-26) indicates the number of patch records following this *MON statement, in decimal. Leading zeros must be entered.

Figure 10 illustrates a typical system program patch operation. Sector patches and PTFs are installed in the same manner.

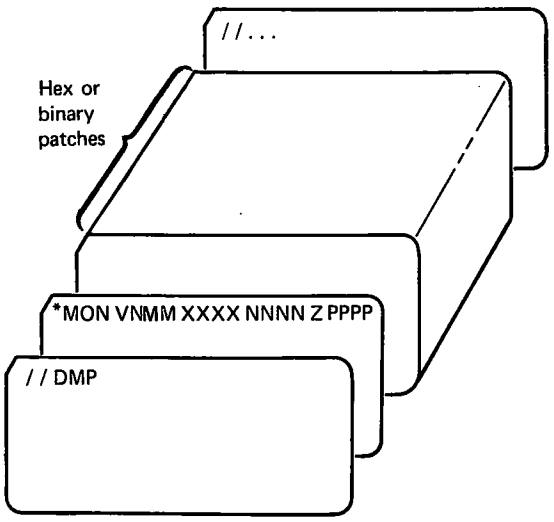
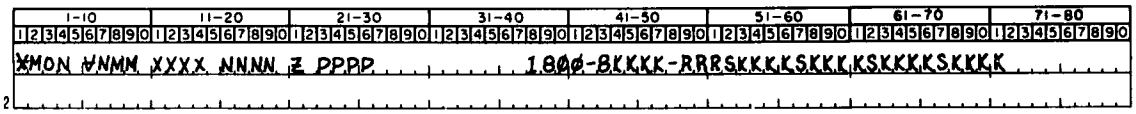


Figure 10. Sequence of Input Statements for a System Program Patch

Installing Program Temporary Fixes (PTFs) to System Programs and Recording Them in a Log

Installing PTFs to system programs is handled in the same manner as described under "Patching System Programs." However, in order to record the installation of a PTF in the PTF log the following format is required for the *MON control statement:



The VNMM, XXXX, NNNN, Z, and PPPP fields are the same as described under "Patching System Programs."

1800-8 (columns 38-43) is the PTF sign-on indicator, used to indicate that a PTF is being installed.

KKKK (columns 44-47) is the number unique to the PTF being installed. If this field is 0000, it indicates that a PTF is being removed from the PTF log (see SKKKK field).

-RRR (columns 48-51) is reserved for IBM program support code. The first column in this field must contain a minus sign.

SKKKK (columns 52-71) is a sequence of fields used to provide control information for PTF installation and recording. They serve three purposes:

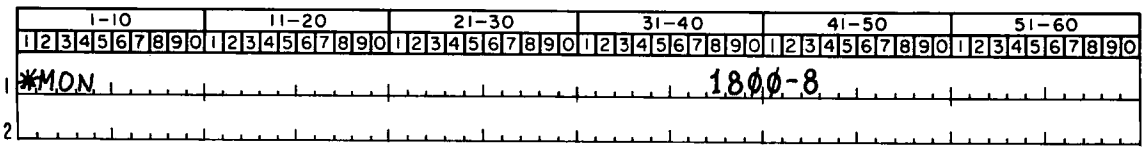
1. They indicate whether prior PTFs are prerequisite to the PTF currently being installed or removed. A prerequisite PTF is indicated when the S part of the field is blank. The KKKK part contains the PTF number in decimal.

2. They indicate whether prior PTFs are made obsolete by the PTF currently being installed. An obsolete PTF is indicated when the S part of the field contains a minus sign. The KKKK part contains the PTF number, in decimal.

3. If the S part of a field contains a minus sign and columns 44-47 contain zeros, the PTF specified in the field will be removed from the log.

Printing the PTF Log

You can obtain a copy of the current PTF log by using an *MON statement containing the following entries:

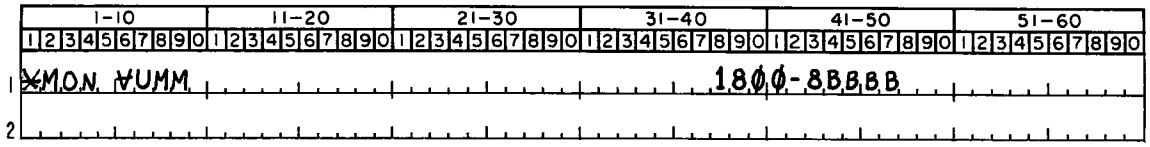


This is an example of the PTF log:

```
// JOB   02 FEB 70 10.316 HRS
// DMP   02 FEB 70 10.318 HRS
*MON                                  1800-8
VER/MOD.  2000.
PTF NO. 0001.
PTF NO. 0002.
PTF NO. 0003.    obsolete
PTF NO. 0004.
PTF NO. 0005.
PTF NO. 0006.
PTF NO. 0007.
PTF NO. 0008.    obsolete
PTF NO. 0009.
PTF NO. 0010.
PTF NO. 0011.
PTF NO. 0012.
DMP FUNCTION COMPLETED
```

Updating the Version and Modication Level of the System

DMON can be used to update the system Version number and modification level number in the Disk Communications Area (DCOM). DMON updates the PTF log by recording the number associated with the PTF in the *MON control statement. The *MON control statement for this operation is as follows:



where:

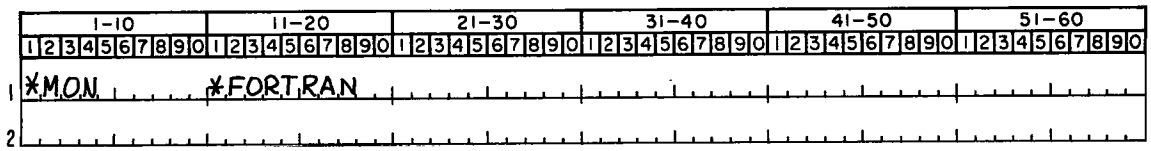
V	(column 6) is the Version number. It can be only one greater than the current Version number. If the Version number is updated, the Modification level must be zero.
U	(column 7) is a Version/Modification level update indicator.
MM	(columns 8-9) is the Modification level number. It can be only one greater than the current Modification level number.
BBBB	(columns 44-47) is one less than the first PTF number assigned to the new Modification level.

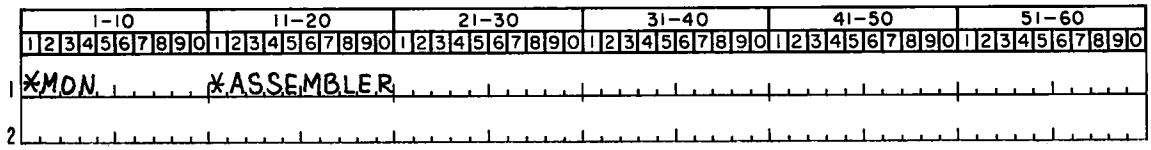
Note: A PTF number is ignored if it is specified as a PTF to be deleted, obsoleted, or specified as a prerequisite and its value is less than the BBBB field described above.

Deleting the FORTRAN Compiler and/or Macro Assembler from Disk and Packing the Disk

DMON may be used to delete the Macro Assembler and/or the FORTRAN Compiler from the system. Disk areas are repacked following the deletion.

The *MON statements for deleting the FORTRAN Compiler and the Macro Assembler, respectively, are:





When such a statement is processed, the following messages will be printed:

```
YOU ARE DELETING FORTRAN OR THE
ASSEMBLER
IF OK, PUSH START WITH DATA SWITCH 0
OFF
```

Deletion of the Macro Assembler or FORTRAN Compiler is not recommended for a short-term gain of disk room. These programs may be reloaded from cards using *MON (if they are available in card form), but it will take from one to three hours to reload either of them. If the programs are not available in card form, a new system generation must be performed to restore these programs to the system.

Loading New Versions of System Programs Already on Disk

A single *MON control statement can be used to load more than one new object program deck, so long as the programs being replaced are system programs and the last deck is followed by an *CCEND card. If the new version of a program requires more disk sectors than its previous version, DMON will enlarge the program area to make an additional sector available. This increase is limited to one sector. Similarly, if a new version of a program requires fewer disk sectors than its previous version, DMON allows packing of the program area. Packing is also limited to one sector.

Any DMON function that involves an expansion or contraction of one sector of the affected program will cause the following message to be printed:

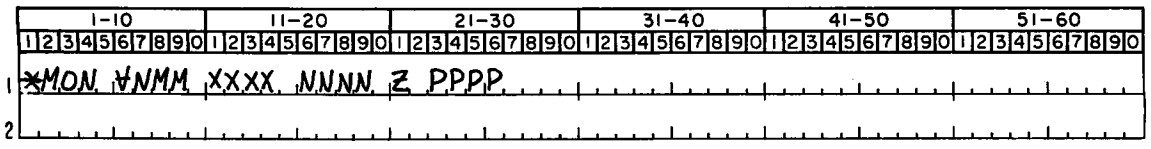
```
UPDATE REQUIRES DISK MOVE
```

If data switch 1 is on, the program performs the move and continues. If data switch 1 is OFF, the following message is printed and the program executes a WAIT:

```
IF OK, PUSH START WITH DATA SWITCH 0
OFF
```

Turn ON data entry switch zero and press console START if you want to abort the current operation.

The control statement format for loading a new version of a system program is:



VNMM — (columns 6-9) indicates that this PTF is applicable to MPX systems of Version V and Modification levels from MM through MM+N. The numbers must be specified in hexadecimal.

XXXX — (columns 11-14) is the name of the program or program phase, left-justified, as it appears in LET. If the *MON statement is being used to replace two or more programs, this is the name of the first one.

NNNN — (columns 16-19) is the number of *MON statements, including the statement, in decimal. This field is used only on the first *MON statement of each DMON function. Leading zeros must be entered.

Z — (column 21) must be B, indicating that binary patch cards follow the *MON statement.

PPPP — (columns 23-26) is a decimal number equal to or greater than the total number of object cards following the *MON control statement.

The load of a new version of a system program is terminated by an *CCEND control card; this card must follow the final (or only) object deck being loaded.

Typical input for a system program update is illustrated in Figure 11.

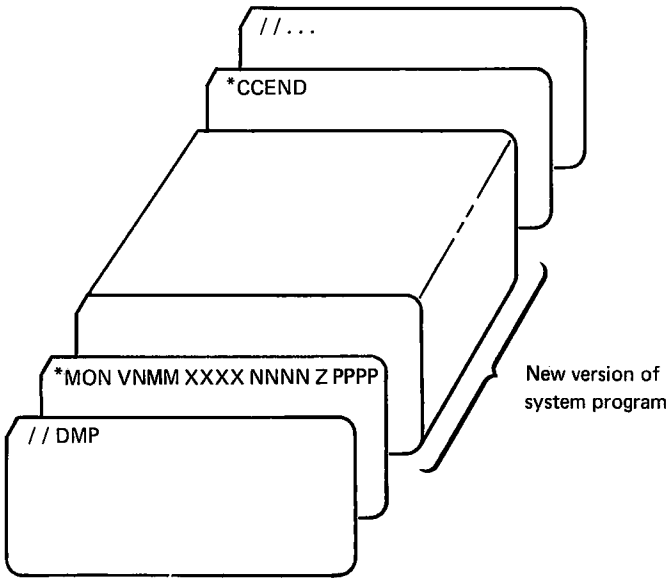
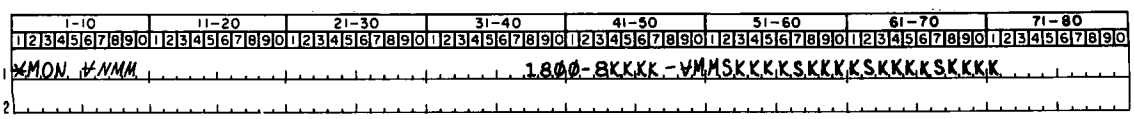


Figure 11. Input for a System Program Update

Replacing Library Subroutines

Changes to the MPX Subroutine Library require the deletion of the old program and the storing of the new one. *MON just updates the Version and Modification level word; the actual loading of the new program is performed by a DMP *STOREMD operation.

The format of the *MON control statement is:



The fields were described in the section "Installing Program Temporary Fixes (PTFs) to System Programs."

Typical input for Subroutine Library maintenance is illustrated in Figure 12.

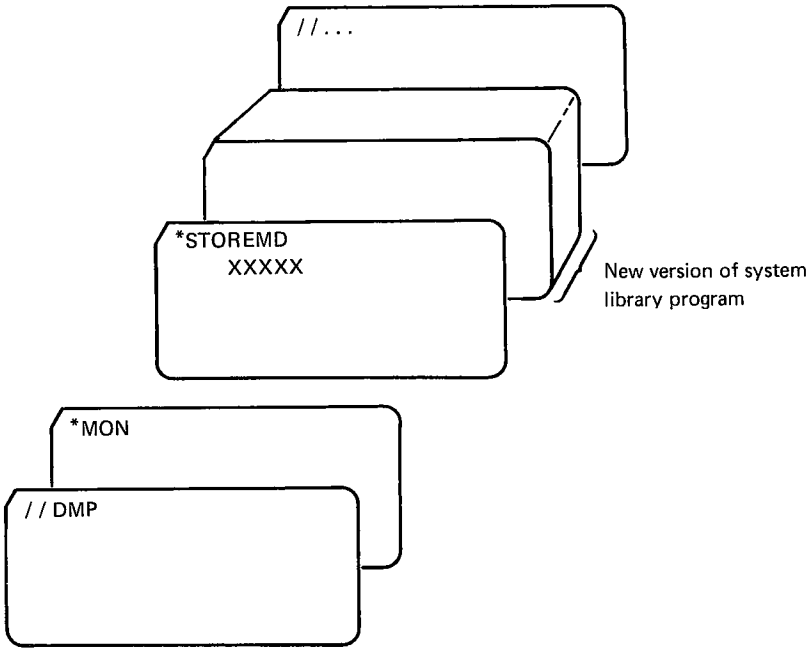
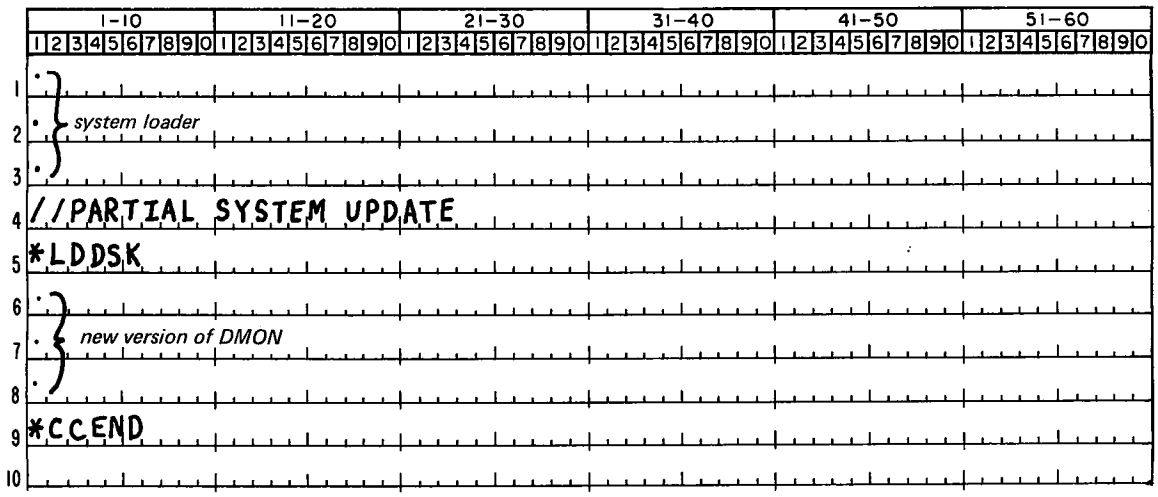


Figure 12. Deck for Updating the Subroutine Library

Loading a New Version of DMON

DMON cannot be used to replace itself. New versions of DMON must be loaded by the System Loader, under control of BOM. The following input is required to load a new version of DMON.



Data switch 14 ON will suppress all printing (except for error messages) during the loading process. When a new version of DMON is loaded, an *MON function should be performed to update the PTF log in DCOM.

Note: The sequence shown for replacing DMON is also valid for replacing any other system program.

BOM Card Utilities

A group of utility programs that can be loaded into main storage by the BOM Absolute Loader are punched during the system-generation process.

These object programs are:

BDCRL	-	BOM 2311 Disk Dump to Cards and Reload Program
BDUMP	-	BOM 1810 Disk Dump to Cards Program
BDPAT	-	BOM Disk Patch Program
BDPIP	-	BOM 2311 Disk Pack Initialization Program
BDUPL	-	BOM Disk Duplication Program
BDWAP	-	BOM Disk Write Addresses Program
BLIST	-	BOM 80-80 List and Sequence Check Program
BRELD	-	BOM 1810 Disk Dump Reload Program

During the operation of each of these programs instructions will be printed on the system printer. You will have the ability to select various options by entering data through the console switches. The functions requested and the data entered are printed during the operation of these programs.

You can return to BOM at any time by pressing console STOP, RESET, and START with all sense switches OFF.

BDCRL - BOM 2311 Disk Dump to Cards and Reload Program

BDCRL can dump all or part of the contents of a 1316 disk pack on drive 0 to cards, and can reload such data from cards to disk. The program can be used to dump contents from one disk pack onto cards and to then reload these contents onto another pack. Contents to be dumped can begin at any cylinder address and end at any other cylinder address.

DUMP OPERATING PROCEDURES

1. Load BOM into main storage.
2. Place BDCRL into the card read punch hopper followed by blank cards.
3. Execute the BOM absolute loader (sense switch 0 ON, data switch 15 ON).
4. The following message is printed when program execution begins:

```
BOM 2311 DISK DUMP/RELOAD PGM
CYLINDER ADDRESS RANGE 0000-00C7
SET DATA SW 0 ON TO DUMP
SET DATA SW 0 OFF TO RELOAD
```

5. Set data switch 0 ON.
6. Press console START.
7. The following message is printed:

```
DATA SW = STARTING CYLINDER ADDRESS
```

8. Set the address of the first cylinder to be dumped in the data switches.
9. Press console START.
10. The following message is printed:

```
DATA SW = LAST CYLINDER ADDRESS
```

11. Set the address of the last cylinder to be dumped in the data switches.
12. Press console START.

The program will now begin the dump of the 1316 disk pack to cards. When the dump is complete, the program returns to step 4. Press console STOP, RESET, and START to end the BDCRL operation.

The volume label record (cylinder 0, track 0, record 3) contains a pointer to the Volume Table of Contents (VTOC). If cylinder 0 is not dumped, the VTOC is assumed to be at the disk address currently indicated by the existing volume label record on the disk to be reloaded. You normally should dump cylinder 0 when dumping the VTOC (any time you dump less than the entire disk).

RELOAD OPERATING PROCEDURES

1. Load BOM into main storage.
2. Place BDCRL into the card read punch hopper followed by the cards to be reloaded (obtained from a previous BDCRL disk dump to cards) and two blank cards.
3. Execute the BOM Absolute Loader (sense switch 0 ON, data switch 15 ON).
4. The following message is printed when program execution begins:

```
BOM 2311 DISK DUMP/RELOAD PGM
CYLINDER ADDRESS RANGE 0000-00C7
SET DATA SW 0 ON TO DUMP
SET DATA SW 0 OFF TO RELOAD
```

5. Set data switch 0 OFF.
6. Press console START.
7. The following message is printed:

```
DATA SW = LAST CYLINDER ADDRESS
```

8. Press console START.

When the reload is complete, the program returns to step 7. Press console STOP, RESET, and START to end the BDCRL operation.

CARD FORMAT

The format of data on cards is Card Data (CDD) Format. Each new record begins on a new card. The first word of each new card contains zero. The formats of the other cards are as follows:

First Cylinder Break Card

Word	Contents
1	/FFFF
2	Checksum
3	Reserved
4	Cylinder Address
5	/FFFF
6-59	Reserved
60	Sequence number

Subsequent Cylinder Break Cards

Word	Contents
1	/FFF
2	Checksum
3	Reserved
4	Cylinder Address
5-59	Reserved
60	Sequence number

Cylinder Data Card

Word	Contents
1	Relative data word address to the beginning of the record
2	Checksum
3	Word count of number of data words
4-59	Data words
60	Sequence number

End-of-All-Cylinder Cards

Word	Contents
1	Reserved
2	Checksum
3	/FFFF
4-59	Reserved
60	Sequence number

BDUMP - BOM 1810 Disk Dump to Cards Program

This program dumps all or part of the contents of 1810 drive 0 onto cards. Drive 0 may be physical or mapped. The file protect status of each sector is included in the dump. The data can be reloaded to a new 1810 drive by using the BOM 1810 Disk Dump Reload Program (BRELD).

OPERATING PROCEDURES

1. Load BOM into main storage.
2. Place BDUMP into the card read punch hopper followed by blank cards.
3. Execute the BOM Absolute Loader (sense switch 0 ON, data switch 15 ON).
4. The following message is printed when program execution begins:

```
BOM DISK DUMP TO CARDS DRIVE 0
CYLINDER ADDRESS RANGE 0000-01FF
DATA SW = FIRST CYLINDER ADDRESS
```

5. Set the address of the first cylinder to be dumped in the data switches.
6. Press console START.
7. The following message is printed:

```
DATA SW = LAST CYLINDER ADDRESS
```

8. Set the address of the last cylinder to be dumped in the data switches (maximum of 00C7 for a physical 1810 drive, maximum of /01FF for a mapped 1810 drive).

9. Press console START.

The program will now begin the dump of the 1810 drive to cards. When the dump is complete, the program returns to step 4. Press console STOP, RESET, and START to end the BDUMP operation.

CARD FORMAT

The output of the BOM Disk Dump to Cards Program is in binary format, 3/4 of a 16-bit binary word to a card column. This format allows sixty data words to be punched in each 80-column card (1-1/3 columns equal one binary word). (Appendix C shows the layout of a binary card.) The disk data output is in the following format:

Cylinder Break Card

Word	Contents
1	/FFFF
2	Checksum
3	Reserved
4	Sector address
5	File protect status (/8XXX=on)
6-59	Reserved
60	Sequence number

Cylinder Data Card

Word	Contents
1	Relative address of data on cylinder
2	Checksum
3	Word count of number of data words
4-59	Data words
60	Sequence number

End-of-All-Cylinder Cards

Word	Contents
1	Reserved
2	Checksum
3	/FFFF
4-59	Reserved
60	Sequence number

BDPAT - BOM Disk Patch Program

This program allows you to alter any data words on any 1810 drive (physical or mapped) in the system using disk address cards and data cards loaded as card sets.

A card set is defined as a disk address card followed by any number of data cards. The data cards cause the specified words to be modified. Any number of card sets may be loaded by the disk patch program.

OPERATING PROCEDURES

1. Load BOM into main storage.

2. Place BDPAT into the card read punch hopper followed by the card sets and two blank cards.

3. Execute the BOM Absolute Loader (sense switch 0 ON, data switch 15 ON).

4. The following messages are printed when program execution begins:

```
BOM DISK PATCH PROGRAM
DISK ADDR ORIGIN IS AAAA
DATA ADDR ORIGIN IS BBBB
ABSA SECA RELA OLDV NEWV
CCCC DDDD EEEE FFFF GGGG
  .    .    .    .    .
  .    .    .    .    .
  .    .    .    .    .
```

Note that the patch has already taken place when these messages are printed.

AAAA is the drive number and sector address of the first sector of the disk area being patched.

BBBB is the main-storage address that would be occupied by the first data word of the area being patched if it were loaded into main storage.

CCCC is the main-storage address of the word being patched.

DDDD is the address of the sector actually being patched.

EEEE is the relative address within the sector of the word being patched.

FFFF is the old value of the patched word.

GGGG is the new value of the patched word.

If an invalid data card or a blank card is read, the program prints the following message and executes a WAIT:

```
DATA CARD ERROR OR
NO MORE DATA CARDS
THEREFORE JOB HALTED
```

If the WAIT is due to a data card error, correct the card and place the entire card set, starting with the disk address card, back into the card read punch hopper. Press reader START and console START. If the WAIT is not due to a data card error, you may place more card sets in the hopper and press console START to continue, or you may turn off all sense switches and press console STOP, RESET, and START to return to BOM.

CARD FORMAT

Disk Address Card

Columns	Value
1	*(An asterisk signifies a disk address card)
2-5	Drive number and first sector address (DSSS) of disk area to be patched (hexadecimal disk addresses are listed in Appendix B)
6	Blank

Columns	Value
7-10	Main-storage address of the first data word of the area being patched
11-80	Not used

Data Card

Columns	Value
1-4	Main-storage address of the word being patched (must be equal to or greater than the entry in columns 7-10 on the disk address card)
5	Blank
6-9	The first data word
10	Blank
11-14	The second data word
.	
.	
.	
75	Blank
76-79	The fifteenth data word

BDPIP - 2311 BOM Disk Pack Initialization Program

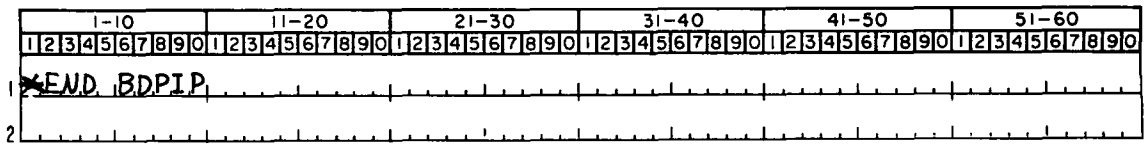
The 2311 BOM Disk Pack Initialization Program in card form (BDPIP) is one of the card utilities punched during system generation.

The 2311 BOM Disk Pack Initialization Program functions according to 2311 initialization control statement specifications. The program has two functions:

- An *INITLZ control statement causes the program to initialize the 1316 disk pack. Initialization includes the following:
  - Optional checking of the existing volume serial number against the one specified in the control statement. The remaining *INITLZ functions are aborted if the volume serial numbers do not match.
  - Checking for defective tracks, with or without an optional surface analysis, with automatic assignment of alternates for those tracks found to be defective.
  - Writing a volume label record on the pack.
  - Constructing and writing the Volume Table of Contents (VTOC).
- An *GETALT control card causes the program to do the following:
  - If a volume serial number is specified, check it against the existing volume serial number and abort remaining *GETALT functions if the numbers are not identical.
  - Assign an alternate track without testing to see if the track is defective.
  - Test a track and, if it is defective, assign an alternate.

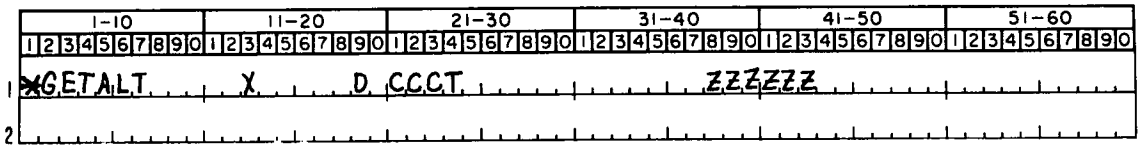
The formats of the control statements follow.

The format of *END BDPIP is:



This control statement indicates the end of a 2311 initialization program job.

The format of *GETALT is:

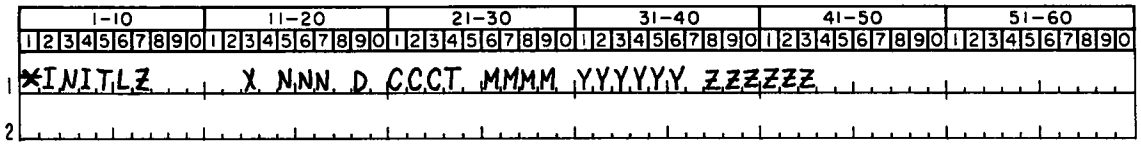


D (Column 19) is the drive number of the 2311. This column must not be blank.

CCCT (Columns 21-24) is the cylinder and track number (both in decimal) of the track for which an alternate is requested. No alternate track can be assigned for track 0, cylinder 0.

ZZZZZZ (Columns 38-43) is the existing volume serial number to be checked. If this field is blank, the job continues without checking. If a volume serial number is specified and does not agree with the volume serial number on the 1316 pack, the job is aborted.

The format of *INITLZ is:



X (Column 13) is blank if the volume is to be initialized with surface analysis; otherwise surface analysis is not performed. This column must be blank when a volume is being initialized for the first time.

NNN (Columns 15-17) is the number (decimal) of times the pattern 55 (hexadecimal) and 00 (hexadecimal) is to be written in every byte of the surface analysis. NNN can range from 01 through 256. If it is blank or 00, it is assumed to be 01.

D (Column 19) is the drive number of the 2311 to be initialized.

CCCT (Columns 21-24) is the cylinder and track number (both in decimal) where the Volume Table of Contents (VTOC) is to begin. The VTOC cannot begin at track 0 of cylinder 0 or at any alternate track.

MMMM (Columns 26-29) is the length of the VTOC in tracks. This field must not be zero. This number can range from 0001 through 1990 (decimal).

YYYYYY (Columns 31-36) is the volume serial number. This field can contain any alphabetic or numeric characters.

ZZZZZZ (Columns 38-43) is the existing volume serial number to be checked. If this field is blank, initialization continues without checking. If a volume serial number is specified and is not equal to the volume serial number on the 1316 pack, the initialization job is aborted.

1. Load BOM into main storage.

2. Place BDPIP into the card read punch hopper followed by 2311 initialization control cards (*END BDPIP must be the last control card) and a blank card.

3. *Execute* the BOM Absolute Loader (sense switch 0 ON, data switch 15 ON).

4. The following message is printed when program execution begins:

```
BOM DISK PACK INITIALIZATION PROGRAM
```

5. Press console START.

6. At this point the first initialization control statement (*INITLZ or *GETALT) is read and the following message is printed:

```
DRIVE SELECTED IS XX, IF CORRECT PUSH
START WITH SENSE SWITCH 0 ON.
```

7. If the drive number is correct, press console START with sense switch 0 ON. If the drive is incorrect:

    1. Press START with sense switch 0 OFF; the program will return to its beginning (step 4).

    2. Then correct the drive number in the first initialization control statement and push console START. The program will return to step 6 and print the message again.

8. Press console STOP, RESET, and START to end the BDPIP operation.

When the 2311 initialization job has been completed, one of the following messages is printed:

1. If no defective tracks were found:

```
THERE WERE NO DEFECTIVE TRACKS FOUND ON THIS PACK
```

2. If defective tracks were found:

```
THE FOLLOWING TRACKS WERE FOUND DEFECTIVE ON THIS PACK:
DEF TRACK ADDR                    ALT TRACK ADDR
      CCCT                              CCCT
```

    where the CCCT fields are the cylinder and track addresses of the defective and alternate tracks.

If program detects a control statement error, it will print an error message and execute a WAIT. When this happens, place the corrected control cards into the hopper, ready the read punch and press console START.

Note: To make your 1316 pack DOS-compatible, it must be initialized under DOS. BDPIP will initialize your 1316 to be OS-compatible.

BDUPL - BOM Disk Duplication Program

The BOM Disk Duplication Program copies the contents of an 1810 or 2311 drive onto another 1810 or 2311 drive. An 1810 mapped or physical drive can be copied onto another 1810 mapped or physical drive, while a 2311 drive can only be copied onto another 2311 drive. The drive being copied is called the source drive and the drive being copied onto is called the object drive.

OPERATING PROCEDURES

1. Load BOM into main storage.

2. Place BDUPL into the card read punch hopper followed by one blank card.

3. Execute the BOM Absolute Loader (sense switch 0 ON, data switch 15 ON).

4. The following message is printed when program execution begins.

```
BOM DISK COPY PROGRAM

SET DATA SWS TO SOURCE AND OBJECT
  DRIVE CODES--XSXO
SET DATA SW 1 ON IF 2311 COPY
```

5. Set the source drive number in data switches 4-7 and set the object drive number in data switches 12-15. Set data entry switch 1 ON if you are copying a 2311 drive.

6. Press console START.

7. The following message is printed if you are copying an 1810 drive:

```
SOURCE DRIVE IS 000M
OBJECT DRIVE IS 000N
IF CORRECT PUSH START WITH DATA SW 0 OFF
```

   where M and N are the disk drive numbers (0-7).

   The following message is printed if you are copying a 2311 drive:

```
2311 DISK COPY
SOURCE DRIVE IS 000M
OBJECT DRIVE IS 000N
IF CORRECT PUSH START WITH DATA SW 0 OFF
```

   where M and N are the disk drive numbers (0-7).

8. M is the source drive number and N is the object drive number. If either of the drive numbers is in error, set data switch 0 ON and press console START to return to step 4. If both numbers are correct, press console START with data entry switch 0 OFF.

When the duplication is finished, the program returns to step 4. Press console STOP, RESET, and START to end the BDUPL operation.

Note that mapped 1810 drives can vary in size. If either the source or object drive, or both of them, is a mapped 1810 drive and they differ in sector size, the following message will be printed as you begin to execute the program:

```
SOURCE SECTORS XXXX
OBJECT SECTORS YYYY
PUSH START WITH SEN SW 1 ON TO EXECUTE
```

where:

   XXXX is the number of sectors on the source drive.
   YYYY is the number of sectors on the object drive.

If you decide to continue with the copy as specified, press console START with sense switch 1 ON. The number of sectors copied will be the smaller of XXXX and YYYY. If you wish to terminate the job, press console START with sense switch 1 OFF and the program will return to step 4.

The main purpose of copying drives of unequal size is to temporarily store the contents of a smaller drive onto a larger one. If you decide to continue copying, you will be unable to use the object drive as a system drive.

BDWAP - BOM Disk Write Addresses Program

The BOM Disk Write Addresses Program checks the disk surface of a 2315 cartridge and writes logical sector addresses in the sectors of those cylinders that are not defective. If a cylinder on the 2315 disk cartridge is damaged so that it is impossible to write on it, BDWAP will ignore that cylinder.

BDWAP must be used to initialize all new 2315 cartridges that are to be used with MPX. It may also be used at any other time that you wish to zero your 2315 cartridge and write addresses on it.

OPERATING PROCEDURES

1. Load BOM into main storage.
2. Place BDWAP into the card read punch hopper followed by a blank card.
3. Execute the BOM Absolute Loader (sense switch 0 ON, data switch 15 ON).
4. The following message is printed when program execution begins:

```
BOM DISK WRITE ADDRESSES PROGRAM
ENTER NO. TRIES ON DATA SW MAX 001F
```

5. Set the number of tries in the data entry switches, right-justified. The entry must be a hexadecimal number in the range /0001 through /001F.

   The number of tries designates the number of time BDWAP will try to write the hexadecimal pattern /E5E5/1313 in every word of the 1810 drive.

6. Press console START.
7. If the number of tries entered is not acceptable, the following message is printed:

```
ENTER NO. TRIES ON DATA SW MAX 001F
```

8. Correct the entry and press console START. If the number of tries entered is acceptable, the following message is printed:

```
 DATA SWITCHES EQUAL LOGICAL DRIVE
DRIVE CODES - HEX 0000 0001 0002
```

9. Set the drive number in the data switches as follows:

   All off - drive 0
   Switch 15 ON - drive 1
   Switch 14 ON - drive 2

10. Press console START.

    The drive number is checked for legality. If the drive number selected is not a legal drive number, the program returns to step 8. If the drive number is a legal drive number, the following message is printed:

```
DRIVE SELECTED IS X, IF CORRECT PUSH
START WITH SENSE SWITCH 0 ON.
```

11. If the drive selection is correct, turn sense switch 0 ON and press console START. If the drive selection is incorrect, turn sense switch 0 OFF and press console START to return to step 8.

Addresses are now written on the 1810 drive and one of the following messages is printed:

a. THERE ARE NO DEFECTIVE CYLINDERS

Upon completion of this printout the program returns to step 4 and is ready to address another 2315 cartridge. To end the BDWAP operation, press console STOP, RESET, and START with all sense switches OFF.

b. If there are one, two, or three defective cylinders, the following message is printed:

```
CYLINDERS 00XX 00XX 00XX ARE DEFECTIVE
```

where the 00XX fields are the physical cylinder numbers (in hexadecimal) of the defective cylinders. The 2315 cartridge can be used. The program returns to step 4 and is ready to address another 2315 cartridge. To end the BDWAP operation, press console STOP, RESET, and START with all sense switches OFF.

c. If cylinder 0 is defective or if there are more than three defective cylinders, the following messages are printed:

```
THIS DISK CARTRIDGE IS NOT ACCEPTABLE TO MPX
TOO MANY CYLINDERS ARE BAD OR CYLINDER ZERO IS BAD
CYLINDERS (0000 OR 00XX ... 00XX) ARE DEFECTIVE
```

where the 00XX fields are the cylinder numbers (in hexadecimal) of the defective cylinders. The 2315 disk cartridge cannot be used with MPX. The program returns to step 4 to allow you to load a new 2315 disk cartridge.

To end the BDWAP operation, press console STOP, RESET, and START with all sense switches OFF.

d. If repeated seek failures occur, the following message is printed:

```
CANNOT COMPLETE SEEK-ABORT JOB
```

The program returns to step 4 to allow you to retry. To end the BDWAP operation press console STOP, RESET, and START with all sense switches OFF.

The physical cylinder numbers (P_C) of the defective cylinders are printed in hexadecimal. For each defective cylinder, the number stored in the Defective Cylinder Table on the disk is the logical sector number that would have been assigned to the first sector in the cylinder, had the cylinder not been defective. To compute the logical sector number (L_S) of the first sector of a defective cylinder, use this formula:

$$L_S = (P_C * 8) - (N * 8)$$

P_C was described in the preceding paragraph, and N is the number of defective cylinders preceding the cylinder for which L_S is to be computed.

The cylinder numbers range from 0 through 202 in decimal, or 0 through CA in hexadecimal.

If, for example, you want to find the logical sector number of the first sector of cylinder 10, and cylinder 10 is the third defective cylinder on the cartridge, you can use the following calculations:

$$\begin{aligned} L_S &= (P_C * 8) - (N * 8) \\ &= 10 * 8 - 2 * 8 \\ &= 80 - 16 \\ &= 64 \end{aligned}$$

BLIST – BOM 80-80 List and Sequence Check Program

The 80-80 List and Sequence Check Program in card form (BLIST), is one of the card utilities punched during system generation. The 80-80 List and Sequence Check Program lists card statements on the list printer and/or checks the progression of sequence numbers in card columns 73-80. If, during a sequence check, a card containing a sequence number lower than the preceding one, or containing blanks, is found, an error message is printed and the system executes a WAIT.

OPERATING PROCEDURES

1. Load BOM into main storage.
2. Place BLIST into the card read punch hopper followed by the cards to be listed and/or sequence checked and a blank card.
3. Execute the BOM Absolute Loader (sense switch 0 ON, data switch 15 ON).
4. The following message is printed when program execution begins:

```
BOM 80/80 LIST--SEQUENCE CHECK
SEN SW 0 ON LIST CARDS
SEN SW 1 ON SEQUENCE CHECK
```

5. Set sense switch 0 ON if you wish to list cards and set sense switch 1 ON if you wish to perform a sequence check. Both switches may be ON. The switch settings may be changed while the program is in execution.
6. Press console START.

To terminate the program and return to BOM, turn OFF all sense switches and press console STOP, RESET, and START, in that order.

BRELD – BOM 1810 Disk Dump Reload Program

This program reloads the data punched into cards by the BOM Disk Dump to Cards Program (BDUMP) onto 1810 drive 0 (physical or mapped). The data is loaded in the same relative sectors it occupied on the original 1810 drive 0. If 1810 drive 0 is on a 2315 disk cartridge, the cartridge must have been previously initialized by the BOM Disk Write Addresses Program (BDWAP).

OPERATING PROCEDURES

1. Load BOM into main storage.
2. Place BRELD into the card read punch hopper followed by the output from the BOM Disk Dump to Cards Program (BDUMP) and two blank cards.
3. Execute the BOM Absolute Loader (sense switch 0 ON, sense switch 15 ON).
4. The following message is printed when program execution begins:

```
BOM RELOAD OF DISK TO CARD DUMP
```

5. Press console START.

The program proceeds to load the data from cards to disk. When the End-of-All-Cylinders card is read, the program returns to step 4. To terminate the BRELD operation, press console STOP, RESET, and START.

A disk error during the reload process causes the job to be aborted. The program returns to step 4.

If the source or object drive is mapped on a 1316 disk pack and the drive you are loading is smaller than the drive that was dumped, all cards are loaded up to the addressed cylinder break card and the following message is printed:

```
BOM RELOAD OF DISK FROM CARDS
SRC SAD 640 EXCEEDS OBJ SAD JOB ABORTED
BOM RELOAD OF DISK FROM CARDS
```

This page intentionally left blank.

MPX Debugging Aids

MPX includes programs to aid you in real-time and batch-processing system debugging. These comprise the following:

- BOM Utility Package
- Disk Pack Initialization Program
- Online Dump Program
- 80-80 List and Sequence Check Program
- 1442 Relocatable Card Dump Program
- Card Dump Analysis Program
- TRAP Subroutines
- CE Coreload Programs
- 1442 Card Reader Diagnostic

BOM Utility Package

This package contains a utility monitor (comprising four basic functions) and three trace subroutines (Full Trace, Check/Stop Trace, and Monitor Trace) which may be selectively included in the system at BOM assembly time via BOM equate cards.

UTILITY MONITOR

The Utility Monitor provides four basic functions: BOM or Executive Reload, Disk Dump, Dump, Main-Storage Dump, and Table Dump. Any of these functions can be executed in the following manner:

1. Press STOP and RESET on the 1800 console.

2. Set sense switches 0 and 1 for the function required according to the following table:

Sense Switch 0	Sense Switch 1	Function Called
OFF	OFF	BOM or Executive Reload
OFF	ON	Disk Dump
ON	OFF	Main-Storage Dump
ON	ON	Table Dump for Check/Stop Trace

3. Press console START.

4. Follow the individual directions for the function called.

BOM OR EXECUTIVE RELOAD

You can force BOM or the Executive to perform a reload by pressing console STOP, RESET, and START with sense switches 0 and 1 OFF.

BOM RELOAD

When this function is called, the following standard BOM sense-switch-option messages are printed; BOM then proceeds to execute the reload function for a batch-processing system.

```
IBM 1800 MPX/BOM XX.XXX   YY MTH ZZ
SEN SW 0 ON ABSOLUTE LOADER
SEN SW 1 ON BP MONITOR
SEN SW 2 ON SET CLOCK VIA DATA SWS
SEN SW 3 ON SET DATE VIA DATA SWS
```

Calling the BOM reload function allows you to return to the initial starting point of BOM without the necessity of reloading the BOM deck, or performing a cold start operation. You should also note that whenever batch processing is in progress under BOM or background processing, the job in progress can be aborted by turning program switch 7 ON and pressing CONSOLE INTERRUPT. This action causes the Batch-Processing Monitor Supervisor to be read into main storage. The Supervisor then starts searching the job stream for a // JOB statement.

EXECUTIVE RELOAD

With the Executive in direct control of a real-time system, you have the capability of initiating a system reload from the operator's console as indicated above. The action taken in a system reload is as follows:

1. Reread from disk the user-included subroutine area if this option was elected at BOM assembly.

2. Reread from disk the SPAR coreloads currently in main storage (as indicated by its word count and sector address in words 1 and 2 of the SPAR areas) if this option was elected at BOM assembly.

3. The system is now unmasked to the user-specified mask condition. If a reload coreload was specified at cold start, the system will reload this coreload and branch to it. If no reload coreload was specified, the system will perform a CALL LEVEL (to the level and bit specified on the BOM equates LDLEV and LDBIT) and a CALL EXIT on the basic level.

DISK DUMP

Calling the disk dump function allows you to dump specified portions of a disk on the list printer. The following printout appears when this function is called:

```
SEN SW 1 OFF FOR 2311 DISK DUMP
```

1810 DUMP

After the above printout, the following procedure is required:

1. Push console START with sense switch 1 ON.

2. This message is printed:

```
DATA SW = DRV CODE AND SEC ADD DAAA
```

3. Set sense switch 0 ON to dump two or more sectors. Leave sense switch 0 OFF to dump one sector.

4. For disk dumps, set sense switches 2 and 3 to indicate the character code required for disk analysis, as follows:

Sense Switch 2	Sense Switch 3	Character Code
OFF	OFF	1053 Printer Code
OFF	ON	1443 Printer Code
ON	OFF	EBCDIC Code
ON	ON	Suppress character output

5. Enter the logical disk drive number (right-justified) in hexadecimal in data switches 0 through 3, and enter the hexadecimal starting sector address in data switches 4 through 15.

As long as sense switch 0 remains ON, sequential sectors starting at the starting sector address are dumped. When sense switch 0 is turned OFF, the sector being dumped is completed and the subroutine returns to step 2.

The format of the disk dump is as follows:

```
SECTOR XXXX FILE PROTECT ON/OFF
RELATIVE DATA ADD YYYY TO SECTOR ZZZZ
DISK READ ERROR PARAMETER 000E

0AAA DDDD DDDD DDDD ---- DDDD *ABCDE----*
```

where:

XXXX	is the sector address.
ON/OFF	either ON or OFF is printed depending up the file protect status of the sector.
YYYY	is the address of the first data word of the sector (relative to the first data word of the first sector dumped).
ZZZZ	is the address of the first sector dumped.
000E	is the completion code returned by BULKN following the read of the sector to be dumped. A correct read is indicated by a completion code of 1. If a completion code other than 1 is printed, check the completion code meaning to determine the validity of the data. (The contents of the read buffer are dumped independently of the completion code returned by BULKN.)

The BULKN completion codes are as follows:

Value	Meaning
1	Successful completion of call
2	Device logically offline
3	Device not ready*
4	Parity error
5	Write select*
6	Data error
7	Data overrun
8	Seek error**
9	File protect error
A	Bad sector address on disk*
B	Address modification

*When one of these errors occur, the drive is automatically taken off line

**If the disk is a fast-access device (1810 Model B), it is taken off line when this error occurs

0AAA — is the relative address of the next data word within the sector.

DDDD — are the data words in the sector. Sixteen data words are printed per line on the 1443 printer; 8 data words per line on the 1053 printer.

ABCDE--- — is an analysis of the data words based on a specified character code selected under step **4**. Periods are printed for both periods and unrecognizable codes. Two characters per word are printed.

If the contents of two or more succeeding lines of the dump are identical (for example, all zeros), only the first is printed. Printing is then suppressed until new data is found. The printer then takes an additional line space and prints the new line. The output is in hexadecimal.

Note: A disk dump uses a large part of VCORE as a buffer. The buffer begins at the higher-numbered addresses of VCORE; therefore, if you want a main-storage dump of this part of VCORE, you should get it prior to doing a disk dump operation which will destroy the contents of the area.

The system prints out the following message:

```
SEN SW 1 OFF FOR 2311 DISK DUMP
```

Then carry out the following procedures:

1. Set sense switch 1 OFF.
2. Press console START.
3. The following message is printed:

```
2311 DATA SW=DRV-CYL-TRACK ADD DCCT
```

4. Set sense switch 0 ON to dump two or more records. Leave sense switch 0 OFF to dump one record.
5. For disk dumps, set sense switches 2 and 3 to indicate the character code required for disk analysis, as follows:

Sense Switch 2	Sense Switch 3	Character Code
OFF	OFF	1053 Printer Code
OFF	ON	1443 Printer Code
ON	OFF	EBCDIC Code
ON	ON	Suppress character output

Note that the character code, not the output device, is selected at this time.

6. Enter the disk drive number (0, 1, 2, 3, 4, 5, 6 or 7) (right-justified) in hexadecimal in data switches 0 through 3 and enter the hexadecimal starting cylinder and track addresses in data switches 4 through 15.
7. Press console START.

As long as sense switch 0 remains ON, sequential records, starting at the starting disk address, are dumped. When sense switch 0 is turned OFF, one more record is dumped, and the subroutine returns to step 3.

The format of the disk dump is as follows:

```
DRIVE X  CYLINDER YY   TRACK ZZ
ERROR PARAMETER 000E
RECORD WW
COUNT
AAAA CCCC CCCC CCCC ....
KEY
BBBB KKKK KKKK KKKK ....
DATA
EEEE DDDD DDDD DDDD ....
```

where

X	is the 2311 drive number.
YY	is the cylinder being dumped.
ZZ	is the track being dumped.
000E	is the FILEN error parameter. The record is printed only for completion codes of 1 (successful I/O operation) and 6 (data check).

WW	is the relative record on track ZZ.
AAAA	is the relative count address.
CCCC	is the count field.
BBBB	is the relative key address.
KKKK	is the key field.
EEEE	is the relative data word address.
DDDD	is the data field.

Note: If an attempt is made to dump a 2311 record which is larger than VCORE, the error message RECORD EXCEEDS VCORE will be printed. Recovery is automatic. When subsequent records that fit into VCORE are read, they will be dumped. 2311 Disk Dump uses a large part of VCORE as a buffer. The buffer begins at the start of VCORE. Therefore, if you want a dump of VCORE, you should get it prior to doing a disk dump operation which will destroy the contents of the area.

MAIN-STORAGE DUMP

Calling the main-storage dump function allows you to dump selected portions of main storage on the list printer. The function can be called directly from your program or via the Utility Monitor sense switch options. The contents of the A-register and the Q-register are destroyed during a dump. When the dump is called via the Utility Monitor, the following procedure is required for execution:

1. Set sense switches 2 and 3 to indicate the character code required for main-storage interpretation, as follows:

Sense Switch 2	Sense Switch 3	Character Code
OFF	OFF	1053 Printer Code
OFF	ON	1443 Printer Code
ON	OFF	EBCDIC Code
ON	ON	Suppress character output

2. Enter the hexadecimal starting address in the data switches.
3. Press console START.
4. Enter hexadecimal stopping address in the data switches.
5. Press console START.
6. When the main storage dump is completed, the system returns to step 1. A new dump can be initiated if desired. The format of the main-storage dump is as follows:

```
ADDR 0000 1111 2222 3333 .... .... 7777(1053)    *ABCD---*
                                   FFFF(1443)
```

where:

ADDR	is the starting main-storage address of the line of data words printed.
0000	is the contents of the ADDR+0 word of main storage.
1111	is the contents of the ADDR+1 word of main storage.

2222	is the contents of the ADDR+2 word of main storage.
ABCDE---	is the interpretation of the data words on the basis of a specified character code (2 characters/word) selected in step 1 above.

Identical lines of printing are suppressed as in disk dump.

TABLE DUMP (FOR CHECK/STOP TRACE)

Calling the table dump function allows you to dump the addresses of the last ten executed instructions in a program. These addresses are stored in a table when the program is run in the trace mode; this table is updated by the Check/Stop Trace subroutine. Thus, if the trace mode is not used (console mode switch set to RUN), the table will contain no entries. The console mode switch must be set to RUN when the table is dumped.

On a table dump for Check/Stop Trace, the format is as follows:

```
TABLE DUMP ADDR-1  ADDR-2 ..............ADDR-10
SS AAAA QQQQ X1X1 X2X2 X3X3
```

where:

ADDR-1	is the address of the last instruction executed on the basic level.
ADDR-2	is the address of the next to last instruction executed on the basic level.
ADDR-10	is the address of the tenth from the last instruction executed on the basic level.
SS	is the status of the carry (first digit) and overflow (second digit) indicators before the execution of the last instruction.
AAAA	is the contents of the A-register before the execution of the last instruction.
QQQQ	is the contents of the Q-register before the execution of the last instruction.
X1X1	is the contents of index register 1 before the execution of the last instruction.
X2X2	is the contents of index register 2 before the execution of the last instruction.
X3X3	is the contents of index register 3 before the execution of the last instruction.

TRACE SUBROUTINES

The trace subroutines are internal to the IBM system and may be set up at any time via CONSOLE INTERRUPT. The subroutines are used to trace a given range of addresses (Full Trace), monitor an address for change (Monitor Trace), or perform a table dump if the instruction in a selected address is executed (Check/Stop Trace).

The setup and execution of the trace subroutines are independent; that is, the trace parameters may be entered at any time and the actual trace performed at a later time when the program is executed. However, only one trace option may be used at a time.

Only basic-level coreloads may be traced.

TRACE DEFINE LIMIT SUBROUTINE

The Trace Define Limit subroutine allows you to select any one of the three trace subroutines. To call the Trace Define Limit subroutine, set program switch 5 ON, program switch 7 OFF, and press CONSOLE INTERRUPT.

Pressing CONSOLE INTERRUPT causes the following message to be printed (unless otherwise indicated, each message is followed by a WAIT):

```
DATA SW0 ON FOR FULL TRACE
```

If a full trace is desired, set data switch 0 ON, press console START, and follow the instructions for "Full Trace."

If you do not want a full trace, leave data switch 0 OFF and press console START. The following message is printed:

```
DATA SW0 ON FOR MONITOR TRACE
```

If a monitor trace is desired, set data switch 0 ON, press console START, and follow the instructions for "Monitor Trace."

If you do not want a monitor trace, leave data switch 0 OFF and press console START. The following message is printed:

```
ADDR CHECK TRACE
DATA SWS = CHECK ADDR AAAA
```

You should now follow the instructions for "Check/Stop Trace."

Note: During the time the Trace Define Limit subroutine is in use, the MPX system is in a suspended state, that is, all levels are masked and the real-time clock is stopped. (If the clock was not stopped, the programmed timers might run out and all be waiting to interrupt as soon as the trace limits were entered and the levels were unmasked.

FULL TRACE

When you select the full trace option, the following message is printed:

```
DATA SWS = LOW TRACE ADDR  LLLL
```

Enter the low limit boundary of the trace in the data switches in hexadecimal, four switches per character, and press console START. The low limit of the trace (LLLL) is printed, followed by the following message:

```
DATA SWS = HIGH TRACE ADDR HHHH
```

Enter the upper limit boundary of the trace in the data switches in hexadecimal. If the trace is desired immediately, that is, the program to be traced is currently suspended from execution, set the console mode switch to TRACE, turn program switch 5 OFF, ensure that data switch 15 is OFF (it can be turned OFF just after pressing console START if data switch 15 is used for address), and press console START. If the program is to be traced at a later time, leave the console mode switch in RUN, turn program switch 5 OFF, and press console START.

The upper limit of the trace (HHHH) is printed, followed by the message:

```
DATA SW 15 ON SUPPRESSES PRINT
```

Without waiting, the Trace Define Limit subroutine returns control to the program that was in execution when CONSOLE INTERRUPT was pressed.

Notes:

1. The trace printout can be suppressed at any time by turning data switch 15 ON. To return the system to normal operation, press console STOP, set the console mode switch to RUN, and press console START.

2. A full trace should not be used with a FORTRAN or assembler language program containing a PAUSE statement. If a PAUSE is encountered while printing in the TRACE mode, the program will loop on itself.

The full trace format is as follows:

```
ADDR INS1 INS2 AAAA QQQQ SS X1X1 X2X2 X3X3
```

where:

ADDR	is the address of the next instruction to be executed.
INS1	is the first word of the next instruction to be executed.
INS2	is the second word of the next instruction to be executed.
AAAA	is the contents of the accumulator before the execution of the next instruction.
QQQQ	is the contents of the Q-register before the execution of the next instruction.
SS	is the status of the carry (first digit) and overflow (second digit) indicators before the execution of the next instruction.
X1X1	is the contents of index register 1 before the execution of the next instruction.
X2X2	is the contents of index register 2 before the execution of the next instruction.
X3X3	is the contents of index register 3 before the execution of the next instruction.

A sample output of a full trace run is listed below:

```
F496  F500  F4D5  0400  0002  11  000B  FEB0  FEB3
F498  4C18  F4A0  0C00  0002  11  000B  FEB0  FEB3
F49A  C031        0C00  0002  11  000B  FEB0  FEB3
F49B  71FF        0400  0002  11  000B  FEB0  FEB3
F49C  70F9        0400  0002  11  000A  FEB0  FEB3
F496  F500  F4D5  0400  0002  11  000A  FEB0  FEB3
F498  4C18  F4A0  0000  0002  11  000A  FEB0  FEB3
F4A0  1090        0000  0002  11  000A  FEB0  FEB3
```

MONITOR TRACE

The Monitor Trace subroutine checks a specified word of main storage after every instruction and notifies you if that word has changed in value since the instruction was executed. Thus, each time the specified word changes in value, monitor trace prints the value of the I-register before the change, after the change, and the new value of the monitored word.

Following this, the subroutine continues execution of the basic-level program and checks for a subsequent change in the value of the monitored word.

When the monitor trace option is selected, the following message is printed:

```
DATA SWS = MONITOR ADDR
```

Enter the address that is to be monitored for a change in the data switches in hexadecimal, four switches per character, turn program switch 5 OFF, and press console START (if an immediate trace is desired, set the console mode switch to TRACE before pressing console START).

The following message is printed:

```
MONITOR CHECK ADDRESS IS CCCC
```

Without waiting, the Trace Define Limit subroutine returns control to the program that was in execution when CONSOLE INTERRUPT was pressed.

The monitor trace format is as follows:

```
AAAA BBBB CCCC
```

AAAA	is the value of the I-register after the word changed.
BBBB	is the value of the I-register before the word changed.
CCCC	is the new value of the word.

CHECK/STOP TRACE

The Check/Stop Trace subroutine performs two functions, as explained below. The first function is performed only if data switch 15 is OFF during program execution.

1. If, during the execution of any basic-level program, the address of the next instruction to be executed is equal to the check/stop address, a table dump is performed followed by a WAIT. To continue execution, press console START. If a WAIT occurs without a table dump, the next instruction in the program being traced is a WAIT, and it may not reference the check/stop address. To continue, press console START. Note that if the check/stop address is branched to as a result of a BSI, or follows an XIO, the Check/Stop subroutine will not wait when the address is reached. BSI and XIO instructions cause the trace level to be masked for one instruction, thus missing the stopping address.

2. A table of the addresses of the last ten instructions that have been executed is maintained by check/stop trace. This table is dumped via the trace define limits subroutine. The mode switch must be set to TRACE when the address table is dumped.

When the check/stop trace option is selected, the following message will have already been printed:

```
DATA SW = CHECK ADDR AAAA
```

Enter the address to be checked in the data switches in hexadecimal (four switches per character), turn program switch 5 OFF and press console START. (If an immediate trace with both options is required, ensure that data switch 15 is OFF and set console mode switch to TRACE before pressing console START. Data switch 15 can be turned OFF just after pressing console START if it is used for the address.) The check address (AAAA) is printed followed by the following message:

```
DATA SW 15 ON SUPPRESSES ADDR CHECK
```

Without waiting, the Trace Define Limit subroutine returns control to the program that was in execution when CONSOLE INTERRUPT was pressed.

DISK PACK INITIALIZATION PROGRAM

The Disk Pack Initialization Program (DPIP) is a program in the Subroutine Library. You can build it into a batch-processing or process coreload and use it to initialize your 1316 disk packs.

DPIP has the same functions as BDPIP, which was described under "BOM Card Utilities." You can use DPIP by means of the following sequence of statements in the input stream:

```
1  // JOB
2  // XEQ DPIP
3  *CCEND
4  .
5  .   *INITLZ and *GETALT statements
6  .
7  *END DPIP
8
```

You can also, of course, use *STORECI to store the coreload so that you don't have to build the coreload every time you want to use it.

ONLINE DUMP PROGRAM

DEBUG

The Online Dump Program (OLDMP) is a program in the Subroutine Library. You can build it into a process or batch-processing coreload and use it to dump selected parts of main storage, 1810 drives, and 2311 drives to the list printer. The size of the partition for which the coreload is built must be at least 3600 words.

To build OLDMP into a batch-processing coreload and execute it, use the following sequence of control statements:

```
1  // JOB
2  // XEQ OLDMP
3  *CCEND
4
```

You can also store the coreload by using *STORECI and execute it in any way you choose.

At the beginning of its execution, OLDMP prints these messages:

```
ON-LINE DUMP PGM
SEN SW 0 CORE DUMP
SEN SW 1 DISK DUMP
SEN SW 2 TERMINATE
```

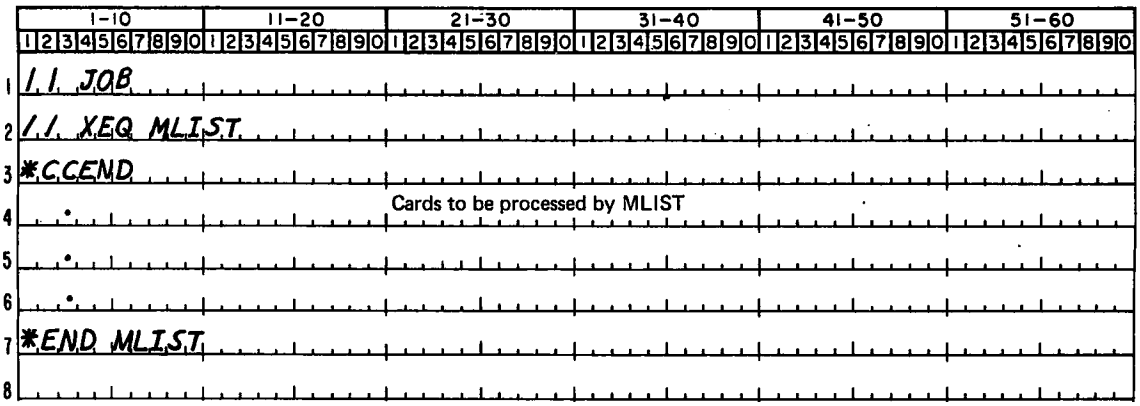
Set the sense switches as indicated in the messages and press console START.

The rest of the operating procedure and the format of the dump are identical to those described under "Disk Dump" and "Main-Storage Dump" in the section "BOM Utility Package." After the dump has been completed, the program prints the initial messages again, allowing you to carry out another dump operation. To stop execution of the program, at this point, when running under BOM, press console STOP, RESET, and START. To stop execution of the program, at this point, when running under the Executive (online), turn sense switch 2 ON and press console START.

80-80 LIST AND SEQUENCE CHECK PROGRAM

The 80-80 List and Sequence Check Program (MLIST) is a program in the Subroutine Library. You can build it into a batch-processing or process coreload and use it to list cards and/or to check their sequence numbers.

MLIST has the same functions as the BOM card utility BLIST. You can use MLIST by means of the following sequence of statements:



You can also store the coreload by using *STORECI and execute it in any way you choose.

1442 RELOCATABLE CARD DUMP PROGRAM (CRDMP)

This is a stand-alone eight-card relocatable program that is used for real-time or batch-processing system debugging. It dumps all of main storage to cards in a special compressed binary format for analysis by the Card Dump Analysis Program (DMPAN). DMPAN then produces a formatted dump on the list printer.

The Relocatable Card Dump Program occupies approximately 340 main-storage locations and can be loaded into any even main-storage address.

Note: If the Relocatable Card Dump Program overlays the CDW table in a partition, DMPAN will suppress the printing of all labels in that partition.

The 1442 Relocatable Card Dump Program is operated in the following manner:

1. Press IMMEDIATE STOP or STOP and RESET on the console.
2. Clear storage protect bits by performing a clear main storage operation with the console mode switch set to DISPLAY.
3. Set the Check/Stop switch ON.
4. Set the console mode switch to LOAD.
5. Set the loading address for the CRDMP program in the data switches in hexadecimal (this must be an even storage address).
6. Press console RESET and LOAD I.
7. Place the eight-card dump program in the card read punch hopper followed by blank cards and press reader START. (Note that the program does not check for blank cards prior to punching. Input cards must therefore be blank or checksum errors will occur when they are read by DMPAN.)
8. Set the console mode switch to SI W/CS.
9. Press console PROGRAM LOAD.
10. After the first card has been read, set the console mode switch to LOAD.
11. Press console RESET and LOAD I (the data switches are still set to the load address).
12. Set the console mode switch to RUN and press START.

The cards punched by CRDMP are stacker selected. After all of main storage has been dumped, the program comes to a WAIT with /XXXX+7 displayed in the I-register. /XXXX is the CRDMP program load address. At this point the main-storage-to-card dump program can be repeated by pressing console START. To return to system operation, perform a cold start.

Note: This program can be loaded into location 0 using the standard program load procedure.

CARD FORMAT

Cards are punched in a compressed binary format that allows 56 data words per card. The card formats are as follows:

FIRST CARD

Word	Contents
1	/FFFF
3	Starting address of dump program
4	Ending address of dump program

DATA CARDS

Word	Contents
1	Beginning main-storage address for data on card
2	Checksum (sum or data words 3-60)
3	Word Count for data words on card

Word	Contents
4-59	Data words
60	Sequence number

LAST CARD

Word	Contents
1	Zero
2	Checksum
3	/FFFF
4	Execution address
5-59	Not used

ERROR PROCEDURES

Punch errors during a dump cause the program to wait with /XXXX+15 displayed in the I-register. /XXXX is the CRDMP program load address. Remove the card in error (last card in stacker) and press console START to repunch that card.

Read errors during the loading of the CRDMP program cause a dynamic WAIT. The loading process must be reinitialized (restart at step 1).

If the card read punch becomes not ready, the program loops until the reader is made ready.

Card Dump Analysis Program (DMPAN)

The Card Dump Analysis Program serves as a diagnostic aid to label all pertinent areas of main storage. It can be used in conjunction with the 1442 Relocatable Card Dump Program to obtain a formatted dump of main storage, in which case the card output of the CRDMP program is the input to DMPAN, or can be used directly to dump any portion of main storage. In either case, the requirements of the dump are dictated by parameters in the DMPAN control statement. DMPAN provides you with the ability to acquire a main-storage map of your real-time or batch-processing system. DMPAN is stored in the User Library on disk.

DMPAN is loaded into, and therefore destroys the contents of, the first 5.1K words of VCORE. If you want your analysis to reflect the original contents of this part of main storage, you should use the 1442 Relocatable Card Dump Program (CRDMP) and use the card deck it produces as input to DMPAN. Otherwise, you may use DMPAN directly. You specify your options in a DMPAN control statement. Figure 13 illustrates the stacked input required for DMPAN.

DMPAN can be executed in minimum VCORE (5140 words) provided the number of executive transfer vectors required by your system does not exceed 35 (105 words).

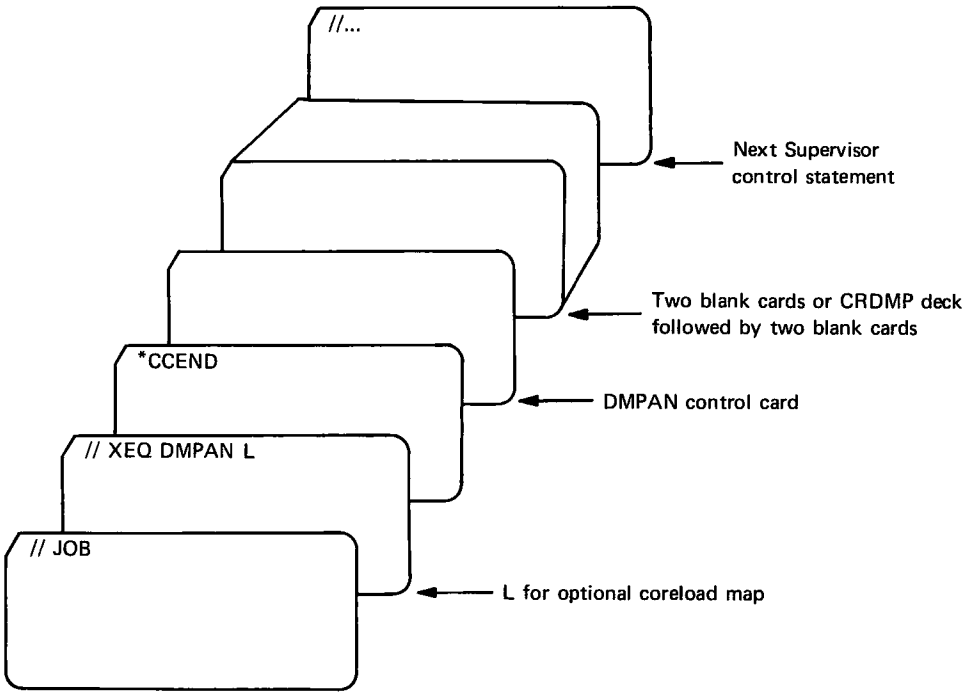


Figure 13. Job Stream for DMPAN

DMPAN CONTROL CARD FORMAT

The control card format for the Card Dump Analysis Program is shown in Figure 14.

Column	Meaning
1-4	Hexadecimal low main-storage address of area to be dumped. If no dumping is required, this parameter must be /FFF0.
6-9	Hexadecimal high main-storage address of area to be dumped.
11	Blank - Print all labels 1 - Print only those labels for the area to be dumped 2 - Do not print any labels
13	Blank (or zero) - Dump line interpreted in 1053 printer code 1 - 1443 printer code 2 - EBCDIC code 3 - Suppress line code interpretation
15	Blank (or zero) - Dump from cards 1 - Dump from main-storage
20-59	Any remarks. These remarks will appear in the page heading for each 1443 page.

Figure 14. DMPAN Control Card Format

Note: If columns 1-4 and 6-9 are all blank, all of main storage is dumped. Thus, a control card with a 1 in column 15 only gives a completely labeled dump of main storage on the list printer with each line interpreted in 1053 printer code.

The output of a dump analysis is a listing on the list printer of the specified main-storage dump which details all major Executive Director programs and tables, Executive I/O programs and tables, partitions and associated tables, and other areas of main storage as required, separately identified and labeled. Various control card options allow specification of start and end addresses of the area to be dumped, the type of printer character code required, and individual suppression of dumping and labeling. This program versatility enables you to request a main-storage map of the dumped system without a listing of the dumped area.

AREAS TO BE DUMPED

Depending on the particular system, the following programs, areas, and tables are labeled separately in the listing.

Fixed Area
Error Trap Area
INSKEL COMMON
All device tables
The following Executive I/O subroutines:

FILEN	IOSET	PUTQ
BULKN	IOTST	GETQ
TYPEN	RSAVE	EACRL
PRNTN	TVSET	ABORT
CEINT	BNDSH	DISKZ
IOEXT	IODRT	RINFO
IOSAV	IOERR	IINTB
TVSAV	DIRCL	RSTDK
TVEXT	STPRT	EACPT
QZSAV	LNKER	RELDO
QZEXT	EREND	ILEVE
BKSAV	PUTQ0	
BKEXT	GETQ0	

Error Message Table
BOM Utility Programs
*ICLT communications table
*Disk save area
*Queue table
*Interval timer tables
*Level coreload list control table
*Empty queue list table
*ICL Table
*Partition tables
*Program timer table
*All level work areas
*Real-time basic level work area
*Background basic level work area
*Interrupt branch table
*Boundary table
*Area busy table
*Start of MIC code
*I/O exit/entry in Executive Director
*Queueing entry in Executive Director
*CALL EXIT entry in Executive Director
*Suspend level entry in Executive Director
*CALL LINK entry in Executive Director
*End-background-processing entry in Executive Director
*FORTRAN I/O buffers in Executive Director
*User-included subroutines
*Executive CLNT
*Executive FIO table

*Executive TV table
*Executive branch table
**Partitions
**VCORE

*These entries appear only for a real-time system.
**For each partition, the following tables, if they exist, are labeled:

CDW	ETV	LPT
FIO	CLNT	DFT
IST (only in SPAR partitions)		

OPERATING PROCEDURES

The Card Dump Analysis Program is executed as a batch-processing relocatable program in VCORE under the control of the Batch-Processing Monitor in a real-time or batch-processing environment.

Depending on the state of the system at the time of an error, control of the Batch-Processing Monitor can be obtained in one of three ways.

1. Cold start to BOM.

2. Cold start to BPMON.

3. Press CONSOLE INTERRUPT with program switch 7 ON (in a real-time system).

Note: If the dump is directly from main storage, two blank cards must follow the control card. This notifies the system that the dump is from main storage instead of from cards.

OUTPUT FORMAT

The format of a dump line is identical to that of the dump format of the BOM Utility Monitor Dump Program, including the optional line interpretation into 1053 Printer, 1443 Printer, or EBCDIC code. The first and last lines of a labeled portion of the dump will list only those words that are within the area being dumped. The remaining words of these lines are filled with asterisks. This format simplifies recognition of the start address of a labeled portion. Each page of a 1443 printer dump will be numbered and can include a header containing your comments.

ERROR AND TERMINATION PROCEDURES

An invalid checksum (word 2) on an input card or a sequence number out of order initiates a terminal error procedure: an EACPT message is printed and the program waits. The card in error is the last card in the stacker. If data switch 15 is OFF when console START is pressed, the program continues to read cards from the point of error. This allows the card in error to be corrected or discarded. If data switch 15 is ON, the program reinitializes itself to read a completely new card deck. Cards with a sequence number of zero are not checksummed. Sequence numbers do not have to be consecutive.

SAMPLE DUMP ANALYSIS

An example of the use of the Card Dump Analysis Program is shown below. In this example, all of main storage was dumped, but only a representative sample of the output, interpreted in 1053 Printer Code, is included.

```
// JOB  09 FEB 70 10.438 HRS
// XEQDMPAN L   09 FEB 70 10.439 HRS
*CCEND


MPX, BUILD DMPAN

CORE LOAD MAP
TYPE NAME ARG1  ARG2

*CDW TABLE C26A  0012
*FIO TABLE C27C  001E
*CNT TABLE C29A  0004
MAIN DMPAN C29E
CLNT DMPAN C29C
CALL HXBIN D4A0
CALL BINHX D4EE
CALL CARDN D524        R
CORE       D6B0  2950
MPX, DMPAN LD XQ


MPX CORE DUMP ANALYSIS        OUTPUT ON 1443 FROM CORE  0000 - 0600                    EBCDIC   CODE      PAGE   1

FIXED AREA    0000  TO  012B

ADDR ***0 ***1 ***2 ***3 ***4 ***5 ***6 ***7 ***8 ***9 ***A ***B ***C ***D ***E ***F  -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F

0000 7024 4C00 210F 0000 0000 0000 FFAB 0001 3027 37AA 0000 3F28 3FA6 4024 40A2 4120 *.......................... . ...*
0010 419E 421C 429A 4318 4396 4414 4492 4510 458E 460C 468A 4708 4786 4804 4882 4900 *................................*
0020 497E 49FC 4A7A 4AF8 4B76 4C00 313E 0008 3970 0001 FFFF FFF6 1AC8 2000 0000 0480 *.=.....8.... ..........6.H.....*
0030 0000 0481 FFFF 0480 FFFF 0481 0018 4D21 1000 2607 4D9A 0000 39BF 39C4 3EA7 0D49 *.............(.....(......D....*
0040 0C3A 0930 8100 0001 00D0 0000 2A0D 1100 2000 3100 0000 4000 5100 0C2A 0C32 0000 *.....................* .....*
0050 0000 0000 0020 0000 0000 0000 2915 8000 0000 0000 0000 0000 0244 9B28 0005 FFCE *.....................KB.C.....*
0060 0003 4214 22BD 22B9 24A1 4BD2 C268 C31A 4B6E 277B 4C1F 5506 0028 0046 4C3D 7FDA *.....$.I.....................*
0070 000A 000B 000C 24CD 4C5B 21C9 2193 0001 238E 3994 000A 0018 113E 1CBD 15B6 0600 *..8..8.....................* &...*
0080 0500 F800 0FF8 00FF 8000 0001 0002 0004 0005 0007 0FFF 2000 0180 0140 509C 4F19 *.....B...O....E&.....4....*
0090 0141 C29A FF00 F000 FF87 37C5 5018 000D 7FFC 2D76 222A 2246 0406 01F4 185A 000F *.....O.....0.............*
00A0 0000 04A0 0000 04A1 1000 0F00 0001 0000 FFFF 00F0 000F 5129 21ED 220A 396E 013F *...0..+.....+.....M.......*
00B0 0000 FFF0 0000 4E6D 0000 3283 4E6C 8008 8010 15B6 1CBD 30CB 3AD4 0018 0019 0009 *.-..C.. .@.....6..........*
00C0 0006 2916 0002 0000 5095 0000 25D6 2301 0546 0000 0000 0000 0000 0000 0000 0000 *.B...*...6.............K.4....*
00D0 05C2 0616 065C 06A2 06F6 0000 0000 0000 073C 0768 0000 0000 07D2 07F4 082A 0790 *...-.............0.....B*
00E0 07B2 0860 0880 08A0 53BD 0000 0000 0546 04D8 046A 0438 0406 0000 0000 0000 05C2 *...*...6.....+..*
00F0 0616 065C 06A2 06F6 0000 0000 0000 8001 8002 8004 335F 2547 25A2 242D 244E 2464 *.....6.....6..........*
0100 0002 260E 0000 0000 0000 0000 0000 2263 2287 2525 24F6 29A6 2980 3004 230C 2323 *.....&....0.....*
0110 2360 5254 C31A 4440 807C 80B2 9DFE A9B6 B186 B762 0000 0000 0000 0000 0000 0000 *.....*
0120 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 **** **** **** *.....*

ERROR TRAP AREA         012D

0120 **** **** **** **** **** **** **** **** **** **** **** **** 00D2 0000 2E1C *.....K....*
0130 6050 B09C 2030 2121 3460 6050 6021 7034 9898 2121 5460 5014 2030 2121 5C5C 3C3C *-&.....--&-.....-&.....**..*
0140 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *.....*

INSKEL COMMON           0212

0400 0000 0000 0000 0000 0000 0000 6600 0414 4C00 0000 0000 0000 0001 0000 0000 0000 *.....*
0410 0002 18B0 5C58 C4F0 0000 0000 0000 0000 0000 C2E4 D3D2 F440 4040 4040 0000 0000 *....*.D0.....BULK4      ....*
0420 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 042E 0000 0001 0000 *.....*
0430 0000 0000 0000 0000 0000 0000 0000 0000 6600 0446 4C00 0000 0000 0000 0001 0000 *.....*
0440 0000 0000 0002 18B0 5C58 C4DC 0000 0000 0000 0000 0000 0000 C2E4 D3D2 F340 4040 4040 *.....*.D.....BULK3     *
0450 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0460 0000 *.....-..*
0460 0001 0000 0000 0000 0000 0000 0000 0000 0000 0000 **** **** **** **** **** **** *.....*

BULK02 DEVICE TABLE     046A

0460 **** **** **** **** **** **** **** **** **** **** 6600 0478 4C00 0000 0000 0000 *.....*
0470 0001 0000 0000 0000 0001 18B0 5C58 C4D8 0010 0000 0000 0000 0000 C2E4 D3D2 F240 *.....*.DQ.....BULK2 *
0480 4040 4040 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *     .....*
0490 0492 0000 0001 0000 0496 4E00 0008 0000 0000 0000 0000 0000 0000 0000 0000 0000 *.....+.....*
04A0 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 4E00 4D00 4C00 FFFE 0000 0000 *.....+.(.....*
04B0 0000 0000 0000 0000 0000 4F00 0000 4F01 0000 0000 04AE 04AF 0000 0000 0000 0000 *.....*
04C0 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 4580 0000 *.....*
04D0 0001 04B6 6600 0478 4C80 04CE 0000 0000 **** **** **** **** **** **** **** **** *...0.....*

BULK01 DEVICE TABLE     04D8

04D0 **** **** **** **** **** **** **** **** 6600 04E6 4C00 0000 0000 0000 0001 0000 *.....W.....*
04E0 0000 0000 0001 18B0 5C58 C4FC 0010 0000 0000 0000 0000 0000 C2E4 D3D2 F140 4040 4040 *.....*.D.....BULK1     *
04F0 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0500 0000 *.....*
0500 0001 0000 0504 4600 000B 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *.....*
```

```
MPX CORE DUMP ANALYSIS        OUTPUT ON 1443 FROM CORE  0000 - 0600                          EBCDIC   CODE      PAGE  2

ADDR ***0 ***1 ***2 ***3 ***4 ***5 ***6 ***7 ***8 ***9 ***A ***B ***C ***D ***E ***F  -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F

0510 0000 0000 0000 0000 0000 0000 0000 0000 4600 4500 4400 FFFE 0000 0000 0000 0000 *................................*
0520 0000 0000 0000 0000 4700 0000 4701 0000 0000 051C 051D 0000 0000 0000 0000 0000 *................................*
0530 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 4580 0001 0544 *................................*
0540 6600 04E6 4C80 053C 0000 0000 **** **** **** **** **** **** **** **** **** **** *...W............................*

BULK00 DEVICE TABLE     0546

0540 **** **** **** **** **** **** 6600 0554 4C00 1414 0000 0000 0001 0258 0000 0A3E *................................*
0550 0001 1880 5C58 C4C4 0010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *....*.DD........................*
0560 0000 0000 0000 0000 0000 0000 024B 0428 0000 0000 0258 0258 056E 2605 0001 025D *...............................)*
0570 0572 2600 000B 8000 0001 0000 0000 2000 0000 0000 0000 0000 0000 0000 D660 2600 *.............................O...*
0580 D558 2600 0047 0258 0047 0258 2600 2500 2400 0002 D52E 6680 0009 C201 0001 2400 *O.....................N.....B.....*
0590 0258 2700 0000 2701 FF06 0000 058C 058D D669 1306 0141 0248 C268 2600 0000 0000 *................O........8.......*
05A0 0001 0000 0000 0000 0000 FFCE 0000 0258 0000 0000 1493 4580 0001 05B2 6600 0554 *................................*
05B0 4C80 05AA 4B88 0000 000C 0000 4580 0001 05BD 6600 05D0 4C00 0000 0000 0000 0C02 *................................*
05C0 0000 0A02 **** **** **** **** **** **** **** **** **** **** **** **** **** **** *................................*

TYPE01 DEVICE TABLE     05C2

05C0 **** **** 6600 05D0 4C00 1967 0000 0000 0001 8000 0000 0000 0003 9CA4 5434 C4FC *............................:.D.*
05D0 0002 0000 0000 0000 0000 05DD 0902 05EE 0F02 FFFF 0F03 0000 0201 0029 8100 F245 0000 *.........................2....*
05E0 0000 0000 0000 1887 0000 0000 0000 0000 0000 10F0 05EB 0000 0000 0001 0000 0000 *...................0............*
05F0 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *................................*

TYPE02 DEVICE TABLE     0616
TYPE03 DEVICE TABLE     065C
TYPE04 DEVICE TABLE     06A2
TYPE05 DEVICE TABLE     06F6
PT1443 DEVICE TABLE     073C
CARD00 DEVICE TABLE     0768
1054   DEVICE TABLE     0790
1055   DEVICE TABLE     07B2
M-TAPE DEVICE TABLE     07D2
AI-BAS DEVICE TABLE     07F4
AI-EXP DEVICE TABLE     082A
DINPUT DEVICE TABLE     0860
DA-OUT DEVICE TABLE     0880
PLOTER DEVICE TABLE     08A0
2841CU DEVICE TABLE     0930
CA PHYS DEVC TABLE 0    0A7A
CA PHYS DEVC TABLE 1    0B0A
CA PHYS DEVC TABLE 2    0B9A
FILEN  SUBROUTINE       0C3A
BULKN  SUBROUTINE       113E
TYPEN  SUBROUTINE       15B6
NO RESPONSE SUBROUTINE  1AC8
PRNTN  SUBROUTINE       1CBD
CEINT  SUBROUTINE       210F
IOEXT  SUBROUTINE       2193
IOSAV  SUBROUTINE       21C9
TVSAV  SUBROUTINE       21ED
TVEXT  SUBROUTINE       220A
QZSAV  SUBROUTINE       222A
QZEXT  SUBROUTINE       2246
BKSAV  SUBROUTINE       2263
BKEXT  SUBROUTINE       2287
IOSET  SUBROUTINE       22B9
```

```
IOTST  SUBROUTINE          22DB
RSAVE  SUBROUTINE          2301
TVSET  SUBROUTINE          230C
BNDSH  SUBROUTINE          2323
IODRT  SUBROUTINE          2360
IOERR  SUBROUTINE          238E
DIRCL  SUBROUTINE          242D
STPRT  SUBROUTINE          244E
LNKER  SUBROUTINE          24A1
EREND  SUBROUTINE          24CD
PUTQO  SUBROUTINE          24F6
GETQO  SUBROUTINE          2525
PUTQ   SUBROUTINE          2547
GETQ   SUBROUTINE          25A2
EACRL  SUBROUTINE          25D6
APORT  SUBROUTINE          2607
DISKZ  SUBROUTINE          280E
RINFO  SUBROUTINE          2916
IINTB  SUBROUTINE          2980
PSTOK  SUBROUTINE          29A6
EACPT  SUBROUTINE          2D76
RELDO  SUBROUTINE          3004
ILEVE  SUBROUTINE          3027
ERROR  SG TABLE            30CB
BOM UTILITY SUBROUTINES    313E
ICLT COMMUNICATION TBL     3970
DISK SAVE AREA             398A
QUEUE TABLE                3992
INTERVAL TIMER TABLES      39BC
LEVEL CORE LOAD LCT        39F8
EMPTY QUEUE LIST TABLE     3AD4
ICLT TABLE                 3AD7
PARTITION TABLES           3CF3
PROG TIMER TABLE           3EA7
LEVEL WORK AREA  0         3F06
LEVEL WORK AREA  1         3F84
LEVEL WORK AREA  2         4002
LEVEL WORK AREA  3         4080
LEVEL WORK AREA  4         40FE
LEVEL WORK AREA  5         417C
LEVEL WORK AREA  6         41FA
LEVEL WORK AREA  7         4278
LEVEL WORK AREA  8         42F6
LEVEL WORK AREA  9         4374
LEVEL WORK AREA 10         43F2
LEVEL WORK AREA 11         4470
LEVEL WORK AREA 12         44EE
LEVEL WORK AREA 13         456C
LEVEL WORK AREA 14         45EA
LEVEL WORK AREA 15         4668
LEVEL WORK AREA 16         46E6
LEVEL WORK AREA 17         4764
LEVEL WORK AREA 18         47E2
LEVEL WORK AREA 19         4860
LEVEL WORK AREA 20         48DE
LEVEL WORK AREA 21         495C
LEVEL WORK AREA 22         49DA
LEVEL WORK AREA 23         4A58
REAL-TM BASIC LEV WK AR    4AD6
```



```
BACKGRD BASIC LEV WK AR    4B54
INTRPT BRANCH TABLE        4BD2
BOUNDARY TABLE             4C1F
AREA BUSY TABLE            4C30
START OF MIC CODE          4C3D
I/O EXIT ENTRY (EXIT1)     4C5B
QUEUING ENTRY (CNQUE)      4D9A
CALL EXIT ENTRY (EXIT)     4E6C
SUSPEND LEVEL ENTRY        4F19
CALL LINK ENTRY (LINK)     509C
TSUBX SUBROUTINE           5129
FORTRAN I/O BUFFERS        53BA
USER INCLUDED SUBRTNS      5506
EXECUTIVE CLNT             7FDA
EXECUTIVE FIO TABLE        7FDE
EXECUTIVE BRANCH TABLE     7FFC
PARTITION  1    8000 TO 9D4B
CDW    TABLE    8002
BLANK CDW TABLE
PARTITION  2    9D4C TO A903
CDW    TABLE    9D4E
BLANK CDW TABLE
PARTITION  3    A904 TO B0D3
CDW    TABLE    A906
BLANK CDW TABLE
PARTITION  4    B0D4 TO B6AF
CDW    TABLE    B0D6
BLANK CDW TABLE
PARTITION  5    B6B0 TO C267
CDW    TABLE    B6B2
BLANK CDW TABLE
VARIABLE CORE   C268 TO FFFF
CDW    TABLE    C26A
FIO    TABLE    C27C
ETV    TABLE    C29A
CORELOAD PROGRAMS          C29E
END OF CORE                FFFF
```

Optional Material Tape Dump Program

The Optional Material Tape Dump Program (MOPTP) provides you with the ability to punch, list, or punch and list selected modules from the optional material tape. MOPTP is stored in the User Area and is executed as background-processing job. The desired module and option are specified in control cards that you punch. Figure 14.1 illustrates the stacked input required for MOPTP.

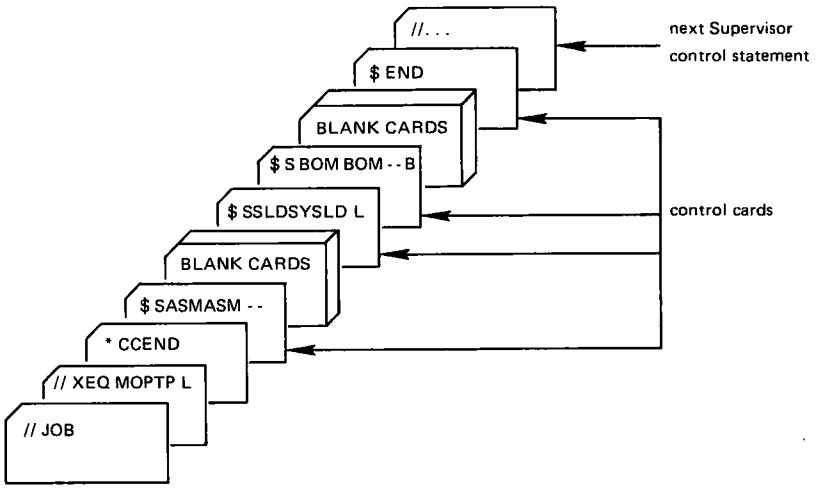


Figure 14.1 Example of Stacked Input for MOPTP

MOPTP Control Card Format

The control card format for the Optional Material Tape Dump Program is shown in Figure 14.2

Column	Meaning
1-10	Name of the desired module. These begin with $S, and all unused positions must be filled with dashes (for example: $SASMASM--). An index of the modules is supplied with the optional material tape.
12	Output Option L - List only B - Punch and list Blank or any other character - Punch only

Note: The last control card must contain $END in columns 1-4 MOPTP to exit.

Operating Procedures

The Optional Material Tape Dump Program is executed as a batch- or background-processing program in VCORE.

The following steps are required to execute the Optional Material Tape Dump Program:

1. Punch the required Supervisor control cards (Figure 14.1).
2. Punch the MOPTP control cards required to select the desired module and output option (Figure 14.2).
3. Place the optional material tape on either of the tape drives. The program will pause to allow specification of the drive selected.

4. Place the Supervisor and MOPTP control cards in the card reader (Figure 14.1) and ready the reader. Note that modules may be selected in any sequence but if modules are requested in any order other than that shown in the index, tape rewind may occur before processing of some of the modules.
5. Request execution of the Batch-Processing Monitor.
6. The control cards will be read. When MOPTP has begun execution, the following messages are printed on the list printer. (The program pauses with /0003 in the accumulator):

   SENSE SW 5 SELECTS TAPE DRIVE.
   OFF FOR DRIVE 0, ON FOR DRIVE 1.

   Set sense switch 5 as required and press START. The MOPTP program will read the provided control cards and dump the selected modules as directed.

Output Format

The output of the Optional Material Tape Dump Program is punched cards, a listing, or both, in 1800 assembler source format. The cards are punched on 1442-0 and are acceptable as input to the 1800 assembler. Listings are printed on the list printer, which may be an 1816/1053 or a 1443.

Error and Termination Procedures

Termination of the Current Module

Turning on sense switch 3 while a module is being dumped will cause dumping of that module to be terminated and initiate reading of the next control card.

I/O Error Retry

All I/O errors will cause a pause with /0909 in the accumulator. Pressing START with sense switch 4 OFF will cause a retry of the I/O operation. In the case of a tape read error. a backspace will be performed first. Pressing START with sense switch 4 ON will cause the program to exit.

MOPTP Errors Detected

The following errors are detected by the MOPTP program. (All messages are printed on the list printer):

- INVALID CONTROL CARD
  A control card containing one of the following errors was read:

  Column 1 does not contain a $.
  Column 2 is neither S nor E.
  Columns 3-10 contain a blank.
  Column 11 is not blank.

  After printing the message, the program pauses with /0001 in the accumulator. It will continue reading control cards if START is pressed.

- NEED BLANK CARDS
  A nonblank card was detected during punching. The program pauses with /0002 in the accumulator. It retries the punch operation if START is pressed.

- UNABLE TO LOCATE $SXXXXXXXX
  The end of tape has been reached without locating the selected module. The program rewinds the tape and continues reading control cards.

Trap Subroutines

The trap subroutines are provided as real-time diagnostic aids that allow you to transfer the contents of a main-storage area to a buffer without stopping the system. If, in a real-time system, you do not want the system to stop when an EAC error occurs (sense switch 6 OFF), you may determine the cause of the error by using the trap subroutines.

Most programs do not have the built-in ability to retain error information. Thus, any information that could be obtained from a post-mortem dump would be of questionable value. It is therefore desirable to have a procedure that will give a dump of the affected area of main storage after the error occurs, but before the error information is lost. This objective is accomplished by "trapping" the error, and transferring the error information to a buffer. At some later time, this information can be dumped to an output device for closer analysis.

To achieve the data transfer, a CALL TRAP is made defining the error type, the error condition, and the name of the coreload in which the error occurs. The CALL TRAP statement can be issued from any coreload within your system. Once the error occurs, a CALL DTRAP can be executed from any coreload in the system to dump the area of the main storage specified in the CALL TRAP parameters on the system printer. The number of words trapped is dependent on the size of the trap buffer defined by the BOM equate cards TBUFS at system generation.

The TRAP and DTRAP subroutines reside in the User Library, while the trapping mechanism exists within EAC in the Executive I/O. The calling sequences for the trap subroutines are described in the Subroutine Library manual, Order Number GC26-3724.

The following example illustrates the use of the trap subroutines. Assume the following coreload, called SEEK1, has been built and is called for execution

```
// JOB 00.911 HRS
// ASM SEEK1
*LIST


0000 0  0818          START XIO     DATA       SENSE DATA SWS
0001 01 4C100000            BSC  L  START,-     LOOP IF DATSW 0 IS OFF
0003 0  1001                SLA     1          IF ON--SHIFT OUT
0004 0  0000                DC      0
0005 0  1801                SRA     1            DATSW 0 AND
0006 0  D018                STO     ARSA+1     STORE IN I/O AREA(SEC ADD)
0007 30 02913495            CALL    BULKN       SEEK TO
0009 1  0012                DC      LIST              SEC ADD
000A 0  C007                LD      LIST       TEST FOR
000B 01 4C20000A            BSC  L  *-3,Z               BUSY
000D 0  C80E                XIO     DATA       SENSE DATA SWS
0008 01 4C28000D            BSC  L  *-3,+Z       LOOP IF DATSW 0 IS ON
0010 01 4C000000            BSC  L  START
                      ********************************
                      *         BULKN I/O LIST       *
                      ********************************
0012 0  0000          LIST  DC      0          LINK/BUSY
0013 0  0000--              DC      0          TYPE-1 EXIT
0014    0004                BSS     4          SYSTEM RESERVED PARAM
0018 0  0000                DC      0
0019 0  5000                DC      /5000      SEEK CONTROL PARAM
001A 1  001E                DC      AREA       I/O AREA ADDRESS
001C    0001          DATA  BSS  E  1          IOCC TO
001D 0  0740                DC      /0740            SENSE DATA SWS
001E 0  0140          AREA  DC      320        WD COUNT
001F 0  0000                DC      *-*        SEC ADD
0020    0140                BSS     320
0160    0000                END     START

    NO ERRORS IN ABOVE ASSEMBLY.
SEEK1
DMP FUNCTION COMPLETED
// XEQ SEEK1 L
*CCEND


MPX, BUILD SEEK1

CORE LOAD  MAP
TYPE NAME  ARG1  ARG2

*CDW TABLE 61AA  0012
*FIO TABLE 61BC  001E
*CNT TABLE 61DA  0004
MAIN SEEK1 61DE
CLNT SEEK1 61DC
CORE       6340  1CC0
MPX, SEEK1 LD XQ
```

As soon as the system attempts to execute the DC 0, the following op-code error occurs and the system performs a restart:

```
00.676  I-LEVL  OP-CODE ER  SEEK1  FF18 80000 61E3 007F 344A 625A RS
```

The system restart effectively destroys any information that could be obtained by a main-storage dump. The error data must then be trapped. To trap the error data, write a CALL TRAP program to trap on the parameters listed in the EAC printout. The starting address of the program is found on the coreload map (in this case it's the address MAIN SEEK1 61DE). The trap program can be included in any coreload. When the trap program is loaded and executed, it sets up indicators in the system that cause the error data to be trapped when the error occurs.

The following program, called TRAP1, will trap the op-code error generated by the execution of SEEK1.

```
// JOB 00.899 HRS
// ASM TRAP1
*LIST


0000 30 236415C0     START CALL     TRAP
0002 1  0014               DC       LEVAR
0003 1  0009               DC       MESS1
0004 1  000C               DC       MESS2
0005 1  0011               DC       MESS3
0006 0  61DE               DC       /61DE
0007 30 059C98C0           CALL     EXIT
0009    0006       MESS1 DMES     1-LEVL'E
000C    000A       MESS2 DMES     OP-CODE ER'E
0011    0006       MESS3 DMES     SEEK1 'E


0014 0  FF18       LEVAR DC       /FF18
0016    0000             END      START

    NO ERRORS IN ABOVE ASSEMBLY.
TRAP1
DMP FUNCTION COMPLETED
// XEQ TRAP1 L
*CCEND


MPX, BUILD TRAP1

CORE LOAD  MAP
TYPE NAME  ARG1  ARG2

*CDW TABLE 61AA  0012
*FIO TABLE 61BC  001E
*CNT TABLE 61DA  0004
MAIN TRAP1 61DE
CLNT TRAP1 61DC
CALL TRAP  61FR        R
CORE       6276  1DBA
MPX, TRAP1 LD XQ
```

Now if SEEK1 is executed, the error data is stored in the trap buffer. To read the trap buffer write a program, in this case called TRAPD, to call DTRAP. This program can be included in any coreload; however, the most logical place is the restart coreload for the partition in which the error occurs.

The following program will dump the op-code error data for analysis. The output is on the system printer:

```
// JOB 00.925 HRS
// ASM TRAPD
*LIST


0000 30 048D9057      JO    CALL    DTRAP
0002 30 059C98C0            CALL    EXIT
0004    0000                END     JO

    NO ERRORS IN ABOVE ASSEMBLY.
TRAPD
DMP FUNCTION COMPLETED
// XEQ TRAPD L
*CCEND


MPX, BUILD TRAPD

CORE LOAD  MAP
TYPE NAME  ARG1  ARG2

*CDW TABLE 61AA  0012
*FIO TABLE 61BC  001E
*CNT TABLE 61DA  0004
MAIN TRAPD 61DE
CLNT TRAPD 61DC
CALL DTRAP 61E3
CORE       62B4  1D4C
MPX, TRAPD LD XQ
```

The system printer output is listed below:

```
ADDR  ***0  ***1  ***2  ***3  ***4  ***5  ***6  ***7
61D8  ****  ****  ****  ****  ****  ****  081B  4C10
61E0  61DE  1001  0000  1801  D018  4480  007C  61F0
61E8  0007  4C20  618E  080E  4C28  61EB  4C00  61DE
61F0  0000  0000  0000  0000  0000  0000  0000  5000
61F8  61FC  0000  0000  0740  0140  0000  0000  0000
6200  0000  0000  0000  0000  0000  0000  0000  0000
6208  0000  0000  0000  0000  0000  0000  0000  0000
6210  0000  0000  0000  0000  0000  0000  0000  0000
6218  0000  0000  0000  0000  0000  0000  0000  0000
6220  0000  0000  0000  0000  0000  0000  0000  0000
6228  0000  0000  0000  0000  0000  0000  0000  0000
6230  0000  0000  0000  0000  0000  0000  0000  0000
6238  0000  0000  0000  0000  0000  0000  0000  0000
6240  0000  0000  0000  0000  0000  0000  0000  0000
6248  0000  0000  0000  0000  ****  ****  ****  ****
```

For comparison purposes, the same area of main storage was dumped before program execution by the BOM Utility Monitor.

```
61D0 7005 0000 0000 0000 0000 0000 0000 7005 0000 2214 54B1 0000 0000 081B 4C10 *M..............M....GP........F*
61E0 61DE 1001 0000 1801 D018 4480 007C 61F0 C007 4C20 61E8 080E 4C28 61EB 4C00 61DE *..F....B.68&,.J.4...I.............*
61F0 0000 0000 0000 0000 0000 0000 0000 5000 61FC 0000 0000 0740 0140 0000 0000 0000 *..............0..1.....S.S.......*
6200 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *................................*
```

CE Coreload Programs (CECLD,CECLX)

Two CE Coreload programs, CECLD and CECLX, enable you to print out and modify the status of I/O devices on the system. CECLX includes extensions that can do 2311 disk storage drive, communications adapter, and 2790 data communication system functions additional to the functions of CECLD.

The CE Coreload programs are stored in the User Library on disk. CECLD requires approximately 1000 words of main storage and CECLX requires approximately 5000 words of main storage. The functions available with CECLD and CECLX are:

- Set On/Offline status
- Reset hardware error count
- Set logical and physical device assignments for disk storage units, 1053 printers, and communications adapters
- Read out and reset Executive Director error counts

A 2311 extension of CECLX enables you to do the following:

- Dump and reset the 2311 Error Statistics Table
- Dump and reset the CE Error Log

A communications adapter extension in CECLX enables you to do the following:

- Dump and reset the Error Statistics Table
- Dump and reset the CE Error Log
- Initialize the communications a lapter line trace buffer
- Dump communications adapter line trace buffer

A 2790 extension of CECLX enables you to do the following:

- Dump and reset the 2790 error counts
- Dump and reset the 2790 error log
- Start a 2790 loop segment

Initial entry to CECLX is to the basic program, but special CECLX branch functions allow transfer of control to:

- The 2311, 2790, or communications adapter extension, from either the basic program or the other extension
- The basic program, from either extension
- The Batch-Processing Monitor Supervisor, from either the basic program or its extensions

Both coreloads handle various kinds of errors by producing coded displays in the A-register accompanied by attempts to type a message on the system printer (unsuccessful attempts if the printer has been taken off line).

HOW TO BUILD CE CORELOADS

Each CE Coreload program may be built into any kind of coreload. In a real-time system, a CE Coreload program is usually executed as a process coreload, queued to service the programmed interrupt which the system generates when the CE INTERRUPT button is pressed. The interrupt level and ILSW bit associated with the CE INTERRUPT button are determined during BOM assembly by the CELVL and CEBIT equate cards.

If the CE Coreload is built to service an interrupt, the interrupt should probably be assigned to a low level so that it won't disrupt normal operation of the system.

In a batch-processing system, the CE Coreload is executed as a batch-processing coreload under control of the Batch-Processing Monitor.

HOW TO USE A CE CORELOAD

You specify CE Coreload functions using the CE sense switches, which are located on the CE panel directly below the Processor-Controller console. You initiate each function by pressing console START. The system executes a WAIT when each function is completed.

When execution of the CE Coreload begins, the following message is printed:

```
SET FUNC IN CE SWITCHES
```

Unless you intend to branch to communications adapter, 2311, or 2790 extensions of CECLX, your first function must be Select Device Type.

CE CORELOAD FUNCTIONS

The following functions can be done by both CE Coreloads (CECLD and CECLX):

SELECT DEVICE TYPE

1. Set CE sense switches to 0010 XXXX, where XXXX has one of the following values.

Setting of XXXX	Selected Device
0000	1810 disk storage unit
0001	1816/1053 printer
0010	1443 printer
0011	1442 card read punch
0100	2401 magnetic tape
0101	AI - basic
0110	AI - expander
0111	1054 paper tape reader
1000	1055 paper tape punch
1001	DI
1010	DAO
1011	1627 plotter
1100	communications adapter
1101	2311 disk storage drive

2. Press console START.

For each logical device of the unit type specified, the following items are typed out:

- Logical device number
- Physical unit identification
- On/Offline status
- Hardware error count, if applicable.

This page intentionally left blank.

SET ON/OFFLINE STATUS

1. Set CE sense switches to:

   0100 0YYY to take the unit off line
   0101 0YYY to put the unit on line

where:

YYY	is the logical device number of the unit (for 1053s use the logical device number minus one).

2. Press console START.

The new status of the logical unit is typed out.

RESET ERROR COUNT

1. Set CE sense switches to 0110 0YYY.

where:

YYY	is the logical device number of the unit (for 1053s use the logical device number minus one).

2. Press console START.

The new status of the logical unit is typed out. Note that you should reset the error count at system generation to ensure that it is initialized properly before you use your system.

SWITCH LOGICAL UNIT ASSIGNMENTS

This function is only applicable to 1810 disk storage units, 2311 disk storage drives, 1053/1816 printers, and communications adapters.

1. Set CE sense switches to 11 XXX YYY.

where:

XXX	is the physical device number (in binary) to be assigned to the logical device number YYY (for 1053s use the physical device number minus one).
YYY	is the logical device number to be assigned to the physical device XXX (for 1053s use the logical device number minus one).

2. Press console START.

The new status of the logical unit is typed out.

READ OUT OR RESET EXECUTIVE ERROR COUNTS

1. Set CE sense switches to 1010 000X.

where:

X = 0 to type out Executive error counts

X = 1 to reset all error counts.

2. Press console START.

The error counts are not typed out for the reset function. Note that you should reset these error counts at system generation to ensure that they are set properly before you use your system. You can terminate any CE coreload dump function at any time by changing the setting of the CE switches.

EXIT FROM CORELOAD

1. Set CE sense switches to 0000 0000.

2. Press console START.

The following is printed where XXXX is the error count.

		LABEL in EX-DIR
NON-PROCESS SAVE	XXXX	CWTD
NON-PROCESS READ	XXXX	CRTSP
QUEUE AREA READ	XXXX	RQCE
QUEUE TABLE FULL	XXXX	QFULL
COMMON VCORE READ	XXXX	EIRD
INTERRUPT SAVE	XXXX	SERR
INTERRUPT RESTORE	XXXX	CABR
BP MONITOR READ	XXXX	CRMON
SPECIAL SAVE	XXXX	BDRML

Error Word	Description
CWTD ------	BULKN detected an error writing VCORE to the NPSV.
CRSTP------	BULKN detected an error reading the NPSV area to VCORE.
RQCE ------	BULKN detected an error while reading a queued coreload into a partition.
QFULL-----	QLEVL returned to MIC an error parameter other than 1, indicating the coreload queue was full, or the entry was already in the queue.'
EIRD-------	BULKN detected an error when reading an interrupt coreload, or a coreload queued to the basic level.
SERR-------	BULKN detected an error writing VCORE to the interrupt save area (INSV).
CABR-------	BULKN detected an error reading the interrupt save area into VCORE.
CRMON-----	BULKN detected an error reading the batch-process monitor bootstrap loader into VCORE.
BDRML-----	BULKN detected an error writing VCORE to the special save area.

This setting causes a CALL EXIT to be executed.

ERROR MESSAGES

If the principal 1053 and all its backup units are offline, each attempt by the CE Coreload to type a message will cause a WAIT with a unique display in the A-register. These waits and their associated messages are as follows:

Message	A-register
	/F001 (See Note)
(Type out of CE switch settings)	/F002
CE CORELOAD	/F003
SET FUNC IN CE SWITCHES	/F004
DEVC OR UNIT NOT ON SYST	/F005
INVALID DEVICE CODE	/F006
INVALID DEVICE FOR SWITCH	/F007
NO DEVICE SELECTED	/F008
TURN ALL SWITCHES OFF TO EXIT	/F009
LOGICAL UNIT NO. TOO LARGE	/F00A
INVALID FUNCTION	/F00B
(Executive Director error count)	/F010
OFF LINE SYST-FUNC IGNORED	/F013
CHANGE TO CONTINUE	/FFFF
(Status line for device unit)	/0001
SYST PTR ERR-DUMP ABORTED	/8001

Note: /F001 is displayed when a value is to be set in the CE switches (following SET FUNC IN CE SWITCHES).

The Q-register displays the most recent setting of the CE sense switches in bits 8 - 15. Bit 0 of the Q-register is set to 1 if the requested function was successfully completed.

CECLX BRANCH FUNCTIONS

The following functions enable you to branch to the CECLX communications adapter, 2311, or 2790 extensions. Each branch function can be executed by either the basic part of the CECLX program or the other extension.

Branch to Communications Adapter Extension

1. Set CE sense switches to 0000 0001.
2. Press console START.

This page intentionally left blank.

Branch to 2311 Extension

1. Set CE sense switches to 0000 0010.
2. Press console START.

Branch to 2790 Extension

1. Set CE sense switches to 00000011.
2. Press console START.

CECLX COMMUNICATIONS ADAPTER EXTENSION FUNCTIONS

The communications adapter extension of CECLX is entered through the basic program or one of the extensions of CECLX using the Branch to Communications Adapter Extension function described above. This extension enables you to do the following:

Dump Error Statistics Table

1. Set CE sense switches to:

   0010 ZYYY to dump without reset
   0011 ZYYY to dump and reset

where:

Z = 0 and YYY = logical device number for the table, or
Z = 1 to dump and/or reset the Error Statistics Tables for all communications adapter units (YYY is not used).

2. Press console START.

The dump is printed on the system printer.

Dump CE Error Log

1. Set CE sense switches to:

   0100 0000 to dump without reset
   0101 0000 to dump and reset

2. Press console START.

To terminate the dump and cancel the reset functions, change any of the switch settings during the dump. The dump is printed on the system printer.

Set Up to Start/Stop CA Line Trace

1. Set CE sense switches to:

   0110 0YYY to set up to start trace, where YYY is the logical unit number of the CA line

   0111 0YYY to set up to stop trace, where YYY is not used

2. Press console START.

Dump Communications Adapter Trace Buffer

1. Set CE sense switches to 1000 0000.
2. Press console START.

The dump is printed on the system printer.

This function should not be selected unless you have first initialized the trace buffer by the CECLX function "Start CA Line Trace." Note that the contents of the trace buffer will be destroyed by the use of the CE diagnostics or your own use of CE Core.

Return to Basic Part of CECLX

1. Set CE sense switches to 1110 0000.

2. Press console START.

Error Messages

Message	A-Register
SELECT CA FUNCTION	/5004
INVALID FUNCTION FOR CA	/5005
DEVICE OR UNIT NOT ON SYSTEM	/5006
ERR STATISTICS TBL NOT DEFINED	/5007
SYS PTR ERR — DUMP ABORTED	/8001
(When a value is to be set in the CE sense switches)	/5001

CECLX 2311 EXTENSION FUNCTIONS

The 2311 extension of CECLX is entered through the basic program or one of the extensions using the "Branch to 2311 Extension" function. This extension enables you to do the following functions:

Dump Error Statistics Table

1. Set CE sense switches to:

   0010 ZYYY to dump without reset
   0011 ZYYY to dump and rest

where:

Z = 0 and YYY = logical device number for the table, or
Z = 1 and YYY to dump and/or reset the Error Statistics Tables for all units of the type chosen (YYY is not used).

2. Press console START.

The dump is printed on the system printer.

Dump CE Error Log

1. Set CE sense switches to:

   0100 0000 to dump without reset
   0101 0000 to dump and reset

2. Press START.

To terminate the dump and cancel the reset function, change any of the switch settings during the dump. The dump is printed on the system printer.

Return to Basic Part of CECLX

1. Set CE sense switches to 1110 0000.

2. Press console START.

Error Messages

Message	A-Register
SELECT 2311 FUNCTION	/6004
INVALID FUNCTION FOR 2311	/6005
DEVICE OR UNIT NOT ON SYSTEM	/6006
ERR STATISTICS TBL NOT DEFINED	/6007
SYS PTR ERR — DUMP ABORTED	/8001
(When a value is to be set in the sense switches)	/6001

CECLX 2790 EXTENSION FUNCTIONS

The 2790 extension of CECLX is entered through the basic program or one of the extensions by the "Branch to 2790 Extension" function. This enables you to do the following operations:

Dump 2790 Error Counts

1. Set the CE sense switches to:

   0010000x to dump without reset
   0011000x to dump and reset area station error counts

where:

x = 0 for loop number 1
x = 1 for loop number 2

2. Press Console START

The following message is printed:

ENTER AREA ATN ADDR

3. Set the CE sense switches to the area station address in hexadecimal (80-FF) or to 0, which indicates that the error counts for all area stations are to be printed. The system error count for that loop will also be printed if the sense switch setting was 0 or /FF (system error counts not reset).

4. Press console START. The error counts are printed on the system printer.

A minus sign printed before the count means that a threshold error has occurred during this terminal time out period.

NOT ACTIVE printed instead of the loop system error count means that the 2790 coreload is not in main storage or the loop is not active.

Dump 2790 Error Log (can be used as an error sort)

1. Set the CE sense switches to:

   01000000 for loop 1
   01000001 for loop 2

2. Press console START

The following message is printed:

ENTER AREA STN ADDR

3. Set the CE sense switches to the area station address in hexadecimal (80-FF) or to 0, which indicates that the error log entries for all area stations are to be printed.

4. Press console START. The error log entries are printed on the system printer.

2790 Error Sort Program

In addition to the CE Coreload dump of the 2790 Error Log, which can be used to sort errors by area station number, an Error Sort Program is available for this purpose.

The Error Sort Program is part of the 2790 On Line Diagnostic Programs which the system requires to be stored on drive 0 of your system. They are provided to you by your customer engineer with control cards that store them as a batch-processing coreload. You must assure that this program is maintained on your system residence disk.

After the AREA STATION THRESHOLD EAC message (see "Recovery From Errors" in the 1800/2790 MPX Data Communication System Programming manual, Order Number GC26-3732) has been printed, the errors in the error log must be sorted so that you may determine if a faulty data entry unit needs to be replaced or if an area station needs maintenance.

You can sort the errors by executing the Error Sort Program as a batch-processing job as follows:

```
          1-10       11-20      21-30      31-40      41-50      51-60
     1234567890 1234567890 1234567890 1234567890 1234567890 1234567890
1    // JOB
2    // XEQ @CEX@  FX
3    *// EDIT  LAAA  99
4    *
5    *
6    *
7    *// END
8
```

where L is the loop number (1 or 2) and AAA is the area station address (0-126).

Additional EDIT cards may be run to obtain error sorts on additional area stations if you want.

The output of the Error Sort Program is a listing of the last eight errors logged against the requested area station, printed in this format:

```
// JOB   07 JUL 70 09.489 HRS
// XEQ @CEX@  FX   07 JUL 70 09.490 HRS
DATE  188
            POLLING FRAME       ERROR FRAME
 TIME  DATE AS/DV CC/DA  CSW    AS/DV CC/DA LOOP TR

09.397 188 /82C0 /0E1F  /0400 /82C0 /3E00 /01 /00
09.397 188 /82C0 /0E1F  /0400 /82C0 /3E00 /01 /00
09.396 188 /82C0 /0E1F  /0400 /82C0 /3E00 /01 /00
09.348 188 /82C0 /0E1F  /0400 /82C0 /3E00 /01 /00
09.348 188 /82C0 /0E1F  /0400 /82C0 /3E00 /01 /00
09.347 188 /82C0 /0E1F  /0400 /82C0 /3E00 /01 /00
09.347 188 /82C0 /0E1F  /0400 /82C0 /3E00 /01 /00
09.342 188 /82C0 /0E1F  /0400 /82C0 /3E00 /01 /00
END OF ERROR LOG
```

Refer to the 2790 Adapter section of the 1800 Functional Characteristics manual, Order Number GA26-5918, for a detailed description of the Device Address, Channel Control Word, and Channel Sense Word.

The error sort information can be used to determine what action is required. If the device address (DV) values for four or more errors indicate the same device, the action described in this table is recommended:

DV Value (polling frame)	Device	Action
00	No Device	No device selected, check DV value in error frame for possible action.
40	Printer	Check for NO PAPER EAC message. Have personnel at machine location check for visible typewriter malfunctions.
80	Local Badge Reader	Have supervisory personnel verify operation of the suspect device. If the test proves the device to be malfunctioning, it should be taken out of service and repair service notified.
81	1st Remote Badge Reader	
82	2nd Remote Badge Reader	
83	3rd Remote Badge Reader	
84	Card Reader	
88	Keyboard	
8C	Digital Device (OEM)	
CO-DF	DEU	Decode the specific data entry unit (DEU) address from the DV value and have that unit tested or replaced according to established DEU service procedures.

Reset the 2790 Error Log

1. Set the CE sense switches to:

   01010000

2. Press console START.

The following message is printed:

2790 ERROR LOG RESET REQUESTED. IF OK,
TURN ON CE SENSE SWITCH 15 AND PRESS START

3. Set CE sense switch 15 ON. Do not change the other sense switches.

4. Press console START

Zeros will be written on each sector of the 2790 error log file (E2790).

Start/Reset a 2790 Segment

1. Set the CE sense switches to:

   011ABCD0 to start segments on loop 1
   011ABCD1 to start segments on loop 2

where:

A, B, C, or D is 0 to start the respective segment or 1 if the segment is not to be started.

2. Press console START.

The following message is printed on the system printer:

2790 SEGMENT START COMPLETION CODE = /00XX

If the completion code is /0001, the start was performed successfully. Note that this function will also restore a bypassed area station. A completion code of /0002-/001F indicates an error condition. A description of these completion codes can be found under "Completion Code Parameter for LACCN" in the 1800/2790 MPX Data Communication System Programming manual, Order Number GC26-3732.

Certain error conditions cause an Area Station to go into Send Address Mode. If you attempt to restart a segment where this condition exists the address of the Area Station in Send Address Mode is reported with the following message:

AREA STN IN SEND ADDR NODE IS /XX

Set On/Offline Status

1. Set the CE sense switches to:

1000000Y to take loop off line
1000001Y to put the loop on line

where:

Y = 0 for loop 1
Y = 1 for loop 2

2. Press console START.

The following message is printed on the system printer:

YOU REQUESTED LOOP X ON/OFF. IF OK TURN SW 11 AND PUSH START.

If switch 11 is turned on the following message is printed:

COMPLETION CODE /00XX

If the completion code is /0001, the function was performed successfully. A completion code of /0002 - /001F indicates an error condition. See the 1800/2790 MPX Data Communication System Programming manual, Order Number GC26-3732, for a description of these completion codes.

Return to Basic Part of CECLX

1. Set the CE sense switches to 1110 0000.

2. Press console START.

ERROR MESSAGES

If the 1053 (that is defined to be either the system and/or list printer) and all of its backup units are offline, each attempt by CECLX to type a message will cause a WAIT with a unique display in the A-register. These waits and their associated messages are as follows:

Message	A-Register
	/7001 (See Note)
SELECT 2790 FUNCTION	/7004
INVALID FUNCTION FOR 2790	/7005
ENTER AREA STN ADDR	/7006
2790 ERR LOG FILE NOT DEFINED	/7007
DISK ERROR	/7008
2790 ERROR LOG RESET REQUIRED. IF OK, TURN ON CE SENSE SWITCH 15 AND PRESS START.	/7009
..RESET COMPLETED	/700A
SYS PTR ERR -DUMP ABORTED	/8001

Note: /7001 is displayed when a value is to be set in the CE sense switches.

This page intentionally left blank.

EXAMPLES OF CE CORELOAD OPERATION

Figure 15 is an example of how to build and use CECLD. In the example, CECLD is built into a batch-processing coreload and used to take a 1053 printer off line and assign another printer to cover for it.

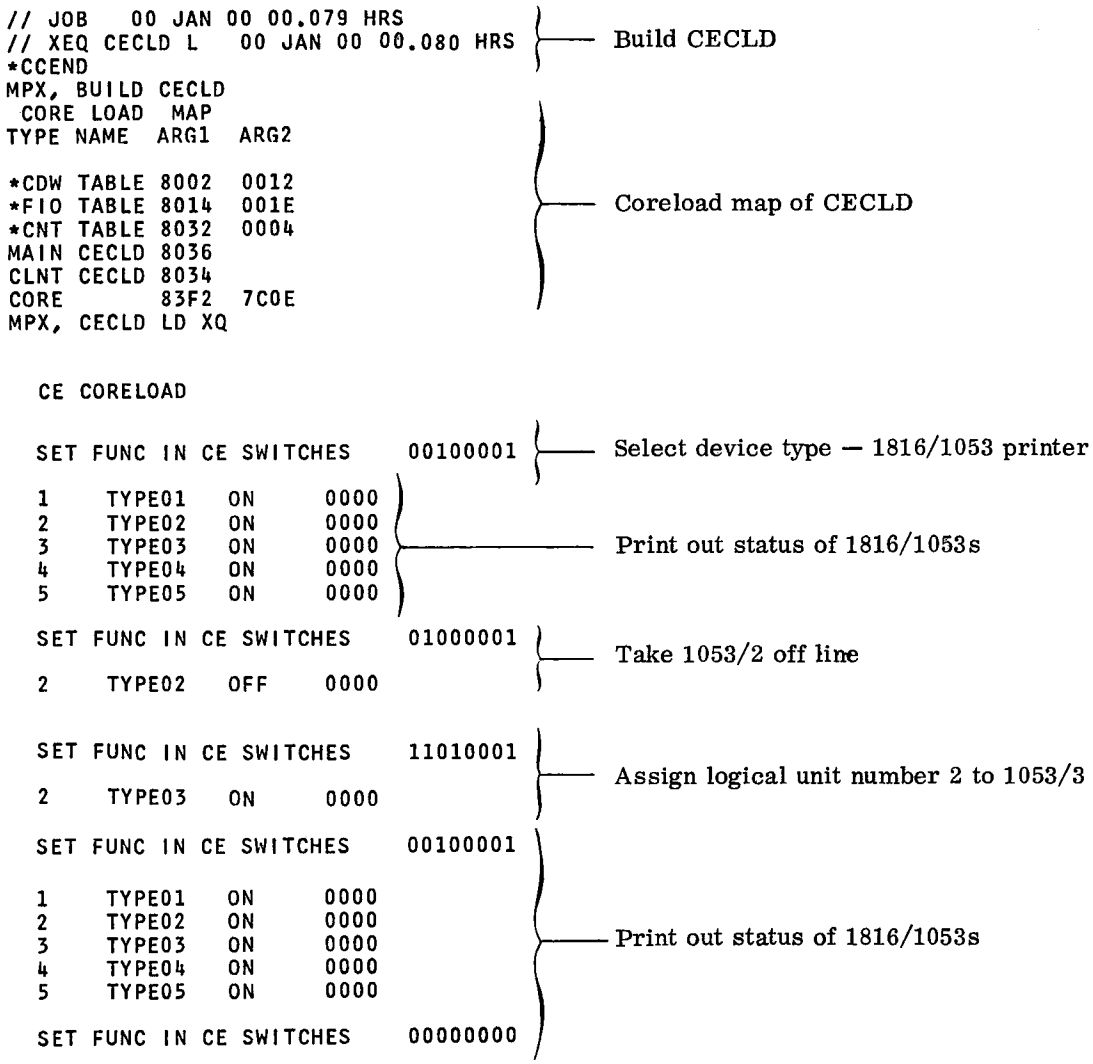


Figure 15. Using CECLD

Figure 16 is an example of how to build and use CECLX. In the example, CECLX is built into a batch-processing coreload and used to print the Error Statistics Tables and CE Error Logs for two communications adapters, two 2311 drives, and a 2790 data communication system with two loops.

```
// JOB   00 JAN 00 00.204 HRS
// XEQ CECLX L   00 JAN 00 00.204 HRS
*CCEND
MPX, BUILD CECLX
 CORE LOAD  MAP
TYPE NAME  ARG1  ARG2

*CDW TABLE 8002  0012
*FIO TABLE 8014  001E
*CNT TABLE 8032  0004
MAIN CECLX 8036
CLNT CECLX 8034
CALL EBPRT 8D72
CALL EBPRT 8D72
CALL PRT   8E1C
CORE       8E68  7198
MPX, CECLX LD XQ
```

Build CECLX

Coreload map of CECLX

```
 CE CORELOAD

 SET FUNC IN CE SWITCHES    00000001
 SELECT CA FUNCTION         00101000
```

Dump EST

```
  ERROR STATISTICS TABLE FOR CA LINE NO. 0, DAY   000, HOUR  17
      HARDWARE  - NONE
      MASTER    - NONE
      SLAVE     - NONE

  ERROR STATISTICS TABLE FOR CA LINE NO. 1, DAY   000, HOUR  00
      HARDWARE  - NONE
      MASTER
  CODE    COUNT
  /0031   /0091
  /0035   /0042
  /0036   /0004
  /0037   /001D
      SLAVE
  CODE    COUNT
  /0041   /000E
  /0046   /0004
  /0047   /0003

  ERROR STATISTICS TABLE FOR CA LINE NO. 2, DAY   000, HOUR  65
      HARDWARE
  CODE    COUNT
  /0005   /0007
      MASTER
  CODE    COUNT
  /0031   /0001
  /0036   /0007
  /0037   /0002
      SLAVE     - NONE
```

EST dump

```
SELECT CA FUNCTION        01000000
 COMMUNICATIONS ADAPTER CE ERROR LOG      DAY 000, HOUR  00, MSGS LOGGED  006

   TIME      NO. OF  OPERATING  FUNCTION   ERROR
 DAY HOUR    ERRORS      DSW       CODE     CODE

044    14    /0007     /2080     /2501     /0041
044    14    /0006     /10A0     /2501     /0043
044    14    /0001     /08A0     /2501     /0042
044    14    /0001     /10A0     /2501     /0043
044    14    /0001     /08A0     /2501     /0042
044    14    /0001     /10A0     /2501     /0043
```

Print CA CE error log

CA CE error log

Figure 16. Using CECLX

```
SELECT CA FUNCTION      10000000 ———————————————————— Print CA trace

   LINE 4
   DSW   1000    FUNCTION CODE   1304    COMPLETION CODE   0001
R  0000
   DSW   10A0    FUNCTION CODE   2204    COMPLETION CODE   0001
R  1001   2DFF
   DSW   2080    FUNCTION CODE   2504    COMPLETION CODE   0041
T  E004   1070   FF00
R  0000          (TIMEOUT)
   DSW   10A0    FUNCTION CODE   2504    COMPLETION CODE   0001
T  E004   1070   FF00
R  0053   1002   C3C1   D9C4  . . .  F840    C9F9   C9F9   267F
   DSW   10A0    FUNCTION CODE   2504    COMPLETION CODE   0001
T  E004   1061   FF00
R  0053   1002   C3C1   D9C4  . . .  F840    C9F9   C9F9   26FF
   DSW   10A0    FUNCTION CODE   2504    COMPLETION CODE   0023
T  E004   1070   FF00
R  0001   377F
   DSW   08A0    FUNCTION CODE   3704    COMPLETION CODE   0001
T  2004   1037   FF00
 END OF TABLE
```

CA trace dump *

```
SELECT CA FUNCTION      0 0000010 ———————————————— Specify a 2311 function

SELECT 2311 FUNCTION    01000000 ———————————————— Print 2311 CE error log

 2311 DISK FILE CE ERROR LOG              DAY 001, HOUR  00, MSGS LOGGED  014

    TIME     NO. OF   DEVICE    COMMAND   CHANNEL        UNIT SENSE BYTES          2311       DESIRED
  DAY HOUR   ERRORS    UA/US   FLAG/CODE  STATUS     0/1      2/3      5        ARM/HEAD     CC/TT

  043  14    /0006    /A102     /001D     /4000    /0040    /00C8    /0500     /0A03
  043  14    /0001    /A102     /001D     /4000    /8000    /00C8    /0500     /2509
  043  14    /0006    /A102     /4031     /4000    /0002    /00C8    /0000     /1601
  043  14    /0007    /A102     /0006     /4000    /0400    /00C8    /0600     /5700
  043  14    /0002    /A102     /0005     /4000    /0800    /00C8    /0500     /8205
  043  14    /0004    /A102     /0005     /4000    /2000    /00C8    /0500     /6308
  043  14    /0031    /A102     /4007     /4000    /0100    /00C8    /0000     /5003     /5103
  043  14    /0008    /A102     /4031     /4000    /0008    /00C8    /0000     /B507     /B407
  043  14    /0014    /A102     /4007     /4000    /1000    /00C8    /0000     /4502
  043  14    /0005    /A20C     /4007     /5000    /0000    /0000    /0000     /2107
  043  14    /0002    /A20C     /4007     /1000    /0000    /0000    /0000     /5301
  043  14    /0001    /A000     /0000     /0080    /0000    /0000    /0000     /7900
  043  14    /0001    /A000     /0000     /0800    /0000    /0000    /0000     /9305
  043  14    /0001    /A000     /0000     /0400    /0000    /0000    /0000     /A702
```

2311 CE error log

The column labeled 2311 ARM/HEAD contains the arm and head position at the time of the error. This is determined by reading the home address after the error.

The column labeled DESIRED CC/TT contains the cylinder and track position that was specified by the failing CCW.

```
SELECT 2311 FUNCTION    00101000 ———————————————— Print 2311 EST

 ERROR STATISTICS TABLE FOR 2311 UNIT NO. 0, DAY  000, HOUR  00
 CODE    COUNT
 /0001   /0000011B
 /0002   /0003
 /0008   /0005
 /000B   /0003
```

2311 EST dump

Figure 16. Using CECLX (Cont.)

* Lines of the trace dump labeled R are received data. A T indicates transmitted data. The first word is the indicator/byte count word. The remaining words contain the data transmitted over the line. Where the message length exceeds 14 characters, an ellipsis indicates characters omitted from the middle of the message. The function code shown in the trace dump is from the BSCIO call list, not the actual machine IOCC.

Table A. Communications Adapter Completion and Error Codes

This table lists all completion and error codes used by the BSCIO subroutine and indicates where each is used or recorded.

Completion code	Trace area	Retry counter	Error Statistics Table	CE Error Log	EAC message		Description
X	X					/0001	Operation successful
X						/0002	Device off line
X					X	/0003	Modem not ready
X	X	X	X	X		/0004	Data parity
X	X	X	X	X		/0005	Parity control
X	X	X	X	X		/0006	Data check
X	X	X	X	X		/0007	Data overrun
X					X	/0008	No-response
X	X					/0009	Modem failure
X						/0011	List aborted-abort discontinue
X						/0012	Initial not done
X						/0013	Wrong state of line
X						/0014	ID function not done
X						/0015	Two RVIs in a row
X						/0016	List aborted by restart
X	X					/0021	RVI received
X	X					/0022	DISC received
X	X					/0023	EOT received
X	X					/0024	EOT received to WACK
X	X	X				/0025	ENQ received to ENQ-Contention
X	X					/0026	Poll address received
X	X					/0027	Selection address received
X	X	X	X	X		/0031	Timeout as master
X	X	X	X	X		/0032	Invalid sequence received
X	X	X	X	X		/0033	Message resulted in negative response
X	X		X	X		/0034	Forward abort by slave
X	X	X	X			/0035	WACK received
		X	X			/0036	TTD sent
			X			/0037	Message transmitted successfully
X	X	X	X	X		/0041	Timeout as slave
X	X	X	X	X		/0042	Invalid sequence received
	X		X	X		/0043	Retransmission requests received
X	X		X	X		/0044	Forward abort by master
X	X	X	X			/0045	TTD received
		X	X			/0046	WACK sent
			X			/0047	Messages received successfully
	X		X			/0050	Message aborted - (Data..ENQ)
	X				X	/0000	Storage protect
					X		Command reject
					X		Call error

Table B. 2311 Error Conditions

Condition	Retries	Error Code	CE Error Log	Error Statistics Table Word	EAC Message
Interface Control Check	1	RS/RL	Yes	2	Yes
Channel Data Check	1	RS/RL	Yes	2	Yes
Not Operational	1	RS/RL	Yes	2	Yes
Equipment Check	0	RS/RL	No	3	Yes
No Record Found	n	9	Yes	No	Yes
Seek Check	n	8	Yes	8	Yes
Intervention Required	0	3	No	No	Yes
Bus Out Check	n	4	Yes	4	Yes
Data Check	n	6	Yes	6	Yes
Overrun	n	7	Yes	7	Yes
Missing Address Marker	n	RS/RL	Yes	9	Yes
Command Reject	0	RS/RL	No	No	Yes
Track Overrun	0	5	No	5	Yes
End of Cylinder	0	A	No	No	No
File Protect	0	D	No	No	No
Overflow Incomplete	0	RS/RL	No	A	Yes
Program Check	n	RS/RL	Yes	D	Yes
Unit Exception	0	C	No	C	No
Incorrect Length	0	B	No	B	No

Table C. 2841/2311 Command Codes

COMMAND		COMMAND CODE						DATA ADDRESS	COUNT
		Single Track Operations			Multiple Track Operations				
		Decimal	Hexadecimal	Binary	Decimal	Hexadecimal	Binary		
Control	No Operation	03	03	0000 0011				X	Not Zero
	Seek	07	07	0000 0111				CPU storage location seek address	6
	Seek Cylinder	11	0B	0000 1011					6
	Seek Head	27	1B	0001 1011					6
	Set File Mask	31	1F	0001 1111				CPU storage location of mask byte	1
	Space Count	15	0F	0000 1111				X	Not Zero
	Transfer in Channel	X8	X8	XXXX 1000				CPU storage location of next CCW (Must be divisible by 8)	X
	Recalibrate	19	13	0001 0011					
	Restore	23	17	0001 0111				X	Not Zero
Sense	Sense I/O	04	04	0000 0100				CPU storage location to which six sense bytes are sent	6
Switching	Release Device	148	94	1001 0100				X	Not Zero
	Reserve Device	180	B4	1011 0100				X	Not Zero
Search	Home Address Equal	57	39	0011 1001	185	B9	1011 1001	CPU storage location of search argument	4 (usually)
	Identifier Equal	49	31	0011 0001	177	B1	1011 0001		5 (usually)
	Identifier High	81	51	0101 0001	209	D1	1101 0001		5 (usually)
	Identifier Equal or High	113	71	0111 0001	241	F1	1111 0001		5 (usually)
	Key Equal	41	29	0010 1001	169	A9	1010 1001		From 1 to 255
	Key High	73	49	0100 1001	201	C9	1100 1001		From 1 to 255
	Key Equal or High	105	69	0110 1001	233	E9	1110 1001		From 1 to 255
	Key and Data Equal*	45	2D	0010 1101	173	AD	1010 1101		Number of bytes (including mask bytes) in search argument
	Key and Data High*	77	4D	0100 1101	205	CD	1100 1101		
	Key and Data Equal or High*	109	6D	0110 1101	237	ED	1110 1101		
	Continue Scan Equal*	37	25	0010 0101					
	Continue Scan High*	69	45	0100 0101					
	Continue Scan Equal or High*	101	65	0110 0101					
	Continue Scan, No Compare*	85	55	0101 0101					
	Continue Scan, Set Compare*	117	75	0111 0101					
Read	Home Address	26	1A	0001 1010	154	9A	1001 1010	CPU storage location to which areas read will be transferred	5
	Count	18	12	0001 0010	146	92	1001 0010		8
	Record R0	22	16	0001 0110	150	96	1001 0110		Number of bytes to be transferred
	Data	06	06	0000 0110	134	86	1000 0110		Number of bytes to be transferred
	Key and Data	14	0E	0000 1110	142	8E	1000 1110		Number of bytes to be transferred
	Count, Key and Data	30	1E	0001 1110	158	9E	1001 1110		Number of bytes to be transferred
	Initial Program load (IPL)	02	02	0000 0010					Number of bytes to be transferred
Write	Home Address	25	19	0001 1001				CPU storage location from which areas to be written will be transferred	5
	Record R0	21	15	0001 0101					8+Key Length+Data Length of Record R0
	Count, Key and Data	29	1D	0001 1101					8+Key Length+Data Length
	Special Count, Key and Data*	01	01	0000 0001					8+Key Length+Data Length
	Data	05	05	0000 0101					Data Length
	Key and Data	13	0D	00C0 1101					Key Length + Data Length
	Erase	17	11	0001 0001					8 + Key Length + Data Length

* Special Feature
X Not Significant

Table D. 2841/2311 Status and Sense Bytes

	CSW Status Byte	Sense Byte 0	Sense Byte 1	Sense Byte 2	Sense Byte 3 2311	Sense Byte 4	Sense Byte 5*
Bit 0	Attention	Command Reject	Data Check in Count Area	Unsafe	Ready	0	
Bit 1	Status Modifier	Intervention Required	Track Overrun	not used	On Line	0	
Bit 2	Control Unit End	Bus-Out Parity	End of Cylinder	Serializer Check	Unsafe	0	
Bit 3	Busy	Equipment Check	Invalid Sequence	not used	not used	0	See Note
Bit 4	Channel End	Data Check	No Record Found	ALU Check	On Line	0	
Bit 5	Device End	Overrun	File Protected	Unselected File Status	End of Cylinder	0	
Bit 6	Unit Check	Track Condition Check	Missing Address Marker	not used	not used	0	
Bit 7	Unit Exception	Seek Check	Overflow Incomplete	not used	Seek Incomplete	0	

*Sense Byte 5 is used only for the overflow feature. The setting of the bits is determined by the conditions existing at the time of the interruption and the type of operation in progress. For further information on the bit configurations, refer to the description in the text of the manual.

This page intentionally left blank.

```
SELECT 2311 FUNCTION        00000011

SELECT 2790 FUNCTION        00110000

ENTER AREA STN ADDR         00000000

  2790 ERROR COUNT  LOOP NO. 1

  AREA STN  COUNT
     /80    0600
     /81    0601
     /82    0614
     /83    0603
     /84    0605
     /85    0601
     /86    0600

  END--COUNT PRINTED ONLY IF DIFFERENT FROM PRECEDING.


  SELECT 2790 FUNCTION      01000000

  ENTER AREA STN ADDR       00000000

    2790 ERROR LOG  AREA STN=ALL   DAY 155

                 POLLING FRAME            ERROR FRAME
   DATE   TIME   AS/DV   CC/DA   CSW    AS/DV   CC/DA  LOOP  TRAN
    155  08.213 /8140   /0D80  /4000   /8140   /0D40  /01   /00
    155  08.212 /8140   /0D80  /4000   /8140   /0D40  /01   /00
    155  08.212 /8140   /0D80  /4000   /8140   /0D40  /01   /00
    155  08.212 /8140   /0D80  /4000   /8140   /0D40  /01   /00
    155  08.210 /8140   /0D80  /4000   /8140   /0D40  /01   /00
  END OF ERROR LOG

SELECT 2790 FUNCTION        01010000

2790 ERROR LOG RESET REQUESTED.  IF OK,
TURN ON CE SENSE SW 15 AND PRESS START.  RESET COMPLETED.
SELECT 2790 FUNCTION        01100000

  2790 SEGMENT START, COMPLETION CODE /0001
```

Figure 16. Using CECLX (Cont.)

1442 Reader Diagnostic

The MPX system contains a test to diagnose 1442 card read punch read errors and stacker select while the real-time system is operating. This program is part of the Supervisor, Phase SUPD. The 1442 must be taken off line by the CE Coreload before the test can be run.

When the 1442 is taken off line by the CE Coreload, the following message is printed on the system printer, and background processing is terminated:

```
// ERROR END OF ALL JOBS
```

OPERATING PROCEDURES

1. Clear the 1442 card reader.

2. Turn program switch 7 ON, CE switch 15 ON, and press CONSOLE INTERRUPT.

3. The following messages are printed:

```
CUST ENG 1442 C002 CE SWS ALL OFF TO EXIT
CUST ENG 1442 C001 LOAD TEST + MAKE READY
CUST ENG 1442 E002 READER NOT READY
```

4. Load your punched reader test complement cards in the reader and press reader START. Do not put a blank card behind the test cards.

   The cards for the reader test must be punched so the first 40 columns complement the second 40 columns. For example, if there is a 12 punch in column 1, column 80 must contain an 11 through 9 punch. If column 2 is laced (all punches), column 79 must be blank. A complement bit pattern must be used for all complementary pairs of columns on the card. Figure 17 illustrates a typical reader test complement card.

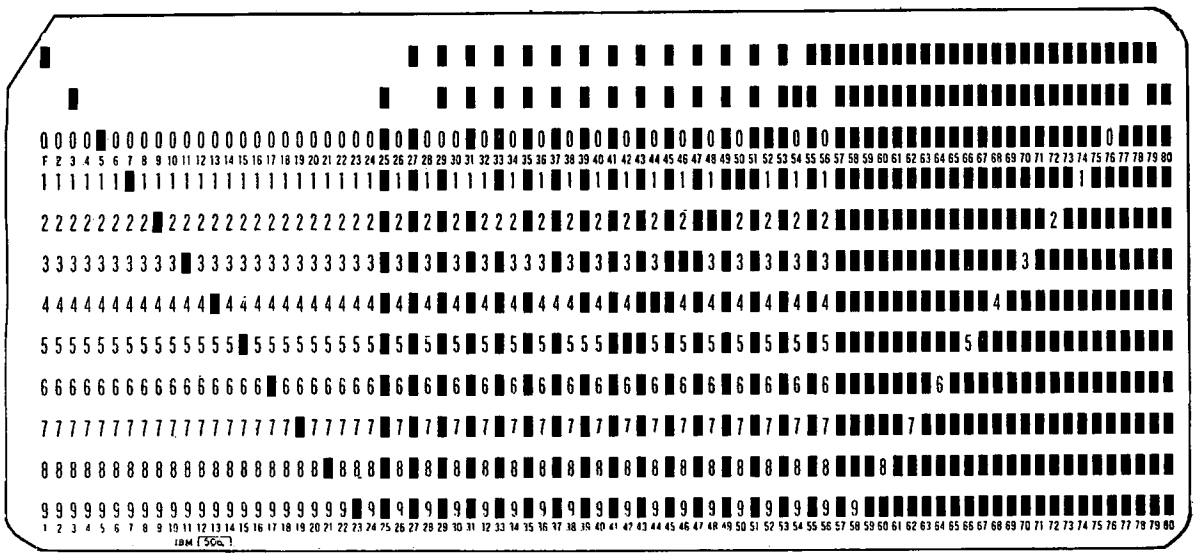


Figure 17. Sample Complement Input Card for 1442 Reader Diagnostic

When the complement cards are placed in the reader and the reader is made ready, the reader diagnostic issues a call to bring the reader on line, read a card, and put the reader back off line. The card that was read is first checked for picking up a digit in column 40 or 41 and then checked for dropping a digit in these same two columns. Columns 1 and 80 are the last columns checked. Any time an error is detected, it is typed as an E000 message (see "1442 Diagnostic Error Messages").

To suppress printing of all messages, turn CE switch 8 ON.

5. Every other card that is read is stacker selected. Following the reading of the complement cards, the program will again print the reader not ready message. To read the last card, press reader START. The following message is printed:

```
CUST ENG 1442 C003 READER LAST CARD
```

6. To return the reader to online status, turn CE switch 15 OFF and call the CE Coreload, or if the CE Coreload was loaded from cards, do a cold start.

1442 DIAGNOSTIC ERROR MESSAGES

```
CUST ENG 1442 E000 DIGIT XX PICKED COL YY OR YY
```

or

```
CUST ENG 1442 E003 DIGIT XX DROPPED COL YY OR YY
```

(XX is the card digit that was picked or dropped during reading. YY is the column in which the error was found.)

```
CUST ENG 1442 E001 READER OFF-LINE
```

(Reader was placed online by diagnostic but was found offline by read.)

```
CUST ENG 1442 E002 READER NOT READY
```

(The reader became not ready following the reading of the last card.)

```
CUST ENG 1442 E003 READER PARITY ERR
```

(A parity error occurred during the last read command.)

```
CUST ENG 1442 E004 READER FEED CHECK
```

(A feed check occurred during the last read command.)

```
CUST ENG 1442 E005 READER READ CHECK
```

(A read register check occurred during the reading of the last card.

```
CUST ENG 1442 E006 INVALID CONDITION
```

(CARDZ returned on invalid error parameter.)

This page intentionally left blank.

Appendix A. System Loader Assignment Cards

The assignment card is used to specify the I/O device and machine function assignments to interrupt levels and bit position on the level. The assignments are in the form of Interrupt Assignment Codes (IAC), which are fixed for each device, and Logical Unit Numbers (LUN), which are selected by you for linkage for the FORTRAN programs. You must punch the assignment cards.

An Interrupt Assignment Code (IAC) has been assigned by IBM for all possible devices. These assignments cannot be changed. Assignment cards for interrupt levels greater than 23 should not be made, except for the special use of 99. If a value greater than 23 and other than 99 is punched in the assignment card, an error message is printed and the assignment is ignored.

The possible assignment values for the FORTRAN Logical Unit Numbers (LUN) are 01 through 44. The same LUN cannot be assigned to more than one device nor can a device have more than one LUN assigned to it.

Table 1 lists IAC/LUN assignments for the MPX system. The format of the assignment card is shown in Table 2.

Notes:

1. Continuation of bit assignments for a level onto another card is done as follows:

   Place any nonblank character in column 72.

   Place the total number of IAC codes assigned for the level in column 4 of all cards for the level, for example:

   ```
   cc   1   4    7                          72
        99  04   24, 25/20, 27               X
        99  04   44
   ```

2. DMON and all its phases must be the first module loaded in a //TOTAL SYSTEM LOAD operation. If an *LDDSK card and an *ASSIGNMENT card are both being used in the same load or update operation, the *LDDSK must follow the *ASSIGNMENT.

Table 1. IAC/LUN Assignments for MPX System

Device	IAC	Possible LUN Values
Interval timers	00	No LUN assignable
First 1816/1053 on printer group 1	01	01 through 44
First 1442 card read punch	02	01 through 44
1054/1055 paper tape units	03	01 through 44
First 1810 disk or first mapped 1810**	04	01 through 44
1627 plotter	05	01 through 44
1443 printer	06	01 through 44
First 2790 adapter	07	No LUN assignable
Second 1810 disk or second mapped 1810**	08	01 through 44
Third 1810 disk or third mapped 1810**	09	01 through 44
First analog-to-digital converter (ADC)	10	No LUN assignable
Digital input	11	No LUN assignable
Digital analog output	12	No LUN assignable
System/360 channel adapter	13	No LUN assignable
First magnetic tape	14	01 through 44
First 1816/1053 on printer group 2	15	01 through 44
Analog input expander, second analog-to-digital converter	16	No LUN assignable
Second 1442 card read punch	17	01 through 44
2841 control unit	18	No LUN assignable
Second 2790 adapter	19	No LUN assignable
Fourth communications adapter line 0	20	No LUN assignable
First communications adapter line 0	21	No LUN assignable
Second communications adapter line 0	22	No LUN assignable
Third communications adapter line 0	23	No LUN assignable
Fourth mapped 1810*	24	01 through 44
Fifth mapped 1810*	25	01 through 44
Sixth mapped 1810*	26	01 through 44
Seventh mapped 1810*	27	01 through 44
Eighth mapped 1810*	28	01 through 44
RPQ***	29-31	No LUN assignable
Console interrupts	32	No LUN assignable
Process interrupts	33	No LUN assignable
First comparator	34	No LUN assignable
Second comparator	35	No LUN assignable
Second 1053 on printer group 1	36	01 through 44
Third 1053 on printer group 1	37	01 through 44
Fourth 1053 on printer group 1	38	01 through 44
Second 1053 on printer group 2	39	01 through 44
Third 1053 on printer group 2	40	01 through 44
Fourth 1053 on printer group 2	41	01 through 44
1816 keyboard on printer group 1*	42	01 through 44
1816 keyboard on printer group 2*	43	01 through 44
Magnetic tape drive 2*	44	01 through 44
First communications adapter line 1	45	No LUN assignable
Second communications adapter line 1	46	No LUN assignable
Third communications adapter line 1	47	No LUN assignable
Fourth communications adapter line 1	48	No LUN assignable
RPQX***	49-63	No LUN assignable

*indicates that a dummy interrupt level number 99 is required.
**indicates that a dummy interrupt level number 99 is required if a mapped 1810 is assigned.
***RPQX is an extension of RPQ.

Table 2. Assignment Card Format

Card Column	Possible Values	Meaning
1-2	00 through 23, or 99	The entry in this field specifies the interrupt level to which this card is applicable. A separate card must be made for each interrupt level used. An entry of 99 in this field is a dummy interrupt level entry to provide FORTRAN linkages for the 1816 keyboards, the second magnetic tape unit, or mapped 1810 drives.
3	Blank	Must be blank
4-5	01 through 16	The entry in this field specifies the number of Interrupt Level Status Word (ILSW) bits that are assigned to this interrupt level.
6	Blank	Must be blank
7-72	A,B/X,C,..	A,B/X,C,.. represent a group of IAC and IAC/LUN assignments. See Table 1 for possible values. The IAC and IAC/LUN assignments must be separated by commas. An element of the group must be either an IAC only or an IAC/LUN combination. An IAC/LUN combination must consist of two numbers separated by a slash. Some IACs have no LUN assignable to them (see Table 2). If a LUN is assignable to an IAC, but no LUN is assigned, the System Loader equates LUN to the IAC by default. A particular LUN cannot be assigned to more than one device. If column 72 contains a nonblank character, a continuation card must be included.

This page intentionally left blank.

Appendix B. Decimal and Hexadecimal Disk Addresses

SECTOR ADDRESS BASE 10	SECTOR ADDRESS BASE 16	CYLINDER ADDRESS BASE 10	CYLINDER ADDRESS BASE 16	SECTOR ADDRESS BASE 10	SECTOR ADDRESS BASE 16	CYLINDER ADDRESS BASE 10	CYLINDER ADDRESS BASE 16
+00000	0000	+00000	0000	+00800	0320	+00100	0064
+00008	0008	+00001	0001	+00808	0328	+00101	0065
+00016	0010	+00002	0002	+00816	0330	+00102	0066
+00024	0018	+00003	0003	+00824	0338	+00103	0067
+00032	0020	+00004	0004	+00832	0340	+00104	0068
+00040	0028	+00005	0005	+00840	0348	+00105	0069
+00048	0030	+00006	0006	+00848	0350	+00106	006A
+00056	0038	+00007	0007	+00856	0358	+00107	006B
+00064	0040	+00008	0008	+00864	0360	+00108	006C
+00072	0048	+00009	0009	+00872	0368	+00109	006D
+00080	0050	+00010	000A	+00880	0370	+00110	006E
+00088	0058	+00011	000B	+00888	0378	+00111	006F
+00096	0060	+00012	000C	+00896	0380	+00112	0070
+00104	0068	+00013	000D	+00904	0388	+00113	0071
+00112	0070	+00014	000E	+00912	0390	+00114	0072
+00120	0078	+00015	000F	+00920	0398	+00115	0073
+00128	0080	+00016	0010	+00928	03A0	+00116	0074
+00136	0088	+00017	0011	+00936	03A8	+00117	0075
+00144	0090	+00018	0012	+00944	03B0	+00118	0076
+00152	0098	+00019	0013	+00952	03B8	+00119	0077
+00160	00A0	+00020	0014	+00960	03C0	+00120	0078
+00168	00A8	+00021	0015	+00968	03C8	+00121	0079
+00176	00B0	+00022	0016	+00976	03D0	+00122	007A
+00184	00B8	+00023	0017	+00984	03D8	+00123	007B
+00192	00C0	+00024	0018	+00992	03E0	+00124	007C
+00200	00C8	+00025	0019	+01000	03E8	+00125	007D
+00208	00D0	+00026	001A	+01008	03F0	+00126	007E
+00216	00D8	+00027	001B	+01016	03F8	+00127	007F
+00224	00E0	+00028	001C	+01024	0400	+00128	0080
+00232	00E8	+00029	001D	+01032	0408	+00129	0081
+00240	00F0	+00030	001E	+01040	0410	+00130	0082
+00248	00F8	+00031	001F	+01048	0418	+00131	0083
+00256	0100	+00032	0020	+01056	0420	+00132	0084
+00264	0108	+00033	0021	+01064	0428	+00133	0085
+00272	0110	+00034	0022	+01072	0430	+00134	0086
+00280	0118	+00035	0023	+01080	0438	+00135	0087
+00288	0120	+00036	0024	+01088	0440	+00136	0088
+00296	0128	+00037	0025	+01096	0448	+00137	0089
+00304	0130	+00038	0026	+01104	0450	+00138	008A
+00312	0138	+00039	0027	+01112	0458	+00139	008B
+00320	0140	+00040	0028	+01120	0460	+00140	008C
+00328	0148	+00041	0029	+01128	0468	+00141	008D
+00336	0150	+00042	002A	+01136	0470	+00142	008E
+00344	0158	+00043	002B	+01144	0478	+00143	008F
+00352	0160	+00044	002C	+01152	0480	+00144	0090
+00360	0168	+00045	002D	+01160	0488	+00145	0091
+00368	0170	+00046	002E	+01168	0490	+00146	0092
+00376	0178	+00047	002F	+01176	0498	+00147	0093
+00384	0180	+00048	0030	+01184	04A0	+00148	0094
+00392	0188	+00049	0031	+01192	04A8	+00149	0095
+00400	0190	+00050	0032	+01200	04B0	+00150	0096
+00408	0198	+00051	0033	+01208	04B8	+00151	0097
+00416	01A0	+00052	0034	+01216	04C0	+00152	0098
+00424	01A8	+00053	0035	+01224	04C8	+00153	0099
+00432	01B0	+00054	0036	+01232	04D0	+00154	009A
+00440	01B8	+00055	0037	+01240	04D8	+00155	009B
+00448	01C0	+00056	0038	+01248	04E0	+00156	009C
+00456	01C8	+00057	0039	+01256	04E8	+00157	009D
+00464	01D0	+00058	003A	+01264	04F0	+00158	009E
+00472	01D8	+00059	003B	+01272	04F8	+00159	009F
+00480	01E0	+00060	003C	+01280	0500	+00160	00A0
+00488	01E8	+00061	003D	+01288	0508	+00161	00A1
+00496	01F0	+00062	003E	+01296	0510	+00162	00A2
+00504	01F8	+00063	003F	+01304	0518	+00163	00A3
+00512	0200	+00064	0040	+01312	0520	+00164	00A4
+00520	0208	+00065	0041	+01320	0528	+00165	00A5
+00528	0210	+00066	0042	+01328	0530	+00166	00A6
+00536	0218	+00067	0043	+01336	0538	+00167	00A7
+00544	0220	+00068	0044	+01344	0540	+00168	00A8
+00552	0228	+00069	0045	+01352	0548	+00169	00A9
+00560	0230	+00070	0046	+01360	0550	+00170	00AA
+00568	0238	+00071	0047	+01368	0558	+00171	00AB
+00576	0240	+00072	0048	+01376	0560	+00172	00AC
+00584	0248	+00073	0049	+01384	0568	+00173	00AD
+00592	0250	+00074	004A	+01392	0570	+00174	00AE
+00600	0258	+00075	004B	+01400	0578	+00175	00AF
+00608	0260	+00076	004C	+01408	0580	+00176	00B0
+00616	0268	+00077	004D	+01416	0588	+00177	00B1
+00624	0270	+00078	004E	+01424	0590	+00178	00B2
+00632	0278	+00079	004F	+01432	0598	+00179	00B3
+00640	0280	+00080	0050	+01440	05A0	+00180	00B4
+00648	0288	+00081	0051	+01448	05A8	+00181	00B5
+00656	0290	+00082	0052	+01456	05B0	+00182	00B6
+00664	0298	+00083	0053	+01464	05B8	+00183	00B7
+00672	02A0	+00084	0054	+01472	05C0	+00184	00B8
+00680	02A8	+00085	0055	+01480	05C8	+00185	00B9
+00688	02B0	+00086	0056	+01488	05D0	+00186	00BA
+00696	02B8	+00087	0057	+01496	05D8	+00187	00BB
+00704	02C0	+00088	0058	+01504	05E0	+00188	00BC
+00712	02C8	+00089	0059	+01512	05E8	+00189	00BD
+00720	02D0	+00090	005A	+01520	05F0	+00190	00BE
+00728	02D8	+00091	005B	+01528	05F8	+00191	00BF
+00736	02E0	+00092	005C	+01536	0600	+00192	00C0
+00744	02E8	+00093	005D	+01544	0608	+00193	00C1
+00752	02F0	+00094	005E	+01552	0610	+00194	00C2
+00760	02F8	+00095	005F	+01560	0618	+00195	00C3
+00768	0300	+00096	0060	+01568	0620	+00196	00C4
+00776	0308	+00097	0061	+01576	0628	+00197	00C5
+00784	0310	+00098	0062	+01584	0630	+00198	00C6
+00792	0318	+00099	0063	+01592	0638	+00199	00C7

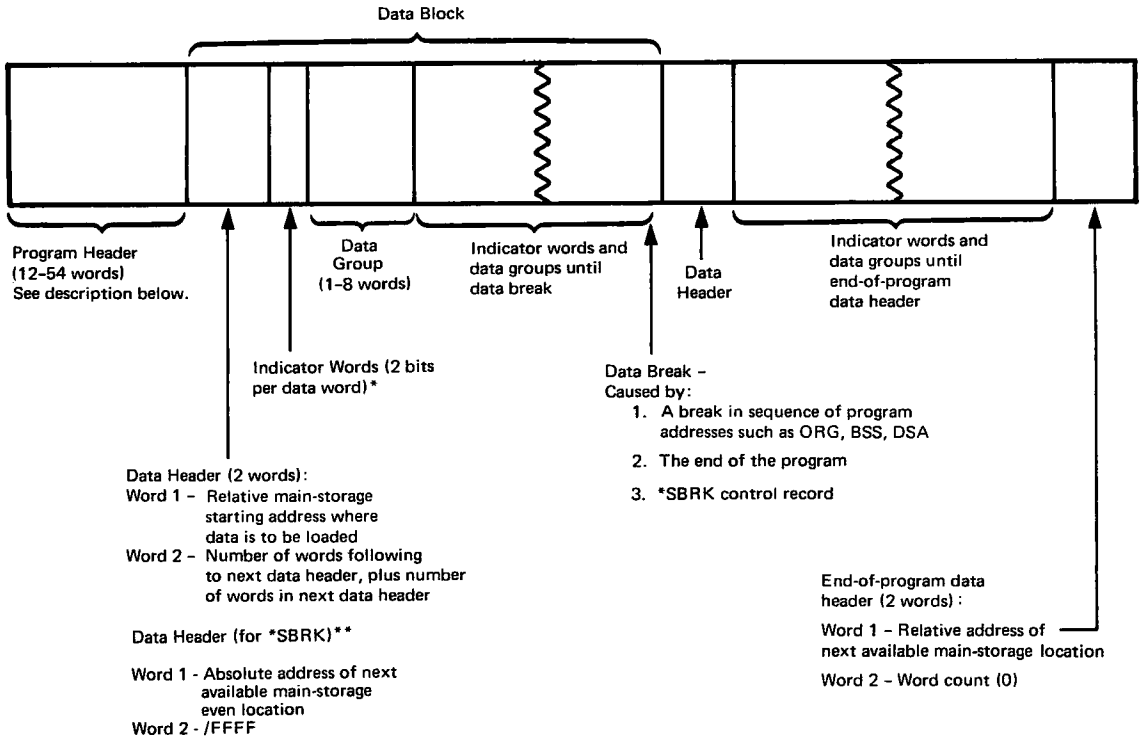
APDX B

This page intentionally left blank.

Appendix C. Data Formats

Disk System Format (DSF)

Unless otherwise instructed, the Disk Management Program (DMP) automatically converts programs in Card System Format (CDS) to Disk System Format (DSF), when programs are stored on disk. Likewise, programs in DSF are converted to CDS when dumping from to the 1442 card read punch. Disk System Format is shown in Figure 18; Card System Format is described elsewhere in this appendix.



*The indicator bits generated by the MPX system FORTRAN Compiler and Macro Assembler must have the following configuration:

00	Absolute	Indicator bits from two consecutive data words required.
01	Relocatable	
1000	LIBF	
1100	CALL	
1101	DSA	
1110	INSKEL COMMON	
1111	COMMON in a coreload	

** The data group following this header contains only three words; the first word is an indicator word of all zeros and the next two words contain the name (from the *SBRK control statement) in name code. Following this three word data group is another data header.

Figure 18. Disk System Format

APDX C

PROGRAM HEADER FORMAT

The contents of the program header record (see Figure 18) vary with the type of subroutine with which it is associated. The first 12 words of the program header record for the six

types of programs are identified except for word 6, which is 9 less than the number of words in the program header record. The format of these 12 words is as follows:

Word	Contents
1	Zero
2	Checksum if source was cards, otherwise zero
3	Type
4	Word 4 of each program header will contain the length of INSKEL COMMON defined in the program (as supplied by the FORTRAN Compiler)
5	Length of coreload COMMON (words)
6	Length of program header record minus 9
7	Indicator bits for LIBFs in the FORTRAN I/O table
8	Length of program, including program header record, in disk blocks
9	Number of files defined
10-11	Name of entry point 1
12	Address of entry point 1 (absolute for type 1, relative to zero otherwise)

The indicator bits for word 7 are as follows:

Bit Position	Indication	
0	= 1	if 2311 subroutine DSOR or DSCR is required by FORTRAN I/O or assembler-language program I/O
1,2	= 00	if integer size unspecified
	= 01	if single word integers
	= 10	if multiple word integers
3,4	= 00	if precision unspecified
	= 01	if standard precision
	= 10	if extended precision
5	= 1	if TYPEN/EBPRT required by FORTRAN I/O
6	= 1	if CARDN/HOLEB required by FORTRAN I/O
7	= 1	if PRNTN/EBPRT required by FORTRAN I/O
8	= 1	if PAPTN/PAPEB required by FORTRAN I/O
9	= 1	if MAGT required by FORTRAN I/O
10	= 1	if TYPEN/HOLEX required by FORTRAN I/O
11	= 1	if PLOTX required by FORTRAN I/O
12	= 1	if MPX compiled/assembled
13	= 1	if reentrant program
14	= 1	if unformatted disk I/O
15	= 1	if unformatted magnetic tape I/O

The indicators in bits 1-4 and 12-13 are required in all header types. Bits 5-11 and 14-15 are required only in type 2 headers.

After the first 12 words, the program header record format depends on the type of program. The header record for types 1 and 2 (absolute and relocatable mainline, respectively) consists of the first 12 words. The program types and their header record formats are shown below:

Type Code	Type of Program
1	Mainline (absolute)
2	Mainline (relocatable)
3	Subroutine, not an ISS, referenced by LIBF
4	Subroutine, not an ISS, referenced by CALL
5	Interrupt service subroutine (ISS) referenced by LIBF
6	Interrupt service subroutine (ISS) referenced by CALL

Program formats for type 3 and 4 programs:

Words	Contents
13-14	Name of entry point 2 (30 bits, right-justified)
15	Address of entry point 2 (relative to zero)
16-17	Name of entry point 3 (30 bits, right-justified)
18	Address of entry point 3 (relative to zero)
19-54	Three words per entry point as above, to a maximum of 14 entry points. The header record ends at the last defined entry point; thus, it is of variable length.

Program formats for type 5 and 6 programs:

Words	Contents
13	Contains zero
14	Contains the number of interrupt service entry points in the ISS
15	Reserved

Disk Data Format (DDF)

Disk Data Format (DDF) describes the format of information placed in the Disk Fixed Area or in Batch-Processing Working Storage as a result of the DMP control statement *STOREDATA. Disk Data Format consists of 320 binary words per sector; there are no headers, trailers, or indicator words.

Card System Format (CDS)

Words in columns 1 through 72 are punched in binary form, 1 1/3 columns for each word. Each card may contain 54 words in binary form. Columns 73 through 80 may contain card identification or sequence numbers, in IBM card code. Data cards are punched with up to 45 words in columns 13 through 72. See Figure 19.

MAINLINE HEADER CARD

A mainline header card contains the size of the common area and the size of the Batch-processing Working Storage. It is the first card of the mainline program. Its format is as follows:

Words	Contents
1	Reserved
2	Checksum
3	Type code (first 8 bits): 0000 0001 - absolute 0000 0010 - relocatable
4	Word 4 of each mainline header card will contain the length of INSKEL COMMON defined in the program (as supplied by the FORTRAN Compiler)
5	Length of COMMON storage area (FORTRAN mainline program only)

Words	Contents
6	0000 0000 0000 0011
7	See Program Header Format, word 7
8	Disk block count
9	Number of files (FORTRAN mainline)
10-11	Mainline name
12	Execution address
13-54	Reserved

The checksum in word 2 is the twos complement of the logical sum of the record count (position of the record within the deck) and the data word(s). The logical sum is obtained by summing the data word(s) and the record count arithmetically with the addition of a one each time a carry occurs out of the high-order position of the A-register.

DATA CARDS

Data cards contain the instructions and data that constitute the assembled program. The format is as follows:

Words	Contents
1	Location. (The relative load address of the first data word of the card or record. Succeeding words go into higher-numbered main-storage locations. The relocation factor must be added to this address to obta obtain the actual load address. For an absolute program the relocation factor is zero.)
2	Checksum
3	Type code (first 8 bits): 0000 1010 Data word count (last 8 bits)
4-9	Relocation indicators (2 bits per data word): 00 - nonrelocatable or absolute 01 - relocatable 10 - LIBF (one-word call) 11 - CALL (two-word call)
10	Data word 1
11-54	Data words 2 through 45

EOP CARD

An end of program card (EOP) is the last card of each program and subroutine. The format is as follows:

Word	Contents
1	Starting location of next subroutine (this number is always even and is assigned by the Macro Assembler)
2	Checksum
3	Type code (first 8 bits): 0000 1111 Last 8 bits: 0000 0000
4	XEQ address, if mainline program
5-54	Reserved

SUBROUTINE HEADER CARD

A maximum of 14 entry points can be defined for each subroutine. The format of the subroutine header card is as follows:

Word	Contents
1	Reserved
2	Checksum
3	Type code (first 8 bits): 0000 0011 - to be called by a one-word call only (LIBF) 0000 0100 - to be called by a two-word call only (CALL)
4-5	Reserved
6	Number of entry points times three
7	See Program Header Format, word 7
8	Program length
9	Reserved
10-11	Name of entry point 1
12	Relative address of entry point 1
13-51	Names and relative addresses of entry points 2 through 14
52-54	Reserved

ISS HEADER CARD

An ISS (interrupt servicing subroutine) header card for each interrupt servicing subroutine identifies the entry point defined by an ISS statement. Only one entry point can be defined for each subroutine. The format of the ISS header card is as follows:

Word	Contents
1	Reserved
2	Checksum
3	Type code (first 8 bits): 0000 0101 - to be called by a one-word call only (LIBF) 0000 0110 - to be called by a two-word call only (CALL) Precision code (last 8 bits): 0000 0000 - undefined 0000 0001 - standard 0000 0010 - extended
4-5	Reserved
6	Six
7	See Program Header Format, word 7
8-9	Reserved
10-11	Subroutine name
12	Relative entry address
13	Zero
14	The number of interrupt service entry points in the ISS
15-54	Reserved

SECTOR BREAK CARDS

Sector break cards are binary cards used by the System Loader and DMON function of DMP to cause programs or phases of programs to start loading at the beginning of a sector. Sequence numbers are punched in columns 73-80 in IBM Card Code. The sector break cards are identified by a 1 punch in column 4 (binary word 3).

A Type 1 sector break card indicates to the System Loader or DMON function of DMP that the following program or phase should start loading at the next new sector address available.

The format of the sector break card is as follows:

Word	Contents
1	Reserved
2	Checksum
3	Type code (first 8 bits): 0000 0001
4-9	Reserved
10-11	Phase name
12-54	Reserved

Card Data Format (CDD)

All words are punched in binary form, 1 1/3 columns for each word. Each card may contain 60 words. Data cards are punched with up to 56 words in columns 5 through 79. See Figure 19.

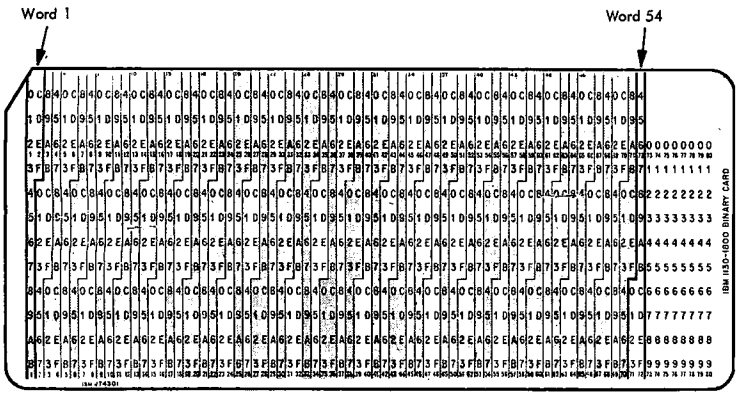


Figure 19. Card Data Format

DISK DUMP FOR PHYSICAL OR MAPPED 1810s

The output from the DUMP function of DMP or the Disk-Dump-to-Cards Utility Program (BDUMP) is in Card Data Format. The 1810 dumped disk data is punched into cards as follows:

CYLINDER BREAK CARD FORMAT

Word	Contents
1	/FFFF
2	Checksum
3	Reserved
4	Sector address
5	File-protection status /8xxx = on
6-59	Reserved
60	Sequence number

DATA CARD FORMAT

Word	Contents
1	Relative address of data on cylinder
2	Checksum
3	Word count (number of data
4-59	words)
60	Sequence number

END OF ALL CYLINDERS FORMAT

Word	Contents
1	Reserved
2	Checksum
3	/FFFF
4-59	Reserved
60	Sequence number

DISK DUMP FOR 2311s

The output from the 2311 Disk Dump and Reload Program (BDCRL) is in Card Data Format. A card deck from a 2311 dump can be identified by the /FFFF in word 5 of the first cylinder break card. Each new record begins on a new card. The first word of these new record cards contains zero. The format of the other cards is as follows:

FIRST CYLINDER BREAK CARD

Word	Contents
1	/FFFF
2	Checksum
3	Reserved
4	Cylinder address
5	/FFFF
6-59	Reserved
60	Sequence number

SUBSEQUENT CYLINDER BREAK CARDS

Word	Contents
1	/FFFF
2	Checksum
3	Reserved
4	Cylinder address
5-59	Reserved
60	Sequence number

CYLINDER DATA CARD

Word	Contents
1	Data word address, relative to the beginning of the record
2	Checksum
3	Word count (number of data words)
4-59	Data words
60	Sequence number

END-OF-ALL-CYLINDER CARDS

Word	Contents
1	Reserved
2	Checksum
3	/FFFF
4-59	Reserved
60	Sequence number

MAIN STORAGE DUMP

The output of the Dump-to-Cards utility program is in binary format, 3/4 of a 16-bit binary word per card column. This format allows 56 data words to be punched in each 80-column card, where 1-1/3 columns equal one binary word. The main storage words are output to cards as follows:

FIRST CARD FORMAT

Word	Contents
1	/FFFF
3	Starting address of dump program
4	Ending address of dump program

DATA CARD FORMAT

Word	Contents
1	Relative address of data in main storage
2	Checksum
3	Word count (number of data words)
4-59	Data words
60	Sequence number

LAST CARD FORMAT

Word	Contents
1	Zero
2	Checksum
3	/FFFF
4	Execution Address
5-59	Not used
60	Sequence number

Print Data Format (PRD)

Print Data Format is the format in which programs and data files are dumped to the print devices. The dump is printed by sector. The dump of each sector consists of a heading followed by the data on the sector dumped.

The heading for each sector dumped consists of the absolute sector address of the sector being dumped and the location of the first data word in the sector dumped relative to the start of the program or data file being dumped.

The body of the dump of a sector consists of from one to twenty lines, each line containing the location of the first data word in the line printed relative to the start of the sector being dumped, followed by 16 hexadecimal data words. Two spaces separate each data word from those adjacent to it.

The printing of lines that contain exactly the same data as the previous line printed is suppressed. One line of blanks is printed to indicate that lines have been suppressed. An exception to the procedure may occur in the printing of the last line of the dump of a sector; the last line is always printed.

Appendix D. MPX Sample Program

The MPX sample program contains five background-processing coreloads written in FORTRAN. The sample program deck is punched during system generation and is run as a background-processing job. System assignments must be made before the sample program can be run.

Operating Procedure

Enter the logical unit number of the input and output device in the source cards as follows. (SAMxxxxx is the source card ID.)

```
I=     SAM00220
```

Set I equal to the LUN of the device which will be used to input user data during the program run. Keyboard input is recommended.

```
J=     SAM00250
```

Set J equal to the LUN of the output device used for listing purposes. List printer is recommended.

Note: If there is no 1443 on the system, it is advisable to remove the *LIST ALL cards from the source deck.

Run the sample program as a batch-processing job.

The source deck is compiled and listed on the list printer. At the same time, operating instructions are printed on the system printer. The system and list printer printouts (these can be the same printer) are listed below. The background-processing coreloads are then built, executed, and finally deleted.

Additional operating procedures are shown on the system printer printout.

SYSTEM PRINTER OUTPUT

```
 IBM 1800 DACS   MPX/BOM   00.000 00 JAN 00
 SEN SW 0 ON ABSOLUTE LOADER
 SEN SW 1 ON LOAD BP MONITOR
 SEN SW 2 ON SET CLOCK VIA DATA SWS
 SEN SW 3 ON SET DATE VIA DATA SWS
// JOB   00 JAN 00 00.000 HRS                                          SAM00010
// *MPX SAMPLE PROGRAM                                                 SAM00020
// * SAMPLE PROBLEM READS AI POINTS AND OUTPUTS READING ON USER        SAM00030
// * SPECIFIED DEVICE. FORTRAN LOGICAL UNIT NUMBERS FOR INPUT          SAM00040
// * AND OUTPUT DEVICES ARE SPECIFIED IN THE FIRST PROGRAM (SAMPL).    SAM00050
// * THE USER MUST PUNCH IN THE VALUES FOR THESE DEVICES.              SAM00060
// * RELAY INPUT POINTS ARE 00000 - 04005.                            SAM00070
// * SS POINTS ARE 04096 - 08101. AFTER POINTS ARE READ AND            SAM00080
// * VALUES OUTPUT,A VALUE IS ENTERED VIA A USER SPECIFIED             SAM00090
// * INPUT DEVICE, FORMAT 15. THE SQUARE ROOT OF THIS VALUE           SAM00100
// * IS CALCULATED AND OUTPUT ON A USER SPECIFIED DEVICE.              SAM00110
// JOB   00 JAN 00 00.026 HRS                                          SAM00120
// FOR SAMPL   00 JAN 00 00.028 HRS                                    SAM00130
END OF COMPILATION

SAMPL
DMP FUNCTION COMPLETED
// DMP                                                                SAM00280
*STORECIL          SAMPL SAMPL                                       SAM00290
*CCEND                                                               SAM00300
MPX, BUILD SAMPL

MPX, SAMPL LD XQ

045 CORELDS NOT END
SAMPA
CL WC OF  0048 STORED AT  04C0
DMP FUNCTION COMPLETED
// FOR SAMPZ   00 JAN 00 00.052 HRS                                    SAM00310
END OF COMPILATION

SAMPZ
DMP FUNCTION COMPLETED
// DMP                                                                SAM00440
*STORECIL          SAMPZ SAMPZ                                       SAM00450
*CCEND                                                               SAM00460
MPX, BUILD SAMPZ

MPX, SAMPZ LD XQ

CL WC OF  0DC8 STORED AT  04C1
DMP FUNCTION COMPLETED
// FOR SAMPY   00 JAN 00 00.088 HRS                                    SAM00470
END OF COMPILATION

SAMPY
DMP FUNCTION COMPLETED
// DMP                                                                SAM00610
*STORECIL          SAMPY SAMPY                                       SAM00620
*CCEND                                                               SAM00630
MPX, BUILD SAMPY

MPX, SAMPY LD XQ

CL WC OF  0E7E STORED AT  04CD
DMP FUNCTION COMPLETED
// FOR SAMPC   00 JAN 00 00.124 HRS                                    SAM00640
END OF COMPILATION

SAMPC
DMP FUNCTION COMPLETED
// DMP                                                                SAM00790
*STORECIL          SAMPC SAMP                                        SAM00800
*CCEND                                                               SAM00810
MPX, BUILD SAMPC

MPX, SAMPC LD XQ

045 CORELDS NOT FND
SAMPA
CL WC OF  0BEA STORED AT  04D9
DMP FUNCTION COMPLETED
// FOR SAMPB   00 JAN 00 00.160 HRS                                    SAM00820
END OF COMPILATION


SAMPB
DMP FUNCTION COMPLETED
// DMP                                                                SAM00990
*STORECIL          SAMPB SAMPB                                       SAM01000
*CCEND                                                               SAM01010
MPX, BUILD SAMPB

MPX, SAMPB LD XQ

CL WC OF  04C4 STORED AT  04E3
DMP FUNCTION COMPLETED
// FOR SAMPA   00 JAN 00 00.189 HRS                                    SAM01020
END OF COMPILATION

SAMPA
DMP FUNCTION COMPLETED
// DMP                                                                SAM01150
*STORECIL          SAMPA SAMPA                                       SAM01160
*CCEND                                                               SAM01170
MPX, BUILD SAMPA
```

```
MPX, SAMPA LD XQ

CL WC OF  03E2 STORED AT  04E7
DMP FUNCTION COMPLETED
// XEQ SAMPL   FX0   00 JAN 00 00.225 HRS                                SAM01180
00001
00005
// JOB   00 JAN 00 00.374 HRS                                           SAM01190
// DMP   00 JAN 00 00.375 HRS                                           SAM01200
*DELET              SAMPC                                             SAM01210
DMP FUNCTION COMPLETED
*DELET              SAMPZ                                             SAM01220
DMP FUNCTION COMPLETED
*DELET              SAMPY                                             SAM01230
DMP FUNCTION COMPLETED
*DELET              SAMPA                                             SAM01240
DMP FUNCTION COMPLETED
*DELET              SAMPB                                             SAM01250
DMP FUNCTION COMPLETED
*DELET              SAMPL                                             SAM01260
DMP FUNCTION COMPLETED
// JOB   00 JAN 00 00.402 HRS                                           SAM01270
// DMP   00 JAN 00 00.403 HRS                                           SAM01280
*DUMPLET                                                              SAM01290
DMP FUNCTION COMPLETED
// END                 END OF SAMPLE PROGRAM. 00 JAN 00 00.417 HRS 99999999
 IBM 1800 DACS  MPX/BOM   00.419  00 JAN 00
SEN SW 0 ON ABSOLUTE LOADER
SEN SW 1 ON LOAD BP MONITOR
SEN SW 2 ON SET CLOCK VIA DATA SWS
SEN SW 3 ON SET DATE VIA DATA SWS

// JOB   00 JAN 00 00.006 HRS                                           SAM00010
// *MPX SAMPLE PROGRAM                                                  SAM00020
// * SAMPLE PROBLEM READS AI POINT AND OUTPUTS READING ON USER          SAM00030
// * SPECIFIED DEVICE. FORTRAN LOGICAL UNIT NUMBERS FOR INPUT           SAM00040
// * AND OUTPUT DEVICES ARE SPECIFIED IN THE FIRST PROGRAM (SAMPL).     SAM00050
// * THE USER MUST PUNCH IN THE VALUES FOR THESE DEVICES.               SAM00060
// * RELAY INPUT POINTS ARE 00000 - 04095.                              SAM00070
// * SS POINTS ARE 04096 - 08191. AFTER POINTS ARE READ AND             SAM00080
// * VALUES OUTPUT,A VALUE IS ENTERED VIA A USER SPECIFIED              SAM00090
// * INPUT DEVICE, FORMAT 15. THE SQUARE ROOT OF THIS VALUE             SAM00100
// * IS CALCULATED AND OUTPUT ON A USER SPECIFIED DEVICE.               SAM00110


// JOB   00 JAN 00 00.026 HRS                                           SAM00120
// FOR SAMPL   00 JAN 00 00.028 HRS                                     SAM00130
** MPX SAMPLE PROGRAM                                                  SAM00140
*LIST ALL                                                             SAM00150
*ONE WORD INTEGERS                                                    SAM00160
*NONPROCESS PROGRAM                                                   SAM00170


      COMMON I,J,L,K,A                                                SAM00180
C     START OF PROGRAM                                                SAM00190
C                                                                     SAM00200
C DEFINE LOGICAL UNIT NUMBER OF INPUT DEVICE.                         SAM00210
      I=06                                                            SAM00220
C                                                                     SAM00230
C DEFINE LOGICAL UNIT NUMBER OF OUTPUT DEVICE.                        SAM00240
      J=3                                                             SAM00250
      CALL LINK(SAMPA)                                                SAM00260
      END                                                             SAM00270


VARIABLE ALLOCATIONS
     I(IC)=FFFF          J(IC)=FFFE          L(IC)=FFFD          K(IC)=FFFC          A(RC)=FFFA

FEATURES SUPPORTED
 NONPROCESS
 ONE WORD INTEGERS

INTEGER CONSTANTS
     6=0000     3=0001

CORE REQUIREMENTS FOR SAMPL
 COMMON      6  INSKEL COMMON      0
  VARIABLES      0  PROGRAM     14


END OF COMPILATION

SAMPL
DMP FUNCTION COMPLETED
// DMP                                                                SAM00280
*STORECIL           SAMPL SAMPL                                       SAM00290

*CCEND                                                                SAM00300
```

```
MPX, BUILD SAMPL

CORE LOAD  MAP
TYPE NAME  ARG1  ARG2

*CDW TABLE 8002  0012
*FIO TABLE 8014  001E
*CNT TABLE 8032  0008
MAIN SAMPL 803C
CLNT SAMPL 8034
CLNT SAMPA 8038
CORE       804A  7FB0
COMM       FFFA  0006
MPX, SAMPL LD XQ

045 CORELDS NOT END
SAMPA
CL WC OF  0048 STORED AT  04C0
DMP FUNCTION COMPLETED
// FOR SAMPZ    00 JAN 00 00.052 HRS                                    SAM00310
** MPX SAMPLE PROGRAM                                                   SAM00320
*LIST ALL                                                              SAM00330
*ONE WORD INTEGERS                                                     SAM00340
*NONPROCESS PROGRAM                                                    SAM00350
*IOCS(TYPEWRITER,1443 PRINTER)                                         SAM00360

      COMMON I,J,L,K,A                                                 SAM00370
      START OF PROGRAM                                                 SAM00380
      A=SQRT(A)                                                        SAM00390
      WRITE (J,8)K,A                                                   SAM00400
    8 FORMAT(' SQUARE ROOT OF',I6,' IS',E20.9)                         SAM00410
      CALL EXIT                                                        SAM00420
      END                                                              SAM00430


VARIABLE ALLOCATIONS
     I(IC)=FFFF          J(IC)=FFFE          L(IC)=FFFD          K(IC)=FFFC          A(RC)=FFFA

STATEMENT ALLOCATIONS
 8    =0000

FEATURES SUPPORTED
 NONPROCESS
 ONE WORD INTEGERS
 IOCS

CALLED SUBPROGRAMS
 FSQRT   FSTO    MWRT    MCOMP   MIOF    MIOI    TYPEN   PRNTN   EBPRT

CORE REQUIREMENTS FOR SAMPZ
 COMMON      6  INSKEL COMMON      0
  VARIABLES      0  PROGRAM     30


END OF COMPILATION


SAMPZ
DMP FUNCTION COMPLETED
// DMP                                                                 SAM00440
*STORECIL           SAMPZ SAMPZ                                        SAM00450

*CCEND                                                                 SAM00460

MPX, BUILD SAMPZ

CORE LOAD  MAP
TYPE NAME  ARG1  ARG2

*CDW TABLE 8002  0012
*FIO TABLE 8014  001E
*VTV TABLE 8032  002A
*CNT TABLE 805C  0004
MAIN SAMPZ 806F
CLNT SAMPZ 805E
CALL EBPRT 807E
CALL FSQRT 8128
LIBF FSTO  81C1  8032
LIBF MWRT  83BC  8035  R
LIBF MIOI  8424  8038
LIBF MIOF  8420  803B
LIBF MCOMP 8407  803E
CALL PRT   88E0
LIBF FMPY  8931  8041
LIBF FADD  89A4  8044
LIBF FDIVX 8A2A  8047
LIBF FADDX 89A8  804A
LIBF ADRCK 8AA6  804D  R
CALL FEROR 8B59        R
CALL SBCAL 8C87
CALL IOU   8CCA        R
CALL IOFIX 8CE4        R
CALL GETBF 8B97
CALL FREBF 8B89
CALL FIOXT 8B8D
LIBF FLOAT 8D14  8050
LIBF IFIX  8D28  8053
LIBF FARC  8D5E  8056
CALL FREUP 8C66        R
LIBF NORM  8D90  8059
FIOB       8DCA  0054
CORE       8E1E  710C
COMM       FFFA  0006
MPX, SAMPZ LD XQ
```

```
CL WC OF  0DC8 STORED AT  04C1
DMP FUNCTION COMPLETED
// FOR SAMPY    00 JAN 00 00.088 HRS                                    SAM00470
** MPX SAMPLE PROGRAM                                                   SAM00480
*LIST ALL                                                              SAM00490
*ONE WORD INTEGERS                                                     SAM00500
*NONPROCESS PROGRAM                                                    SAM00510
*IOCS(KEYBOARD,TYPEWRITER,1443 PRINTER,CARD)                           SAM00520

      COMMON I,J,L,K,A                                                 SAM00530
    6 WRITE(J,7)                                                       SAM00540
    7 FORMAT (' ENTER NUMBER--FORMAT I5')                              SAM00550
      READ (I,2)K                                                      SAM00560
    2 FORMAT(I5)                                                       SAM00570
      A=K                                                              SAM00580
      CALL LINK (SAMPZ)                                                SAM00590
      END                                                              SAM00600


VARIABLE ALLOCATIONS
     I(IC)=FFFF          J(IC)=FFFE          L(IC)=FFFD          K(IC)=FFFC          A(RC)=FFFA

UNREFERENCED STATEMENTS
 6

STATEMENT ALLOCATIONS
 7    =0000  2    =000E  6    =0010

FEATURES SUPPORTED
 NONPROCESS
 ONE WORD INTEGERS
 IOCS

CALLED SUBPROGRAMS
 FSTO    FLOAT   MRED    MWRT    MCOMP   MIOI    TYPEN   HOLEB   PRNTN   EBPRT   CARDN

CORE REQUIREMENTS FOR SAMPY
 COMMON      6  INSKEL COMMON      0
  VARIABLES      0  PROGRAM     36


SAMPY
DMP FUNCTION COMPLETED
// DMP                                                                 SAM00610
*STORECIL           SAMPY SAMPY                                        SAM00620

*CCEND                                                                 SAM00630



 MPX, BUILD SAMPY

 CORE LOAD  MAP
 TYPE NAME  ARG1  ARG2

 *CDW TABLE 8002  0012
 *FIO TABLE 8014  001E
 *VTV TABLE 8032  0018
 *CNT TABLE 804E  0008
 MAIN SAMPY 8066
 CLNT SAMPY 8050
 CALL EBPRT 807A
 CALL CARDN 8124        R
 CALL HOLEB 82AE
 CALL HOLEX 83D9        R
 LIBF MWRT  85D6  8032  R
 LIBF MCOMP 8621  8035
 LIBF MRED  85D2  8038
 LIBF MIOI  863E  803B
 LIBF FLOAT 8AFA  803E
 LIBF FSTO  8B31  8041
 CLNT SAMPZ 8054
 CALL PRT   8B58
 CALL FERDR 8BA5        R
 CALL SBCAL 8CD3
 CALL IOU   8D16        R
 CALL IOFIX 8D30        R
 CALL GETBF 8BE3
 CALL FREBF 8BD5
 CALL FIOXT 8BD9
 LIBF ADRCK 8D60  8044  R
 LIBF IFIX  8E10  8047
 LIBF NORM  8E46  804A
 CALL FREUP 8CB2        R
 FIOB       8E80  0054
 CORE       8ED4  7126
 COMM       FFFA  0006
 MPX, SAMPY LD XQ

CL WC OF  0E7E STORED AT  04CD
DMP FUNCTION COMPLETED
// FOR SAMPC    00 JAN 00 00.124 HRS                                    SAM00640
** MPX SAMPLE PROGRAM                                                   SAM00650
*LIST ALL                                                              SAM00660
*ONE WORD INTEGERS                                                     SAM00670
*NONPROCESS PROGRAM                                                    SAM00680
*IOCS(TYPEWRITER,1443 PRINTER)                                         SAM00690

      COMMON I,J,L,K,A                                                 SAM00700
    4 WRITE (J,5)L                                                     SAM00710
    5 FORMAT(I7,/,' DAT SW 0 ON TO READ ANOTHER POINT')                SAM00720
      PAUSE                                                            SAM00730
      CALL DATSW(0,ITEST)                                              SAM00740
      GO TO (10,6),ITEST                                               SAM00750
   10 CALL LINK (SAMPA)                                                SAM00760
    6 CALL LINK(SAMPY)                                                 SAM00770
      END                                                              SAM00780
```

```
VARIABLE ALLOCATIONS
     I(IC)=FFFF          J(IC)=FFFE          L(IC)=FFFD          K(IC)=FFFC          A(RC)=FFFA      ITEST(I )=0000

UNREFERENCED STATEMENTS
 4

STATEMENT ALLOCATIONS
 5    =0004  4    =0019  10   =002B  6    =002F

FEATURES SUPPORTED
 NONPROCESS
 ONE WORD INTEGERS
 IOCS

CALLED SUBPROGRAMS
 DATSW   COMGO   MWRT    MCOMP   MIOI    PAUSE   TYPEN   PRNTN   EBPRT

INTEGER CONSTANTS
     0=0002      0=0003

CORE REQUIREMENTS FOR SAMPC
 COMMON      6  INSKEL COMMON      0
  VARIABLES      2  PROGRAM     50



END OF COMPILATION

SAMPC
DMP FUNCTION COMPLETED
// DMP                                                                    SAM00790
*STORECIL            SAMPC SAMPC                                         SAM00800

*CCEND                                                                   SAM00810

 MPX, BUILD SAMPC

 CORE LOAD  MAP
 TYPE NAME  ARG1  ARG2

 *COM TABLE 8002  0012
 *FID TABLE 8014  001E
 *VTV TABLE 8032  001B
 *CNT TABLE 804E  000C
 MAIN SAMPC 8073
 CLNT SAMPC 8050
 CALL EBPRT 808E
 LIBF MWRT  830C  8032  R
 LIBF MIOI  8374  8035
 LIBF MCOMP 8357  8038
 LIBF PAUSE 8830  803B  R
 CALL DATSW 8844        R
 LIBF COMGO 8879  803E  R
 CLNT SAMPA 8054
 CLNT SAMPY 8058
 CALL PRT   88B0
 CALL FEROR 88FD        R
 CALL SBCAL 8A2B
 CALL IOU   8A6E        R
 CALL IOFIX 8A88        R
 CALL GETBF 893B
 CALL FREBF 892D
 CALL FIOXT 8931
 LIBF ADRCK 8A88  8041  R
 LIBF FLOAT 8B68  8044
 LIBF IFIX  8B7C  8047
 CALL FREUP 8A0A        R
 LIBF NORM  8BB2  804A
 FIOB       8BEC  0054
 CORE       8C40  73BA
 COMM       FFFA  0006
 MPX, SAMPC LD XQ

045 CORELDS NOT FND
SAMPA
CL WC OF  OBEA STORED AT  04D9
DMP FUNCTION COMPLETED
// FOR SAMPB   00 JAN 00 00.160 HRS                                      SAM00820
** MPX SAMPLE PROGRAM                                                    SAM00830
*LIST ALL                                                               SAM00840
*ONE WORD INTEGERS                                                      SAM00850
*NONPROCESS PROGRAM                                                     SAM00860

                DIMENSION LIST(10),IARA(3)                                      SAM00870
                COMMON I,J,L,K,A                                                SAM00880
                EQUIVALENCE (ITEST,LIST(4))                                     SAM00890
                DATA LIST,IARA(3)/2*0,Z1000,7*0,2/                              SAM00900
                LIST(1)=IADDR(IARA(3))                                          SAM00910
                IARA(2)=K                                                       SAM00920
          4     CALL AISN(LIST(10))                                             SAM00930
          1     IF (LIST(10))1,2,1                                              SAM00940
          2     L=IARA(1)                                                       SAM00950
                GO TO(3,4,4,4,4,4),ITEST                                        SAM00960
          3     CALL LINK(SAMPC)                                                SAM00970
                END                                                             SAM00980

VARIABLE ALLOCATIONS
     I(IC)=FFFF          J(IC)=FFFE          L(IC)=FFFD          K(IC)=FFFC          A(RC)=FFFA
  LIST(I )=0009-0000  IARA(I )=000C-000A

STATEMENT ALLOCATIONS
 4    =0024  1    =002E  2    =0034  3    =0044
```

```
FEATURES SUPPORTED
 NONPROCESS
 ONE WORD INTEGERS

CALLED SUBPROGRAMS
 IADDR   AISN    COMGO   ISTOX

CORE REQUIREMENTS FOR SAMPB
 COMMON      6  INSKEL COMMON      0
  VARIABLES     16  PROGRAM     56



END OF COMPILATION

           SAMPB
           DMP FUNCTION COMPLETED
           // DMP                                                    SAM00990
           *STORECIL           SAMPB SAMPB                          SAM01000

           *CCEND                                                   SAM01010


            MPX, BUILD SAMPB

            CORE LOAD  MAP
            TYPE NAME  ARG1  ARG2

            *COM TABLE 8002  0012
            *FIO TABLE 8014  001E
            *VTV TABLE 8032  0009
            *CNT TABLE 803C  0008
            MAIN SAMPB 8054
            CLNT SAMPB 803E
            CALL IADDR 808C        R
            LIBF ISTOX 8092  8032  R
            CALL AISN  80B2        R
            LIBF COMGO 8267  8035  R
            CLNT SAMPC 8042
            LIBF ADRCK 829E  8038  R
            CALL ABRTQ 834E        R
            CALL FEROR 837F        R
            CALL FREUP 848C        R
            CORE       84C6  7834
            COMM       FFFA  0006
            MPX, SAMPB LO XQ

           CL WC OF  04C4 STORED AT  04E3
           DMP FUNCTION COMPLETED
           // FOR SAMPA    00 JAN 00 00.189 HRS                    SAM01020
           ** MPX SAMPLE PROGRAM                                    SAM01030
           *LIST ALL                                                SAM01040
           *ONE WORD INTEGERS                                       SAM01050
           *NONPROCESS PROGRAM                                      SAM01060
           *IOCS(KEYBOARD,TYPEWRITER,1443 PRINTER,CARD)             SAM01070

      COMMON I,J,L,K,A                                              SAM01080
   10 WRITE (J,1)                                                   SAM01090
    1 FORMAT(' ENTER AI POINT TO BE READ-- FORMAT 15')              SAM01100
      READ(I,2)K                                                    SAM01110
    2 FORMAT(I5)                                                    SAM01120
      CALL LINK(SAMPB)                                              SAM01130
      END                                                           SAM01140


VARIABLE ALLOCATIONS
     I(IC)=FFFF          J(IC)=FFFE          L(IC)=FFFD          K(IC)=FFFC          A(RC)=FFFA

UNREFERENCED STATEMENTS
 10

STATEMENT ALLOCATIONS
 1    =0000  2    =0015  10    =0017

FEATURES SUPPORTED
 NONPROCESS
 ONE WORD INTEGERS
 IOCS

CALLED SUBPROGRAMS
 MRED    MWRT    MCOMP    MIOI    TYPEN   HOLEB   PRNTN   EBPRT   CARDN

CORE REQUIREMENTS FOR SAMPA
 COMMON      6  INSKEL COMMON      0
  VARIABLES      0  PROGRAM     38


END OF COMPILATION

 SAMPA
 DMP FUNCTION COMPLETED
 // DMP                                                           SAM01150
 *STORECIL           SAMPA SAMPA                                  SAM01160

 *CCEND                                                          SAM01170
```

```
 MPX, BUILD SAMPA

 CORE LOAD  MAP
 TYPE NAME  ARG1  ARG2

 *CDW TABLE 8002  0012
 *FIO TABLE 8014  001E
 *VTV TABLE 8032  0018
 *CNT TABLE 804A  0008
 MAIN SAMPA 8069
 CLNT SAMPA 804C
 CALL EBPRT 8078
 CALL CARDN 8122
 CALL HOLER 82AC
 CALL HOLEX 83D7
 LIBF MWRT  85D4  8032
 LIBF MCOMP 861F  8035
 LIBF MRED  85D0  8038
 LIBF MIOI  863C  803B
 CLNT SAMBB 8050
 CALL PRT   8AF8
 CALL FEROR 8B45
 CALL SBCAL 8C73
 CALL IOU   8CB6
 CALL IOFIX 8CD0
 CALL GETBF 8B83
 CALL FREBF 8B75
 CALL FIOXT 8B79
 LIBF ADRCK 8D00  803E
 LIBF FLOAT 8DB0  8041
 LIBF IFIX  8DC4  8044
 CALL FREUP 8C52
 LIBF NORM  8DFA  8047
 FIOB       8E34  0054
 CORE       8E88  7172
 COMM       FFFA  0006
 MPX, SAMPA LD XQ

CL WC OF  0E32 STORED AT  04E7
DMP FUNCTION COMPLETED
// XEQ SAMPL   FXD    00 JAN 00 00.568 HRS                          SAM01180

-10392
DAT SW 0 ON TO READ ANOTHER POINT
ENTER NUMBER--FORMAT I5
SQUARE ROOT OF     5 IS     0.223606859E 01


// JOB    00 JAN 00 00.374 HRS                                      SAM01190
// DMP    00 JAN 00 00.375 HRS                                      SAM01200
*DELET              SAMPC                                         SAM01210
DMP FUNCTION COMPLETED
*DELET              SAMPZ                                         SAM01220
DMP FUNCTION COMPLETED
*DELET              SAMPY                                         SAM01230
DMP FUNCTION COMPLETED
*DELET              SAMPA                                         SAM01240
DMP FUNCTION COMPLETED
*DELET              SAMPB                                         SAM01250
DMP FUNCTION COMPLETED
*DELET              SAMPL                                         SAM01260
DMP FUNCTION COMPLETED




// JOB    00 JAN 00 00.402 HRS                                      SAM01270
// DMP    00 JAN 00 00.403 HRS                                      SAM01290
*DUMPLET                                                          SAM01290

LET

PACK LABEL

01000

      DB/   DB            DB/   DB            DB/   DB            DB/   DB            DB/   DB            DB/   DB
NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR

.DCOM 0010  0000    .MCA  0070  0010    .DMON 0030  0080    .MON1 0040  00B0    .MON2 0030  00F0    .MON3 0010  0120
.SUP  0070  0130    .SUPA 0040  01A0    .SUPB 0050  01E0    .SUPC 0030  0230    .SUPD 0040  0260    .BLD  00F0  02A0
.BLDX 0030  0390    .BLDY 0030  03C0    .DMP  0090  03F0    .DMKB 0010  0480    .DMAF 0010  0490    .DCLE 0020  04A0
.DCLN 0030  04C0    .DCLT 0030  04F0    .DPLT 0040  0520    .DP1A 0040  0560    .DP2A 0040  05A0    .FLSH 0010  05E0
.FLS2 0010  05F0    .GDMV 0010  0600    .DMV2 0010  0610    .DMV3 0010  0620    .INTL 0030  0630    .LUP1 0030  0660
.LUP2 0030  0690    .SCNT 0050  06C0    .SCN2 0030  0710    .DSRC 0010  0740    .SCRD 0030  0750    .SQCH 0040  0780
.SRFL 0030  07C0    .SRF1 0020  07F0    .SRF2 0020  0810    .SRF3 0010  0830    .DEF1 0060  0840    .DEF2 0030  08A0
.DEF3 0030  08D0    .DEF4 0020  0900    .DEF5 0020  0920    .DLBL 0030  0940    .DWRD 0020  0970    .STCI 0040  0990
.SRPT 0030  09D0    .DFLE 0080  0A00    .DELT 0090  0A80    .DCPY 0010  0B10    .DCYI 0030  0B20    .FOR  0080  0B50
.FR28 0020  0BD0    .FR31 0010  0BF0    .FR19 0010  0C00    .FR30 0010  0C10    .FR29 0010  0C20    .FR02 0030  0C30
.FR03 0020  0C60    .FR04 0030  0C80    .FR05 0040  0CB0    .FR06 0030  0CF0    .FR07 0020  0D20    .FR08 0040  0D40
.FR09 0040  0D80    .FR10 0030  0DC0    .FR11 0030  0DF0    .FR12 0030  0E20    .FR13 0030  0E50    .FR14 0040  0E80
.FR15 0040  0EC0    .FR16 0040  0F00    .FR17 0040  0F40    .FR18 0040  0F80    .FR20 0040  0FC0    .FR21 0030  1000
.FR22 0030  1030    .FR23 0020  1060    .FR24 0030  1080    .FR25 0030  10B0    .FR26 0040  10E0    .FR27 0030  1120
.DUM1 0010  1150    .DUM2 0010  1160    .ASM  0060  1170    .CD10 0020  11D0    .DISK 0020  11F0    .PNCH 0010  1210
.PS01 0020  1220    .PS1A 0020  1240    .PS02 0020  1260    .PS03 0020  1280    .PS04 0020  12A0    .PS05 0020  12C0
.PS06 0010  12E0    .PS07 0020  12F0    .PS08 0020  1310    .PS09 0010  1330    .PS10 0020  1340    .PS11 0010  1360
.PS12 0020  1370    .PS13 0020  1390    .PS14 0020  13B0    .PS15 0020  13D0
```

```
      DB/   DB            DB/   DB            DB/   DB            DB/   DB            DB/   DB            DB/   DB
NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR

.ERMS 0020  13F0    .MUP  00D0  1410    .DUM4 0010  14E0    .DUM5 0010  14F0    .HELP 0060  1500    .CEX  0010  1560
.LET  0080  1570    ABRTQ 0004  15F0    AIRN  0019  15F4    AISQN 0019  160D    AISN                  RINDC 0006  1626
BINHX 0004  162C    CANCL 0009  1630    CARDN 0018  1639    CLERQ 0009  1651    CLOCK 0002  165A    COMGO 0005  165C
COMG1               CSPAR 000B  1661    CSPLS               DAOP  0018  166C    DATSW 0004  1684    DCBIN 0007  1688
DEFER 0016  168F    REPET               DELAY               CYCLE               DEFSU               DEQUE 000C  16A5
DIEXP 0007  16B1    DTRAP 000D  16B8    DVCHK 0002  16C5    EADD  000B  16C7    ESUB                  EADDR
ESUBX               ESBR                ESBRX               EBPA  0006  16D2    DATN  000D  16D8    EATAN
EAVL  0003  16E5    EABS                EAXB  0006  16E8    EAXBX               EAXI  0006  16EE    EAXIX
EDVR  0008  16F4    EDVRX               EDIV                EDIVX               ELD   0006  16FC    ELDX
ESTO                ESTOX               ELN   000B  1702    EALOG               EMPY  0004  170D    EMPYX
EPLFX 0002  1711    ESIGN 0004  1713    ESINE 000D  1717    ESIN                ECOSN               ECOS
FSQR  0007  1724    ESQRT               ETNH  0006  172B    ETANH               ETRTN 0004  1731    ETNTR
EXPN  000B  1735    EEXP                FARC  0004  1740    FATN  000C  1744    FATAN                 FAVL  0003  1750
FABS                FAXB  0006  1753    FAXBX               FAXI  0005  1759    FAXIX                 FCTST 0003  175E
FDIV  0008  1761    FDIVX               FDVR                FDVRX               FEROR 0014  1769    FREBF
FIOXT               GETBF               FREUP               SBCAL               FIXIX 0006  177D    FIXI
FLD   0005  1783    FLDX                FSTO                FSTOX               FLN   000B  1788    FALOG
FLOAT 0003  1793    FMPY  0005  1796    FMPYX               FPLFX 0002  179B
```

```
LET

PACK LABEL

 01000
```

```
      DB/   DB            DB/   DB            DB/   DB            DB/   DB            DB/   DB            DB/   DB
NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR

FSBR  000B  179D    FSBRX               FADD                FSUB                FADDX               FSUBX
FSIGN 0004  17A8    FSQR  0007  17AC    FSQRT               FTNH  0006  17B3    FTANH               FTRTN 0004  17B9
FTNTR               FXPN  0009  17BD    FEXP                HOLEB 0011  17C6    HOLEX 0004  17D7    HXBIN 0006  17DB
IABS  0003  17E1    IADDR 0002  17E4    IAND  0002  17E6    IEOR  0002  17E8    IFIX  0004  17EA    IOFIX 0004  17EE
IOR   0002  17F2    ISBAD 0002  17F4    ISTOX 0003  17F6    ITIMR 0005  17F9    LDFAC 0004  17FE    STFAC
SBFAC               DVFAC               LD    0002  1802    LEVEL 0009  1804    LSPAR 0013  180D    MAGT  0028  1820
MASK  0003  1848    MDEND 0007  184B    MFIO  0069  1852    MRED                MWRT                MCOMP
MIOAF               MIOAI               MIOFX               MIOIX               MIOF                MIOI
NORM  0004  18BB    OPMON 0002  18BF    OVERF 0002  18C1    PAPEB 0011  18C3    PAPHL 0014  18D4    PAPPR 0011  18E8
PAPTN 0016  18F9    PAUSE 0002  190F    PLSV  0004  1911    PLRS                PRT   0006  1915    RESMK 0003  191B
REWND 0016  191E    RCKSP               EOF                 SAVMK 0003  1934    SETCL 0003  1937    SLITE 0005  193A
SLITT               SNR   0003  193F    SPAR  0001  1942    SSWTC 0003  1943    STORP 000B  1946    SUBIN 0005  1951
SUBSC 0004  1956    SUSPN 000D  195A    TIME  0003  1967    TRAP  0008  196A    UFIO  001F  1972    URED
UWRT                UIOI                UIOF                UIOAI               UIOAF               UIOFX
UIOIX               UCOMP               UNMK  0003  1991    XMD   0005  1994    XMDS  0004  1999    XSQR  0004  199D
YFIO  0022  19A1    YRED                YWRT                YIOI                YIOF                YIOAI
YIOAF               YIOFX               YIOIX               YCOMP               CNVDT 0006  19C3    CNVDL
SETDT 0007  19C9    SETDL               ZIPCO 000B  19D0    PEFIB 0007  19DB
```

```
      DB/   DB            DB/   DB            DB/   DB            DB/   DB            DB/   DB            DB/   DB
NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR

FLBPE 001C  19E2    PEFLB               FIRPE 0005  19FE    ISIGN 0003  1A03    OPEN  0008  1A06    DINP  0017  1A0E
EBPRT 000A  1A25    HOLPR 000D  1A2F    XDD   000D  1A3C    FSINE 000B  1A49    FSIN                FCOSN
FCOS                ADRCK 000B  1A54    MDFIO 0038  1A5F    MDAF                MDAI                MDCOM
MDF                 MDFX                MDI                 MDIX                MDRED               MDWRT
FBTD  001A  1A97    FDTB                DSCR  0020  1A81    DSOR  0019  1A01    MDAIO 0020  1AEA    WSDSK
RDDSK               WDDSK               IOU   0005  180A    EXDIR 0085  180F    SPECL               BACK
QLEVL               RSCIO 009B  1BC4    BSCTR 0012  1C5F    RESER 000B  1C71    RELEA               .NRMT
STOP  0003  1C7C    EADD  000C  1C7F    EADDX               ESUB                ESUBX               ESBR
ESBRX               EATN  000E  1C8B    EATAN               EAVL  0004  1C99    EABS                EAXB  0005  1C95
EAXBX               EAXI  0007  1CA2    EAXIX               EDVR  000A  1CA9    EDIV                EDIVX
EDVRX               ELD   0006  1CB3    ESTOX               ESTO                ELDX                ELN   000C  1CB9
EALOG               EMPY  0005  1CC5    EMPYX               ESINE 000C  1CCA    ESIN                ECOSN
ECOS                ESQR  0008  1CD6    ESQRT               ETNH  0006  1CDE    ETANH               EXPN  000B  1CE4
EEXP                FSBR  000C  1CEF    FSBRX               FADD                FSUB                FADDX
FSUBX               FARC  0004  1CFB    FATN  000C  1CFF    FATAN               FAVL  0004  1D0B    FABS
FAXB  0006  1D0F    FAXBX               FAXI  0007  1D15    FAXIX               FDIV  0009  1D1C    FDIVX
FDVR                FDVRX               FIXI  0007  1D25    FIXI                FLD   0006  1D2C    FLDX
FSTO                FSTOX               FLN   000B  1D32    FALOG
```

```
LET

PACK LABEL

01000
```

```
      DB/   DB            DB/   DB            DB/   DB            DB/   DB            DB/   DB            DB/   DB
NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR

FLOAT 0002  1D3D    FMPY  0006  1D3F    FMPYX               FSQR  0008  1D45    FSQRT               FXPN  000A  1D4D
FEXP                IABS  0002  1D57    IFIX  0004  1D59    NORM  0005  1D5D    SNR   0002  1D62    XMD   0005  1D64
XMDS  0003  1D69    XSQR  0005  1D6C    BINDC 0006  1D71    BINHX 0004  1D77    DCBIN 0007  1D7B    EBPA  0006  1D82
EBPRT 000B  1D88    HOLEB 0012  1D93    HOLPR 000D  1DA5    HXBIN 0006  1DB2    PAPER 0011  1DB8    PAPHL 0014  1DC9
PAPPR 0011  1DDD    PRT   0005  1DEE    ECHAR 0005  1DF3    ECHRX 0023  1DF8    ECHRI               EGRID 0008  1E16
EPLOT 0005  1E23    ERULE 000B  1E28    EMOVE               EINC                FCHAR 0005  1E33    FCHRX 0023  1E38
FCHRI               FGRID 0007  1E58    FPLOT 0004  1E62    FRULE 000A  1E66    FMOVE               FINC
FTNH  0007  1E70    FTANH               PLOTI 0005  1E77    POINT 0008  1E7C    SCALE 0002  1E84    SCALF 0002  1E86
XYPLT 0007  1E88    TSTOP 0002  1E8F    TSTRT 0002  1E91    TTEST 0003  1E93    TSET                MGOTO 000A  1E96
MFIF                MIIF                MEIF                MIAR  0008  1EA0    MIARX               MFAR
MFARX               MEAR                MEARX               FLIP  0008  1EAB    PLOTX 000E  1EB3    MLIST 001F  1EC1
BSCTR 0001  1EE0    RSCCK 0017  1EE1    ZIPCO 000C  1EF8    USHOL 0008  1F04    HOLUS 0009  1F0C    USERC 0008  1F15
EBCUS 0009  1F1D    USPRT 0008  1F26    USTYP 0009  1F2E    PEFIB 0008  1F37    FIBPE 0005  1F3F    FLBPE 001E  1F44
PEFLB               DMPS  0022  1F62    DUMPS               DMPST               DMP                 DUMP
DMPHX               DMPDC               XDD   000D  1F84    RSCRT 0068  1F91    FSINE 000C  1FF9    FSIN
FCOS                FCOSN               FBTD  0019  2005    FDTB                DPIP  0096  201E    CECLD 00A2  20B4
DMPAN 00DD  2156    FSWN  0003  2233    .TEMP 2236  2280    .E    3180  2280
```

```
FLET

PACK LABEL

 01000

      SCTR/ SCTR          SCTR/ SCTR          SCTR/ SCTR          SCTR/ SCTR          SCTR/ SCTR          SCTR/ SCTR
NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR    NAME  WCNT  ADDR

.IHSV 0139  0318    /NPSV 8000  0451    .MESS 0008  0488    9DUMY 00A0  04C0    /SPSV 8000  0560    .EXEC 0069  05C7
/CLST 013E  0630    /CL01 0140  0631    /CL02 0140  0632    /CL03 03C0  0633    /CL04 03C0  0636    /CL05 0280  0639
/CL06 0280  063B    /CL07 03C0  063D    /E    00A0  0400
DMP FUNCTION COMPLETED
// END                    END OF SAMPLE PROGRAM.   00 JAN 00 00.636 HRS 99999999

CL WC OF  03E2 STORED AT  04E7
DMP FUNCTION COMPLETED
// XEQ SAMPL   FX0   00 JAN 00 00.225 HRS                                    SAM01180
00001
00005
// JOB   00 JAN 00 00.374 HRS                                               SAM01190
// DMP   00 JAN 00 00.375 HRS                                               SAM01200
*DELET              SAMPC                                                  SAM01210
DMP FUNCTION COMPLETED
*DELET              SAMPZ                                                  SAM01220
DMP FUNCTION COMPLETED
*DELET              SAMPY                                                  SAM01230
DMP FUNCTION COMPLETED
*DELET              SAMPA                                                  SAM01240
DMP FUNCTION COMPLETED
*DELET              SAMPB                                                  SAM01250
DMP FUNCTION COMPLETED
*DELET              SAMPL                                                  SAM01260
DMP FUNCTION COMPLETED
// JOB   00 JAN 00 00.402 HRS                                               SAM01270
// DMP   00 JAN 00 00.403 HRS                                               SAM01280
*DUMPLET                                                                    SAM01290
DMP FUNCTION COMPLETED
// END                    END OF SAMPLE PROGRAM. 00 JAN 00 00.417 HRS 99999999
 IBM 1800 DACS  MPX/BOM   00.419  00 JAN 00
SEN SW 0 ON ABSOLUTE LOADER
SEN SW 1 ON LOAD BP MONITOR
SEN SW 2 ON SET CLOCK VIA DATA SWS
SEN SW 3 ON SET DATE VIA DATA SWS
```

Appendix E. Summary of Console Sense, Program, and Data Switch Operations

Program	Console Switch Settings: Sense and Program	Console Switch Settings: Data	Use
BDUMP		All	First and last cylinder address
BDUPL		4-7 and	
		12-15	Drive selection
		0-ON	Reselect drive
		1-ON	2311 copy
BDWAP		All	Number of tries
		15-ON	Select drive 1
		14-ON	Select drive 2
	0-OFF		Reselect drive
BLIST	0-ON		List cards
	1-ON		Sequence check
BOM			
Absolute load		15-ON	Execute or load to disk
Utilities	0-1		Select utility function
	2-3		Select printer code
	0-ON		Sequential sectors (Disk dump function)
BPMON	7-ON		Console interrupt to get Supervisor
	7-ON		Keyboard request to get Supervisor
CA Trace		15-ON	Suppress CA trace
CE Coreload and 1442 Diagnostic			Uses only CE switches
Check/Stop Trace		All	Select trace address
Cold Start (To 2311)		0-ON	Cold start on 2311
		4-7	Mapped 1810 drive number
		8-11	Control unit address
		12-15	Phyical 2311 which contains the system pack
		All	Enter time
		All	Enter date
Cold Start (To 1810)		15-ON	Executive on drive 1
		14-ON	Executive on drive 2
		All	Enter time
		All	Enter date
Dump Analysis		15-ON	Proceed following checksum error
EAC	6-ON		HALT ON ERROR
FORTRAN	2-ON		Suppress listing
		15-ON	Enable trace
Full Trace		All	Set limits
		15-ON	Suppress print
Macro Assembler	2-ON		Suppress listing
Monitor trace		All	Select trace address
Supervisor	4-ON		Bypass not-ready 1443 printer
Trace define limit	5-ON		Console interrupt to select trace limit
	7-OFF		
*MON		1-ON	User option for over or underflow of system program
		0-ON	Abort load in conjunction with switch 1 option
		15-OFF	Print each new word
		14-OFF	Suppress all MON printing but error messages
*SRFLE		15-ON	Suppress list or punch

This page intentionally left blank.

Glossary-Index

This section consists of a glossary merged with an index. The only MPX control statements included in the glossary-index are those for which the format and use are explained in this manual.

IBM 1800 MPX Operating System
Operating Procedures
GC26-3725-4

READER'S COMMENT FORM

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

Possible topics for comment are:

Clarity Accuracy Completeness Organization Index Figures Examples Legibility

Cut or Fold Along Line

What is your occupation? _____

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

Please indicate your name and address in the space below if you wish a reply.

_____

_____

_____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Your comments, please . . .

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Cut Along Line

Fold Fold

First Class
Permit 40
Armonk
New York

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

IBM Corporation
Systems Publications, Dept 27T
P.O. Box 1328
Boca Raton, Florida 33432

Fold Fold

IBM 1800 MPX Operating Procedures (1800-36) Printed in U.S.A. GC26-3725-4

IBM

International Business Machines Corporation
General Systems Division
5775D Glenridge Drive, Atlanta, Georgia 30328
(U.S.A. only)

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

IBM 1800 MPX Operating Procedures (1800-36) Printed in U.S.A. GC26-3725-4

IBM®

International Business Machines Corporation
General Systems Division
5775D Glenridge Drive, Atlanta, Georgia 30328
(U.S.A. only)

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