IRIS R8

Operating System INSTALLATION AND CONFIGURATION MANUAL

POINT 4 DATA CORPORATION

JOANNA GORNIAK

POINT 4 DATA CORPORATION
2569 McCabe Way / Irvine, California 92714

IRIS R8

Operating System INSTALLATION AND CONFIGURATION MANUAL

Revision 06

NOTICE

Every attempt has been made to make this manual complete, accurate and up-to-date. However, all information herein is subject to change due to updates. All inquiries concerning this manual should be directed to POINT 4 Data Corporation.

POINT 4 Data Corporation
2569 McCabe Way
Irvine, CA 92714
(714) 754-4114

REVISION RECORD

PUBLICATION NUMBER: SM-030-0009

Revision	Description	Date
01	First Draft Version	12/31/81
02	Second Draft Version	02/01/82
03	Third Draft Version	03/01/82
04	First Preliminary Version	03/29/82
05	Update to reflect Rev 8.1 changes in DBUG (Section 2.2), Disc Partitioning (Section 4.4), DEFS List (Appendix B); correction to page 4-61; and addition of appendices covering Halts (Appendix E) and Traps (Appendix F)	08/10/82
06	Complete revision including incorporation of material for previously "to be determined" sections and addition of section on CTUTILITY.	03/01/83

LIST OF EFFECTIVE PAGES

Changes, additions, and deletions to information in this manual are indicated by vertical bars in the margins or by a dot near the page number if the entire page is affected. A vertical bar by the page number indicates pagination rather than content has changed.

Page	Rev	Page	Rev	Page	Rev
Cover	-				
Title	06				
ii thru xiv	06				
1-1 thru 1-13	06				
2-1 thru 2-40	06				
3-1 thru 3-31	06				
4-1 thru 4-32	06				
5-1 thru 5-103	06				
Appendix Title	-				
A-1 thru A-8	06				
B-1 thru B-10	06				
C-1 thru C-29	06				
D-1 thru D-10	06				
E-1 thru E-2	06				
F-1 thru F-9	06				
Comment Sheet	06				
Mailer	-				
Back Cover	-				

PREFACE

The IRIS Installation and Configuration Manual has two functions. It is intended as a guide to the initial installation and configuration of the IRIS Operating System. To that end descriptions and directions for the use of the necessary sysgen "tools" (i.e., IRIS installation and configuration service routines) have been included. The manual is also intended as a reference guide for the customization of the IRIS Operating System beyond the initial configuration process. The configuration section of this manual covers those processes which will aid the user in adapting IRIS to the needs of a particular installation.

For the convenience of the user, the following information is provided in Appendices:

Loading Software	Appendix A
Sysgen Log	Appendix B
Software Definitions	Appendix C
LPTD Driver File Listing	Appendix D
IRIS System Halts	Appendix E
Trap Numbers	Appendix F

Standard Notations For This Manual

This manual uses the following standard writing conventions:

User Input User input is always underlined; it may be a command shown in capital letters, a variable such as a filename shown in braces, or locations in memory indicated by an octal number.

<RETURN> Indicates a carriage return. It is required to activate command input. This is not shown unless it is the only command required, a second <RETURN> is required, or it follows a control character (i.e., <CTRL-Z> <RETURN>).

<CTRL-x> Indicates a control character where x is an alpha key. It is entered by holding down the CTRL key and pressing the alpha key indicated. Both keys are then released. A <RETURN> is not required unless otherwise noted.

{variable} Lowercase item enclosed in braces represents a variable such as a filename, password, etc.

Related Manuals

For related information see the following:

Title	Pub. Number
IRIS R8 Peripherals Handbook	SM-030-0015
IRIS Operations Manual	SM-030-0010
IRIS R8 User Manual	SM-030-0011
MIGHTY MUX User Manual	HM-042-0015
MARK 3 Computer System Manual	HM-081-0019
MARK 3 Peripherals Interface Manual	HM-081-0027
MARK 8 Reference Manual	HM-082-0021
POINT 4 Computer User Manual	

CONTENTS

APPENDICES

FIGURES

TABLES

Section 1
INTRODUCTION

This section covers the general preparation needed to set up an IRIS Operating System, the R8 Pico-N, data channel priorities, and describes IRIS components.

1.1 GENERAL PREPARATIONS

The primary focus of this manual is the installation and configuration of the system software. However, there is some overlapping in the initial installation of hardware and software. The following is a summary of the factors to be considered when setting up a system.

- Site preparation and maintenance
- Hardware selection: the computer and its peripheral devices
- Hardware installation and testing
- System Generation (initial installation of the IRIS Operating System)
- Configuring the system for a specific installation
- Testing

Attention should be given to environmental requirements:

- A telephone is recommended for maintenance purposes and it should be within easy reach of the computer and the master terminal.
- A separate line from the main power distribution box is recommended to prevent transients from elevators, air-conditioners, business machines, etc.

  Interference from equipment with frequent stops and starts may cause computer performance to be erratic.

1.2 PICO-N

An IRIS R8 Operating System requires that an R8 Pico-N be installed on the computer backplane. Without this device the IRIS Operating System will not function. The Pico-N is coded to enable specific POINT 4 application packages. It can also be coded to enable specific OEM packages. These packages should be specified when the Pico-N is ordered or returned for modification.

The R8 Pico-N should not affect normal operation of the computer or any of the peripheral devices. It is unnecessary to remove the Pico-N to run diagnostics.

If a hardware problem is suspected, the standard test routines (CPU exerciser, logic test, memory address test, memory checkerboard test, disc reliability test, etc.) should be run before installing the system software.

THE PICO-N ALWAYS REMAINS THE PROPERTY OF POINT 4 DATA CORPORATION. It is supplied under a nontransferrable license with each paid IRIS license.

1.2.1 PICO-N INSTALLATION

The Pico-N consists of a 100-pin connector with encapsulated circuitry (see Figure 1-1). Install the Pico-N as follows:

1. Turn off CPU power.

2. Install the Pico-N by pushing its connector over the 'A' side pins of any slot except the slot which contains the CPU.

3. Be sure the Pico-N is oriented properly (the top and bottom are clearly marked) and all 100 pins are in their connectors.

The Pico-N may be destroyed if installed incorrectly, e.g., shifted either right or left. It does not matter whether the selected slot contains a board. The Pico-N draws power from pins A97 through A100.

NOTE

> For installation of a Pico-N on a MARK 3 Computer, refer to the POINT 4 MARK 3 Computer System Manual.

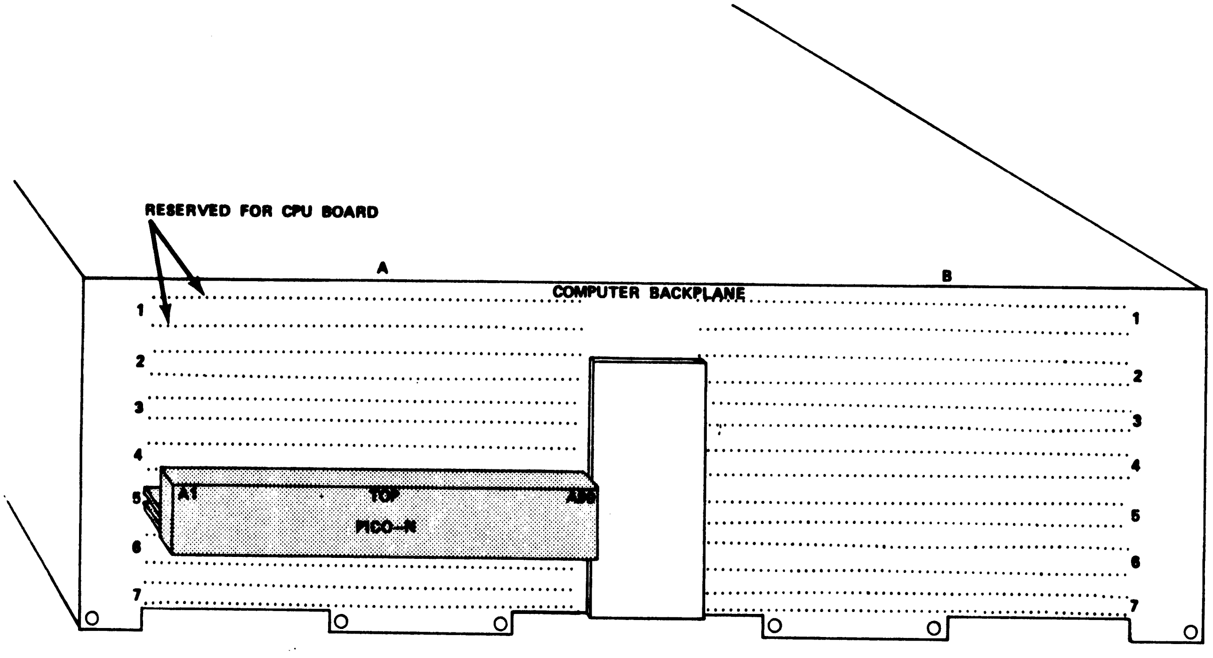


Figure 1-1. Pico-N Installation

1.2.2 TESTING PICO-N

If a problem with the Pico-N is suspected, test it as follows:

1. IPL into a full configuration.

   If, during the IPL, the system responds

   ```
   ??NO PICO-N??
   ```

   the Pico-N may be missing or not plugged in properly.

2. Check that the R8 Pico-N is plugged in correctly. If it is, the Pico-N may be defective (see Section 1.2.3).

1.2.3 PICO-N REPLACEMENT

If a Pico-N is defective, POINT 4 will supply a replacement. Notify the POINT 4 sales representative.

1.3 DATA CHANNEL PRIORITY

Data Channel priorities should be set (highest to lowest) as follows:

1. CPU
2. All disc controllers
3. All magnetic tape controllers
4. POINT 4 MIGHTY MUX
5. POINT 4 LCM(s)

Data Channel priorities may be set either by use of relative slots in the computer chassis or by backplane jumpering. The method depends on the type of computer. Consult the computer hardware manual for the proper method of setting up Data Channel priorities.

The order of interrupt priority is not important as long as all devices can interrupt.

EXCEPTION

> Some non-POINT 4 multiplexers may require the highest interrupt priority.

1.4 IRIS COMPONENTS

Table 1-1 is an alphabetical listing of standard IRIS R8 system components and their functions. Some of these components are loaded, while others are created when the system is configured.

TABLE 1-1. STANDARD IRIS R8 SYSTEM COMPONENTS

Name	Function
ACCOUNTS	Account directory on each logical unit which contains account parameters and charges for each user.
ACCOUNTUTILITY	BASIC program used for the maintenance of the account directory (i.e., ACCOUNTS file).
ASSEMBLE	Absolute Assembler processor.
ASSIGNPF	BASIC program used to move polyfile volumes from one logical unit to another.
BASIC	BASIC language editor and lister.
BASICTEST	BASIC readiness test.
BCONVERT	Processor which converts R7.x BASIC programs to the R8 format.
BLOCKCOPY	Stand-alone program for disc block transfers from one type of disc to another.
BTUP	Block Two Utility Package - Low-level debugger.
BUILDPF	BASIC program used to create and extend polyfiles.
BUILDPFERR	BASIC program which builds an error message file; used by the BUILDPF, QUERYPF, and KILLPF programs.

TABLE 1-1. STANDARD IRIS R8 SYSTEM COMPONENTS (Cont)

Name	Function
BUILDXF	BASIC program used to create indexed files.
BYE	Log-on/Log-off processor.
BZUD	Block Zero Utility Driver - Disc driver unique for each disc controller.
CHANGE	Processor used to change file characteristics.
CLEANUP	Processor used to realign block usage on a logical unit.
CLEANUPX	Same as CLEANUP; allows expansion of LU/0.
CONFIG	System file which contains information about the system's current configuration.
COPY	Processor used to copy files (except polyfiles).
COPYPF	BASIC program used to copy polyfiles; includes option to change polyfile names.
CTUS	Physical interface handler for cassette tape units.
DBUG	Debugging utility package for the POINT 4 MARK series and Nova*-type computers.
DDCOPY	Disc-to-disc copy utility - Unique for each disc controller.
DEC	Decimal arithmetic system subroutine module.

*Nova is a trademark of Data General Corporation.

TABLE 1-1. STANDARD IRIS R8 SYSTEM COMPONENTS (Cont)

Name	Function
DEFS	IRIS software definitions.
DGMX	Driver for a Data General 4060-type multiplexer.*
DISCSUBS	Library file of disc-resident system subroutines.
DISCUTILITY	Disc utility for POINT 4 LOTUS 700 or MARK 3 disc controllers.
DMAP	Disc map for each logical unit - A system file which is built by the sysgen procedure, an IPL, or by the INSTALL processor to map disc block usage.
DSP	Disc Service Processor - IRIS on-line debugging utility package.
EDIT	Text file editor.
EXTRAPORT	BASIC program used to start programs running on a phantom port.
FAULTHISTORY	Information file for system faults.
FAULTPRINT	Processor used to print Trap messages.
FOREIGN	Floppy disc driver used to read/write a specified sector from or to any diskette (regardless of file system) that POINT 4 hardware can support.
FORMAT	Processor used to create a formatted or contiguous data file.

*Not used on a Mark 3.

TABLE 1-1. STANDARD IRIS R8 SYSTEM COMPONENTS (Cont)

Name	Function
GUIDE	BASIC programs that give directions for the configuration of logical units, line printer drivers, the use of BLOCKCOPY, etc.
INDEX	Logical unit file directory - Index on each logical unit which contains the filename and real disc address for each file header.
INSTALL	Processor used to open a logical unit or to create a new logical unit.
KILL	Processor used to delete files.
KILLPF	BASIC program for the deletion of polyfiles.
LCMDIAG1.3	POINT 4 LCM diagnostic program.
LIBR	Processor which lists filenames and file information on a logical unit.
LPTD	Universal line printer driver for a Data General 4060-type multiplexer port.*
LPTM	Universal line printer driver for a POINT 4 310 or MARK 3 multiplexer port.
LPTP	Universal programmed/parallel I/O line printer driver (device code 17).*
LPTP51	Universal programmed/parallel I/O line printer driver (device code 51).*
MAIL	Processor for sending messages from one port to another.
*Not used on a MARK 3.	

TABLE 1-1. STANDARD IRIS R8 SYSTEM COMPONENTS (Cont)

Name	Function
MESSAGES	File containing standard messages.
MK8	Driver for the POINT 4 MARK 8 Computer.
MK8.OBJ	POINT 4 MARK 8 diagnostic program.
MMUX	POINT 4 310 Multiplexer driver.
MTA0	BASIC program interface driver for nine-track magnetic tape or cassette tape unit.
MTAS	Interface driver for a magnetic tape physical unit.*
MUX310DP	POINT 4 310 Multiplexer diagnostic program.
PHA	Phantom port driver.
PLOAD	Program loader used to load new files from paper tape.*
PORT	Processor used to change port attributes and display port activity.
PROTECT	Processor used to make saved BASIC programs unlistable.
PTP	Driver for a high-speed paper tape punch.*
PTR	Driver for a high-speed paper tape reader.*
PTM	Driver for a master Teletype reader/punch.*

*Not used on a MARK 3.

TABLE 1-1. STANDARD IRIS R8 SYSTEM COMPONENTS (Cont)

Name	Function
PZ	Page Zero software definitions in REX.
QUERY	Processor used to display file characteristics and account status information.
QUERYPF	BASIC program used to display polyfile characteristics.
R7TO8ACTCONV	BASIC program used to convert R7 ACCOUNTS files to R8 format.
REHASH	Processor used to reposition index file entries on a logical unit. It also identifies the entry slots which have never been used or have been deleted, and permits speedier INDEX access.
REMOVE	Processor used to close a logical unit.
RETRY	BASIC program used to list the number of unsuccessful disc access attempts that have occurred on each logical unit.
REX	Real-time Executive containing system level modules. Both SIR and DBUG reside in REX.
RUN	Run-time interpreter used to execute a BASIC program.
RUNMAT	Processor used to execute BASIC matrix algebra.
SAVE	Processor used to save BASIC programs.
SCOPE	System Command Processor - System command prompt (#), displayed by REX, indicates that the system is active and ready for input.

TABLE 1-1. STANDARD IRIS R8 SYSTEM COMPONENTS (Cont)

Name	Function
SELF.SN	POINT 4 MARK 5 diagnostic program.
SETTIME	BASIC program used to set system date and time.
SHUTDOWN	Processor that performs all necessary system shutdown functions (i.e., clearing the buffer pool) allowing an orderly transition for a stand-alone operation or a shutdown of the system.
SIR	System Initialization Routine. It is part of the REX file and contains the IPL sequence.
SOV	System Disc Driver Overlay - Unique to each disc controller, it is contained in REX.
SYMBOLS	File containing the Assembler symbols.
SYSL	System Loader used by the sysgen process and then discarded.
TERM.name	Terminal Translation Module - Contains translation tables and subroutines; unique for each type of terminal.
TERMS	Terminal Translator system subroutine module.
TTY	Driver for a secondary Teletype or CRT using device code 50/51.
U.CHANGE	BASIC program used to change the protection level of selected files residing on a logical unit in one job stream.

TABLE 1-1. STANDARD IRIS R8 SYSTEM COMPONENTS (Cont)

Name	Function
U.CONVERT	BASIC program used to convert selected BASIC programs on a given account from R7 format to R8 format in one job stream.
U.COPY	BASIC program used to copy selected files residing on a logical unit in one job stream.
U.KILL	BASIC program used to delete selected files residing on a logical unit in one job stream.
U.PROTECT	BASIC program used to make selected BASIC source code modules unlistable in one job stream.
U.SAVE	BASIC program used to save selected files residing on a logical unit in one job stream.
VERIFY	Processor used to checksum protected BASIC programs.
XREF	BASIC program used to produce a cross-reference listing of variables, GOTOs, etc., contained in a program module.

Section 2
INSTALLATION AND CONFIGURATION SERVICE ROUTINES

This section describes those procedures and system commands which are needed for the initial installation and configuration of the IRIS Operating System.

System commands (unless otherwise noted) are activated by pressing the <RETURN> key. A <RETURN> is not shown unless it is the only command required.

The service routines are discussed in the order in which they are needed:

BTUP	- Block Two Utility Package
DBUG	- Debugging utility package for POINT 4 MARK series and Nova-type computers
DSP	- IRIS on-line debugging utility package
PLOAD	- Program Loader used to load files from paper tape
DISCUTILITY	- Disc Utility for the LOTUS 700/710 and MARK 3 disc controllers
DDCOPY	- Disc-to-disc copy program

Other system commands are discussed in the IRIS R8 User Manual.

2.1 BTUP

The Block Two Utility Package (BTUP) is a debugging package that occupies a single block at real disc address two on logical unit zero.

BTUP is position independent. It normally occupies locations 77000 through 77377 octal in memory. When a disc command is given, BTUP uses the disc driver, BZUD (see Section 2.1.1), which must be in locations 76400 through 76777. Locations 77400 through 77777 are used as a block buffer area.

In addition, BTUP contains the configuration selection and the Initial Program Load (IPL) start-up sequence for REX (see Section 2.6).

To enter BTUP:

1. IPL the system.

2. At the prompt 'PRESS RETURN', enter

   0

   This loads BZUD and BTUP and transfers control to BTUP.

2.1.1 BTUP DISC TRANSFERS

BTUP uses the Block Zero Utility Driver (BZUD) for disc transfers. The BZUD disc driver is also used by DBUG, SYSL, SIR, INSTALL, CLEANUP, and SHUTDOWN. BZUD contains a simple disc driver that is unique for each disc controller.

The partitioning constants at words 1 and 2 in BZUD (locations 76401 and 76402 in memory) determine the disc drive and platter to which the real disc addresses point. The form of the drive and platter selection depends on the driver and is documented in the IRIS R8 Peripherals Handbook.

2.1.2 BTUP'S BAUD RATE

BTUP's baud rate is normally set to 9600 baud. If BTUP is to be used with a POINT 4 MIGHTY MUX (with master terminal mode) at a rate other than 9600 Baud, then the BPCON at word 375 of BTUP (location 77375 with BTUP at its normal location in memory) must be changed. BPCON must contain 5005x octal, where x specifies baud rate as follows:

x	Baud Rate	
0	110	
1	150	
2	300	
3	600	(or 19200 if MUX has the 19.2KB option)
4	1200	
5	2400	
6	4800	
7	9600	

2.1.3 BTUP COMMANDS

BTUP acknowledges execution of a command by printing a space. Illegal commands cause a question mark to be printed. If a disc read or write error occurs, the disc status word is printed followed by a question mark.

Each command to BTUP consists of a single letter followed by a <RETURN> or a <LINE FEED>. The command character may be preceded by an octal parameter as shown in Table 2-1.

WARNING

BZUD must be in locations 76400 through 76777 before a disc transfer command is given. There is no test in BTUP for the presence of BZUD, so the operation of the dG, dW, W, and : commands will be unpredictable if BZUD is not present.

TABLE 2-1. BTUP COMMANDS

Command	Description
a:	Open cell at address a.
a/	Display and open cell at address a.
<LINE FEED>	Display and open next cell. A question mark will be displayed if no cell has been opened.
n<LINE FEED>	Store number n in open cell. Display and open the next cell. Error if no cell has been opened.
<RETURN>	Press <RETURN> - no action.
n<RETURN>	Store number n in open cell. Open the next cell. Error if no cell has been opened.
aC	Copy cells a through a+377 into disc buffer area. Set up addresses 0 through 377 to point to this block.
aM	Move contents of the disc buffer area into locations a through a+377.
G	Set up addresses to point to real memory.
dG	Get (read) disc block d into disc buffer area. Set up addresses 0 through 377 to point to this block.
W	Write block in disc buffer back to its origin (d of the last dG or dW command).
dW	Write block in disc buffer onto disc at address d.
:	Resume IPL sequence.

where

a = any memory address (octal 177776 maximum)
d = any real disc address (RDA)
n = any octal number

2.2 DBUG

DBUG is a position-independent debugging package for POINT 4 MARK series and Nova-type computers. It is external to the IRIS Operating System. The DBUG supplied with IRIS R8 contains both paper tape and CTU (cassette tape unit) interfaces. Paper tape interface commands are the P, R, and V commands described in Section 2.2.2. Interface with a CTU is described in Section 2.2.3.

The REX disc file always contains a copy of DBUG. DBUG may be loaded into memory as follows:

1. Do an IPL.

2. At the prompt, 'PRESS RETURN', enter one of the following:

Enter	Description
1	Brings the system up into a full configuration. Retains DBUG, BTUP, BZUD, and the BZUD buffer area in memory.
2	Brings the system up into a minimum configuration. Retains DBUG, BTUP, BZUD, and the BZUD buffer area in memory.
3	Loads REX, SIR, BTUP, DBUG, and BZUD. Transfers control to DBUG.
<RETURN>	Brings the system up into a full configuration. Does not retain DBUG, BTUP, or BZUD in memory.

If option 1 or 2 is selected to enter DBUG,

- Press STOP, RESET, and START
- Re-enter DBUG at one of the following locations:
  - 73000 (saves the current registers)
  - 73001 (leaves previously saved registers intact, does not save current registers)

If option 3 is selected, control is transferred to DBUG.

2.2.1 DBUG PROCEDURES

All DBUG operations can be performed from the master terminal. This includes transfer of control to a user's program and back to DBUG. The user may interface with paper tape or a cassette tape unit (CTU). Operations are executed by typing the command letter followed by octal parameters as required (except ":" which is preceded by an octal parameter) and ending with a <RETURN> (see also "Multiple Command Entries" in Section 2.2.1.3).

Display of information may be temporarily interrupted by entry of

    <CTRL-S>   (= X-OFF)

The display may be restarted by entering

    <CTRL-Q>   (= X-ON)

If an error is made while entering control information, four choices are available for correcting it:

1. Press <ESC> or <ALT MODE> to delete the type-in and enable a new type-in.

2. Press <CTRL-H> or <RUBOUT> to backspace the last character typed in.

3. If an error is made in entering an octal value (not part of a symbolic instruction), type a few zeroes followed by the correct octal number (DBUG only uses the last six octal digits typed in for an octal word).

4. Press <CTRL-X> to cancel a partially entered command if the system is set up up to handle CTU (i.e., has CTU proms).

2.2.1.1 Re-Entry to DBUG

To re-enter DBUG manually, RESET and START at 73000 or 73001. DBUG's normal starting address is 73000, which saves the CPU status; to preserve the previously saved CPU status, start at 73001 (this also permits a return to a previous breakpoint via the H, J, or T command).

Since BZUD is always loaded with DBUG when the SHUTDOWN command is used, the G and W commands are available.

DBUG may be brought into memory (at a location other than LDBUG) along with a stand-alone program by including an @ symbol and an octal address following the filename. For example,

```
    SHUTDOWN <CTRL-E>{key}<CTRL-E>{filename} @6000 X6000
```

where

key - password assigned by the system manager (the default is X).

@6000 - brings DBUG into memory at location 6000 after loading the selected file or files. DBUG is loaded last, regardless of its position in the command line.

X6000 - specifies that execution is to begin automatically at location 6000.

DBUG may be brought into memory without loading a stand-alone program from the disc. For example,

```
    SHUTDOWN <CTRL-E>{key}<CTRL-E> @{address}
```

loads DBUG into memory at the specified address, and the computer halts.

It is necessary to do an IPL to bring up IRIS after a SHUTDOWN.

2.2.1.2 Addressing Modes

For many commands, DBUG allows either word or byte addressing, using either real memory addresses or "offset" (virtual) memory addresses based on an offset previously entered (by an F command). DBUG is also designed to allow addressing up to 64K words of memory. This is accomplished by having two word-addressing modes (real and virtual), and three byte-addressing modes (virtual plus two real modes: lower 32K and upper 32K). These modes are invoked by the optional second parameter "a" shown for commands D, E, H, I, J, L, and O (except that H and J do not permit byte addresses).

a	Description
omitted	word address, including offset
0	word address, absolute
1	byte address, using offset
2	byte address, lower 32K absolute
3	byte address, upper 32K absolute

For DBUG commands which do not require an "a" parameter, the addressing mode is word address including offset (if any).

2.2.1.3 Multiple Command Entries

A slash (/) allows multiple command entries on one line; it replaces the usual <RETURN>. For example:

```
B1234/B1400,1/J1234
```

NOTE

> Do not use with E, L, N, S, or Z, as it will not increment the operand address.

2.2.1.4 Notes On Using DBUG

The carry light flashes (except in I mode) while DBUG is waiting for an input character to be entered. This is a signal that DBUG is active and will respond to input.

DBUG normally occupies memory locations 73000 through 76377 octal, with re-entry at 73000 or 73001. However, DBUG may be moved at any time by use of its own MOVE instruction (even into upper 32K in a 64K system). After moving, the P command may be used to punch a tape of DBUG for the new location if desired. DBUG cannot punch itself into its own location because it changes certain cells in memory between the time it punches the checksum and the time it punches the data, which produces a checksum error.

To use DBUG with a POINT 4 MIGHTY MUX (with master terminal mode) at a baud rate other than its default rate, enter the desired PCB and PCON in words 2 and 3 relative to the beginning of DBUG. PCB is the port control block address to be used for setting up the MIGHTY MUX, and PCON is the port control word which specifies the desired baud rate and parity mode (e.g., 50057 for 9600 baud, even parity). To disable MIGHTY MUX setup, put a 0 in word 2 of DBUG.

2.2.2 DBUG FUNCTIONS

All DBUG functions are initiated from the master terminal, including transfer of control to a user's program and back to DBUG. DBUG may interface with a paper tape reader or a CTU. Paper tape interface is accomplished with the P, R, and V commands. DBUG function commands and paper tape interface commands are described in Table 2-2. Lower case letters represent parameters that must be entered as octal numbers. All command strings are activated by a <RETURN> unless otherwise noted. CTU interface commands are discussed in Section 2.2.3.

NOTE

> To use the G and W commands, BZUD must be in memory. DBUG is position independent, but BZUD must be at location 76400 (octal) to supply the disc driver for the G and W commands.
>
> DBUG can also operate with an IRIS R7-style BZUP (Block Zero Utility Package). If BZUP is used, it must be located 1000 words below DBUG. For example, if DBUG is at location 73000, BZUP should be at location 72000.
>
> If the G or W command is used and neither BZUD nor BZUP is at its proper location, DBUG outputs a bell, backslash (\), and the value 76400 (i.e., required memory address for BZUD) to indicate the problem.
>
> The partitioning constants in BZUD determine which physical unit is to be used.

TABLE 2-2. DBUG FUNCTIONS

Command	Description
A	Display the contents of registers A0, A1, A2, A3, the carry flip-flop, and interrupt status as they were at the time DBUG was entered. The interrupt status is displayed as an E for enabled or D for disabled. If DBUG was entered from a breakpoint, the display is preceded by that breakpoint location and a colon.
Bx,n (x≠0)	Insert breakpoint n (n=0 or 1; default is 0) in the user program at address x (see below for larger n). If a previous breakpoint n has been established (and has not been modified), it is restored to its original state before this new breakpoint is inserted. The breakpoint itself is a JMP @17 (for breakpoint 0) instruction, and DBUG puts a pointer to its breakpoint routine in location 17 octal. For breakpoint 1, location 16 is used. If control later reaches address x, then x is displayed followed by a display of the registers, carry flip-flop, and interrupt status as in A above. Each breakpoint requires its own page zero cell. If enough such cells are available, up to four breakpoints may be used (numbered 0 through 3). To create additional breakpoints or change their page zero cells, simply insert the desired page zero addresses at locations 10, 12, 14, or 16 relative to the beginning of DBUG. A zero at any of these locations marks the end of the breakpoint list. DBUG itself can be used to set breakpoints and end the list; then Q is used to confirm the new values. **NOTE** The Trace command works by pushing breakpoint zero ahead of itself; therefore, breakpoint 0 is not independently available while using the T command.

(Table continues on next page)

TABLE 2-2. DBUG FUNCTIONS (Cont)

Command	Description
B0,n	Remove breakpoint n (0 if n is omitted), restoring the instruction at that location. Note that a breakpoint cannot be put at location zero.
B	Remove all breakpoints that have been established.
Ca	If an F offset has been established and a>5, converts the absolute address a to virtual form and displays the address preceded by an F.
Cx,y (x≤5)	Change accumulator, carry flip-flop, or interrupt status. ● If x is 0, 1, 2, or 3, then y is stored as the saved value for accumulator x. ● If x is 4, then the saved value of the carry flip-flop is set to 0 or 1 depending on whether y is 0 (i.e., if y=0, set C=0; if y≠0, set C=1). ● If x is 5, the interrupt enable status (ION) is set to 0 (disabled) or 1 (enabled) depending on whether y is 0.
Dx,a (a≤3)	Dump memory in octal, beginning at location x, using addressing mode a. Eight words (or bytes if a byte-address mode is used) are displayed per line, with the address of the first word (byte) on each line.
Dx,n (n>3)	Dump memory in octal, beginning at location x, and displaying n words per line with the address of the first word on each line.

(Table continues on next page)

TABLE 2-2. DBUG FUNCTIONS (Cont)

Command	Description
Ex,a	Enable entry at address x, using addressing mode a. The address (changed to a word address if it was a byte address) is displayed, followed by a colon; a value (octal or symbolic) may then be entered, followed by a <RETURN>. The next address (x+l) will then be displayed and opened for entry, and entry continues into sequential cells until <ESC> is pressed to terminate entry. Relative addresses may be entered either in the form .±n or as an absolute address. Absolute addresses less than 400 (octal) are interpreted as page zero rather than relative. DBUG understands all standard assembler symbols and the arithmetic skips (SGR, SGE, SLS, SLE, SEQ, SNE, SKZ, SNZ, SSP, SSN, SGZ, SZN, SKE, and SKO), in addition to the following special CPU instructions: IOR (62677 = IORST) RDS n (DIA n,CPU = READS n) HLT (63077 = HALT) ITA n (DIB n,CPU = INTA n) IEN (60177 = INTEN) MSK n (DOB n,CPU = MSKO n) IDS (60277 = INTDS) **NOTE** • E<RETURN> without parameter entries causes the present content of the opened location to be displayed in both octal and symbolic form. • E<RETURN><RETURN> causes the next address to be displayed and opened for entry. • E<caret> (up-arrow) without parameter entries causes the previous address to be displayed and opened for entry. • E<slash> (/) without parameter entries causes the same address to be displayed and opened for entry. This feature enables the user to confirm that an entry is entered correctly and to examine it in octal and symbolic form.

(Table continues on next page)

TABLE 2-2. DBUG FUNCTIONS (Cont)

Command	Description
Fx,y	Establish an address offset; i.e., a fixed difference between a real absolute memory address and a virtual address as entered and listed in DBUG. The difference x-y (where x is the real memory address and y is the virtual address on the listing) is added to each address entered and subtracted from each address displayed. If y is not entered, then x is used as the offset. An F is displayed at the beginning of each line whenever a nonzero offset is in effect. Type F0 to revert to direct memory addressing.
F	Save the current offset value, and reinstate offset that was in effect before the current one was established. Displays the offset being reinstated. This allows the user to alternate between two different offsets (or between one offset and real memory).
Gx,y	Get a block from disc. Real disc address x is read into memory locations y through y+377. Gx will read into page zero, and G will read block zero (BZUD) into page zero. See NOTE in Section 2.2.2. If a disc error is detected, a bell, a backslash (\), and the disc controller status word are output.
GF	Get File. Assumes an IRIS-type file header block has been read into the 400-word block immediately below DBUG. Reads the entire file from disc, putting each block at the memory address determined by CORA (word 175) in the header block. If a memory address overlays DBUG or the block below it, transfer stops; DBUG outputs a bell, a backslash (\), and the offending memory address.
Hx,a	Halt with registers and carry restored. The instructions after the halt will restore the interrupt status and then execute a jump to location x, using word addressing mode a. INST STEP may then be used to step through the user's program.

TABLE 2-2. DBUG FUNCTIONS (Cont)

Command	Description
H	Same as Hx,a, except returns to the breakpoint from which DBUG was entered. See J below.
Ix,a	Input ASCII starting at location x, using addressing mode a (a colon is echoed following the <RETURN>). Then input string (similar to .TXTF pseudo-op in the assembler with left-right packing). Input is terminated by pressing <ESC>, which causes a zero byte (or word) to be stored.
Jx,a	Jump to location x (using word addressing mode a) with registers, carry, and interrupt status restored. Same as Hx except that it does not halt before jumping.
J	Return to user program at the breakpoint from which DBUG was entered, after restoring accumulators, carry, and interrupt status. Do not remove the breakpoint. May be used after setting a new breakpoint (same or different), in which case control is still passed to the old breakpoint location from which DBUG was entered. Displays a backslash if DBUG was not entered from a breakpoint.
Kx,y,z	Store the octal constant z in locations x through y, inclusive.
Lx,a	List program, both octal and symbolic, starting at location x, using addressing mode a. To terminate listing, press <ESC>. To list a program at a previous address, enter Lx<caret> (up-arrow) or Lx,a<caret>.
Mx,y,z	Move block in memory. Absolute locations x through y, inclusive, are moved to the area starting at location z. The source and destination areas may overlap in either direction without adverse effects. May be used to move DBUG as long as the destination area does not overlap the source area.

TABLE 2-2. DBUG FUNCTIONS (Cont)

Command	Description
Nx,y,z,m	Search for not-equal. Same as Sx,y,z,m except that it searches for a not-equal condition. **NOTE** Used with a caret (up-arrow), finds the last location below a given point where the search conditions are met.
Ox,a	Output ASCII. The contents of memory starting at location x (using addressing mode a) are displayed as text, two characters per word. Output is terminated if a zero byte is encountered. Control characters (<40 octal) other than <RETURN> are displayed as a caret followed by the corresponding printable character.
Px,y	Punch paper tape from memory locations x through y, inclusive. Will punch on high-speed punch (device code 13) if available and turned on, else punches on TTY (device code 11). To punch on the TTY, enter the command up to but not including the <RETURN>, then turn on the punch, and press <RETURN>. When the punching is complete, turn off the punch before entering the next command. Punches about 2 feet of leader before the data if this is the first P command since DBUG was started or since an end block plus trailer were punched.
Px	Punch an end block with starting address x, followed by about 2 feet of trailer.
P	Punch an end block without starting address, followed by about 2 feet of trailer.
Q	Query breakpoints. Displays the page zero cell corresponding to each available breakpoint and the memory address (if any) where that breakpoint is currently set.

TABLE 2-2. DBUG FUNCTIONS (Cont)

Command	Description
Rx	Read punched paper tape from the master Teletype if x=0 or none, or from the high-speed paper tape reader (device code 12) if x=1. If a checksum error occurs, or if an attempt is made to write into nonexistent memory or to overwrite DBUG itself, further reading is stopped, and the address where the error occurred is displayed. If the tape contains an end block with a starting address, the computer will halt with the starting address in A2. If CONTinue is then pressed, it will jump to the starting address.
Sx,y,z,m	Search locations x through y, inclusive, for the constant z. Each word is first ANDed with mask m before comparison with z. If m is not entered, it is assumed to be 177777; i.e., a search is made for an exact match with z. The use of the mask is best explained by an example: the command Sx,y,60025,160077 will search locations x through y for any I/O instruction for device 25. When a comparison is found, its address and contents are displayed in both octal and symbolic form. **NOTE** When used with a caret (up-arrow), finds the last location below a given point where the search conditions are met.

(Table continues on next page)

TABLE 2-2. DBUG FUNCTIONS (Cont)

Command	Description
Tx	Trace through user program for x steps, beginning where the last breakpoint was encountered or where a previous trace left off, whichever occurred last. Displays a backslash if no such starting point exists. If x=0 or 177777, tracing continues. If x is omitted, traces one step. To start tracing at a given location: 1. Enter a breakpoint at that location 2. Jump to that location (encounters breakpoint) 3. Enter desired trace command For every program step that is traced, displays the memory address, the instruction in symbolic form, the contents of the accumulators, carry and interrupt status. **NOTE** Trace works by pushing breakpoint 0 ahead of itself. Therefore breakpoint 0 is not independently available when using T.
Tx,y	Same as Tx except suppresses intermediate display unless location y is written into by the instruction being traced; i.e., the instruction is a STA, ISZ, or DSZ, and it addresses location y, regardless of addressing mode. Can be used in the form T0,y to determine if location y is ever written into by the user program.
U	Not used
Vx	Verify paper tape from TTY (x=0 or none) or PTR (x=1). If a verification error is found, its address is displayed.
Wx,y	Write a block on disc. Locations y through y+377 are written at real disc address x. Wx will write page zero on the disc. See NOTE in Section 2.2.2. If disc error is detected, a bell, a backslash (\\), and the disc controller status word are output.

TABLE 2-2. DBUG FUNCTIONS (Cont)

Command	Description
WF	Write File. Assumes an IRIS-type file header block has been read into the 400-word block immediately below DBUG, then writes the complete file from memory to disc.
Xx,y	Compute and display a "rotating" checksum over memory locations x through y. The checksum is produced by an ADDL instruction in order to detect a change (i.e., if two words in memory are swapped). Useful for testing if a change has occurred anywhere in a section of memory.
Yx	Set up a return delay (required on some CRTs for proper scrolling). After each carriage return/line feed, DBUG increments an accumulator from x to 0 before proceeding. For maximum delay, set x=0; for no delay, set x=177777. **NOTE** The default delay is stored in word 6 of DBUG.
Zx	Search for relative addressing reference. The 256 words centered on location x (using the "a omitted" addressing mode) are searched for any memory reference instruction that references location x using relative addressing. Any such instruction is listed in octal and symbolic form. **NOTE** When used with a caret (up-arrow) instead of <RETURN>, causes previous address to be displayed.

(Table continues on next page)

TABLE 2-2. DBUG FUNCTIONS (Cont)

Command	Description
x:value	Enter octal or symbolic value. The value given (either octal or symbolic) is stored at location x, using the "a omitted" addressing mode. If value is omitted, displays the present contents of location x followed by a colon, after which a new value may be entered. See the E command for more information. **NOTE** ● x:<RETURN> without parameter entries causes the present content of the opened location to be displayed in both octal and symbolic form. ● x:<RETURN> <RETURN> causes the next address to be displayed and opened for entry. ● x:<caret> (up-arrow) without parameter entries causes the previous address to be displayed and opened for entry. ● x:<slash> (/) without parameter entries causes the same address to be displayed and opened for entry. This feature enables the user to confirm that an entry is entered correctly and to examine it in octal and symbolic form.

2.2.3 DBUG - CTU INTERFACE COMMANDS

All CTU access commands consist of a control character, followed optionally by one or more parameters, and terminated by a <RETURN>. The only exception is <CTRL-X> which cancels any partially entered command immediately. Data is stored on tape in blocks of 256 bytes (128 words) each. Table 2-3 lists the CTU commands used in DBUG. All numeric parameters (x,y below) are in decimal, origin 0.

CTU commands in DBUG may be used in other CTU transfer procedures.

All commands that transfer data into or out of memory default to an initial memory address of 0. To start the transfer at some other address, precede the CTU command with:

{Memory address (octal)}: <RETURN>

DBUG will then display the contents of the chosen location, followed by a colon. This allows examination of the word before starting the tape transfer. Then type the CTU control character (e.g., <CTRL-R> or <CTRL-W>, followed by its parameters and a <RETURN>.

Table 2-4 is a quick-reference guide to the commands used for data transfer from a source to a destination.

TABLE 2-3. CTU COMMANDS IN DBUG

Control Character/ Parameters	Description
<CTRL-A>x,y	Access CTU buffer, i.e., transfer buffer into memory. Transfers y bytes starting at byte x. Default = 256 bytes starting at byte 0.
<CTRL-B>x	Write CTU buffer to tape, at block x.
<CTRL-D>	List directory (index) from tape, if tape is so formatted.
<CTRL-E>	Enquire (error status).
<CTRL-F>	Fill CTU buffer from memory (128 words).
<CTRL-I>x	Initialize (format) selected track to x+1 blocks of 128 words each. Maximum = I999 for 1000 blocks.

TABLE 2-3. CTU COMMANDS IN DBUG (Cont)

Control Character/ Parameters	Description
<CTRL-K>file	Kill the named file, i.e., erase its name from the directory.
<CTRL-O>file	Open the named file, if it is in the directory.
<CTRL-O>file,x,y	Create a directory entry for the named file (max. 5 chars.), starting at block x and containing y+1 blocks of 128 words each.
<CTRL-P>x,y	Put into CTU buffer from memory, transferring y bytes beginning at byte x in the buffer. Default = 256 bytes starting at byte 0.
<CTRL-R>	Read the open file from tape into memory.
<CTRL-R>x,y	Read from tape into memory; read y+1 blocks starting at block x.
<CTRL-S>x	Seek to block x on tape.
<CTRL-T>n	Select track n (0 or 1).
<CTRL-V>	Verify; i.e., read from tape into CTU buffer, checking checksum.
<CTRL-W>	Write from memory to tape into the open file, if any.
<CTRL-W>x,y	Write from memory to tape, writing y+1 blocks starting at block x.
<CTRL-X>	Cancel partially entered command (no <RETURN> required).
<CTRL-Z>	Rewind tape to starting position.

NOTE

<ESC> exits CTU mode and reverts to normal DBUG commands, but does not cancel any partial command that may already have been transmitted to the CTU. Use <CTRL-X> to cancel a partial command.

TABLE 2-4. SUMMARY AND OVERVIEW OF DATA TRANSFER COMMANDS

Source	Destination	Command
Tape	Memory	<CTRL-R>
Memory	Tape	<CTRL-W>
Tape	Buffer	<CTRL-V>
Buffer	Tape	<CTRL-B>
Buffer	Memory	<CTRL-A>
Memory	Buffer	<CTRL-F> complete buffer <CTRL-P> selected byte(s) only

2.3 DISC SERVICE PROCESSOR (DSP)

DSP is an on-line interactive utility package for the debugging and servicing of processors and other files under IRIS. Any location in memory or any file on disc can be accessed by the use of DSP. The system manager may allow limited access to DSP for authorized accounts (see Section 5.11.2.3).

CAUTION

DSP is a powerful tool! Use with care!

2.3.1 DSP ACCESS/EXIT

To use DSP, first log on to the manager's account. DSP is accessed as follows:

```
    DSP <CTRL-E>{key}<CTRL-E>
```

where key is the password assigned by the system manager (the default password is X).

DSP may be exited either with <CTRL-C> or the X command.

- If you exit DSP using <CTRL-C>, it may be reentered from the same terminal without a password. It will have retained the previously selected context (i.e., file, disc block, or memory).
- To prevent unauthorized use of DSP, be sure to exit with an X command when leaving the terminal.

2.3.2 USING DSP

Unless otherwise noted, a <RETURN> is required to activate the command string. The <RETURN> is not shown unless it is the only command required.

Any command which follows an F, G, or H command, examines and/or modifies memory and operates either on real memory, on a file, or on a disc block.

Any address may be specified as a byte address by adding a hyphen to the address. For example, D3025- will dump bytes starting with the right-hand byte of word address 1412, and E17000- will allow entry of bytes starting at the left-hand byte of word address 7400. The contents of any byte address may not exceed 377 octal. If a byte address is given when an enabled driver file (i.e., $file) is selected, then that byte address in real memory is referenced; this eliminates the need to select real memory to examine the driver's buffers.

The memory-resident copy of a discsub or driver (i.e., $file) may be addressed by appending an apostrophe to any address x, where x is the virtual address (as shown in Table 2-5). Using the apostrophe (') the user may examine or modify the memory-resident copy as if it were at the location shown on the assembly listing.

DSP will accept lower case command letters in place of the upper case letters shown on the following pages, except that the N in the LxN command must be entered as upper case.

For a description of the commands used in DSP see Table 2-5.

TABLE 2-5. DSP FUNCTIONS

Command	Description
x:v	Insert the value v at address x. This is very useful for entering into a single memory location. The value v may be either a symbolic instruction or an octal number. If v is omitted, a zero is written into address x. See the E command for more information.
Ax	Append the block which is to contain address x (x does not have to be on a block boundary) to the file selected by the last F command. The first memory address and the real disc address of the appended block will be displayed. The block is filled with 077377 halt instructions.
Bx	Insert a breakpoint at address x. This command is meaningful only if the specified file is a runnable processor. If that processor is then used on the same port, and the breakpoint is encountered, control will revert to DSP, and the contents of the registers and carry flip-flop are displayed. The breakpoint is cleared when it is encountered, and it is also cleared by any F, G, H, or X command. It is impossible to resume processor execution after encountering the breakpoint.

(Table continues on next page)

TABLE 2-5. DSP FUNCTIONS (Cont)

Command	Description
Bx{cond'n}	Insert a conditional breakpoint at address x. A breakpoint may be conditional on a register containing a specified value (indicated by Ar=v, where r is a register number 0 to 3, and v is an octal value), and/or conditional on a memory cell containing a specified value (indicated by x=v, where x is a memory address), and/or the breakpoint may be activated only after executing the instruction at the breakpoint location a specified number of times (indicated by an octal value by itself). For example B7235,A1=260,225=16003,4 will breakpoint the fourth time location 7235 is reached with the value 260 in register A1 and the value 16003 in memory location 225. The conditions may be given in any order, and the memory location may be specified indirectly; e.g., @37422=177723 means that the contents of location 37422 is used as a pointer to a cell that is to be checked for the value 177723.
C{command}	The "command" given is passed on to SCOPE as a system command. This is equivalent to pressing <CTRL-C> and then entering the command.
Dx	Dump octal starting at address x. The contents of storage starting at location x are printed in octal, eight words per line. The address of the first word of the line is printed at the beginning of each line. Listing may be terminated by pressing <ESC>.
Dx,y	Dump table starting at address x. Prints storage starting at location x in octal, y words per line; y ranges from 1 through 10 (octal). The address of the first word in each line prints at the beginning of the line. <ESC> terminates dump.

TABLE 2-5. DSP FUNCTIONS (Cont)

Command	Description
Ex	Enter octal or symbolic sequentially in memory starting at address x. Each entry must be followed by a <RETURN>. If <RETURN> is pressed without a preceding entry, a zero is stored at address x. Machine instructions may be entered in symbolic form, but the device address must be given in octal (rather than using device name) in I/O instructions (e.g., 10 rather than TTI). Labels may not be used, but absolute addresses will be converted to relative if possible. Press <ESC> to terminate entry mode.
F	Select real memory to be examined and/or modified.
F{filename}	Select the file identified by filename to be examined and/or modified. Logical unit zero is assumed unless given in the form LU/filename, where LU is the logical unit number in decimal. **NOTE** If an extended random file is selected, any address x given will refer to a location in the header extenders rather than to the data blocks.
F@	Select this port's active file to be examined and/or modified. The form F@n will select the active file of port number n to be examined and/or modified. The main memory address in the active file header is ignored, and all addressing is relative to the beginning of user storage in the partition.
F.	Select the body of the file of the currently selected file header block (i.e., selected by an H command) for examination and/or modification. An error message is displayed if a file's header is not currently selected.

TABLE 2-5. DSP FUNCTIONS (Cont)

Command	Description
Gu/x or Gx	Select the disc block at real disc address x on logical unit u (where u is in octal) to be examined and/or modified by all following DSP commands. In this mode, only addresses less than 400 octal will be accepted. The simple form Gx assumes logical unit zero.
H	Select the header block of the currently selected file to be examined and/or modified. In this mode, only addresses less than 400 octal will be accepted.
Ix:{text}	Input ASCII string, where "text" is any string of characters terminated by <RETURN>, starting at address x. The result is identical to use of assembler pseudo-op .TXTF with reverse packing (i.e., preceded by .TXTM 1). <RETURN> may be imbedded in the string by entering <CTRL-Z>.
Jx,y	Search for potential address errors. Scans from address x-200 through x+177 for all relative reference instructions spanning address x that are less than y words from maximum relative displacement; i.e., any place that an address error would be caused by inserting y lines of code at location x. Displays these instructions in octal and symbolic form.
Kx,y,z	Store the octal constant z in locations x through y, inclusive.
Lx	List both octal values and symbolic Assembly language instructions starting at address x. Output must be terminated by pressing <ESC>.
LxN	Same as Lx except only the Assembly language instructions are printed.

TABLE 2-5. DSP FUNCTIONS (Cont)

Command	Description
Mx,y,z	Move the contents of locations x through y, inclusive, to locations starting at z. The destination will receive the contents of the original source, even if source and destination overlap.
Nx,y,z	Search location x through y inclusively for a location not equal to the octal constant z. If found, displays the location and its content in octal and symbolic form.
Nx,y,z,m	Same as Nx,y,z but the contents of each cell are ANDed with mask m before being compared with constant z. For example, the command N400,1120,53,101777 searches bits 101777 at locations 400 through 1120 for any instructions not equal to octal 53.
Ox	Output ASCII string starting at address x. Output terminates on any byte equal to 0, 200 octal, or if <ESC> is pressed. Control characters (<40 octal) are displayed with a caret followed by the corresponding printable character.
Px,y	Punch locations x through y, inclusive, on the high-speed paper tape punch in binary loader format. If the system does not have a high-speed punch (no $PTP driver) then DSP attempts to use the master terminal ($PTM driver). **NOTE** Leader is automatically punched when the first Px,y command is given.
Px	Punch an end block with a starting address x, which must be nonzero, then punch trailer. Must be preceded by at least one Px,y command.

TABLE 2-5. DSP FUNCTIONS (Cont)

Command	Description
P	Punch an end block with no starting address, then punch trailer. Must be preceded by at least one Px,y command.
Qx	Query cell continuously. Repeatedly displays the contents of address x in octal, allowing a swap after each display. May be used from one terminal to monitor changes to a cell, either in memory or in a disc file, while executing tasks from another terminal to cause such changes. Terminate by pressing <ESC>.
R	Read binary-format paper tape into the destination selected by last F, G, or H command. Each tape record (about four inches) is read into a buffer and checksummed before data is stored. The first 21 words octal of the last breakpoint snapshot (see U and Y commands) will be lost because the same buffer area is used. If the system does not have $PTR enabled, then $PTM will be assumed. See "Copy Processor" in the IRIS R8 User Manual for restrictions on using $PTM.
Rx	Same as R except that all addresses on the tape are displaced the same amount so that the first word on the tape goes into address x, which must be nonzero.
Sx,y,z	Search locations x through y, inclusive, for the octal constant z. If found, displays the location and its content in octal and symbolic form.
Sx,y,z,m	Same as Sx,y,z except that the contents of each cell are ANDed with mask m before being compared with constant z. For example, the command S400,1120,53,101777 searches locations 400 through 1120, inclusive, for any instruction referencing location 53.

TABLE 2-5. DSP FUNCTIONS (Cont)

Command	Description
T	Not used.
Ux	Display snapshot yanked into FMAP cells of active file at last breakpoint. Start display (in octal dump format) at virtual address x where y <= x <= y+100 and y is the snapshot address set by the last Y command. **CAUTION** The addresses will be wrong if a different Y command has been given since the breakpoint was encountered.
V	Verify paper tape. This and the Vx command are the same as the respective R commands except that information from the tape is compared with the contents of the selected file (or memory) instead of being stored. If a difference is detected, the address and the word from storage are displayed.
Wu/x or Wx	Write the disc block selected by the last G or H command on disc at real disc address x of logical unit u. This command is rejected if u/x is not a legal real disc address or if a single disc block has not been selected. The simple form Wx assumes logical unit zero.
X	Exit from DSP, clear any existing file selection or breakpoint, and prevent re-entry to DSP without the password.
Yx	Set first address of 101 word (octal) memory area to be yanked into the FMAP cells of the active file header as a memory "snapshot" when a breakpoint is encountered. If x=0, do not yank any area of memory.

TABLE 2-5. DSP FUNCTIONS (Cont)

Command	Description
Zx	Search for relative reference. The 256 words centered on location x are searched for any storage reference instruction that references location x using relative addressing. Any such instruction is displayed in octal and symbolic form.
Zx,y	Same as Zx except a search is done for each address x through y.
;	Comment. Any line starting with a semi-colon will be ignored by DSP. This is used mainly to include comments on patch tapes.

2.4 PROGRAM LOADER (PLOAD)

PLOAD is a processor for loading files from paper tape to disc. It is run from the master terminal (device code 10/11). While in operation, PLOAD disables interrupts and prohibits other jobs from running. After a <CTRL-C> exit, the system resumes any operations which were running before PLOAD was called.

A <RETURN> is required following the command string unless otherwise noted. The <RETURN> is not shown unless it is the only command required.

To use PLOAD, log on to the Manager account at the master terminal as shown below. User input is underlined.

Prompt/Input	Description
#PLOAD	
PROGRAM LOADER	

WHICH READER: TTY (0) or PTR (1)? {n}	n=0 - for master Teletype reader (device code 10)
	n=1 - for tape reader (device code 12)

NAME ? {filename}	Enter the filename (1-14 characters).
TYPE ? {type word}	Enter the file type word - An octal number of 1-5 digits (see Table 2-6)

To exit PLOAD and return to command mode press <CTRL-C> (no <RETURN> required)

The file types for specific IRIS files on LU/0 are given in Section 3 of this manual; they are also summarized in the Sysgen Log (see Appendix A).

If PLOAD prints READER OK? then the read process has aborted due to a reader timeout. Try loading the tape again from the beginning.

A second failure is an indication of either a bad tape or a hardware problem in the tape reader.

NOTE

> PLOAD or COPY will load any stand-alone paper tapes (e.g., Diagnostics) supplied by POINT 4 Data Corporation.
>
> Papertapes not supplied by POINT 4 Data Corporation may use a punch format which is incompatible with IRIS even though they can be loaded by the binary loader.

TABLE 2-6. IRIS FILE TYPE CODES

Type	Description
0	Permanent System File
1	System File or Processor
2	BASIC Program or Processor
3	Stand-alone Program or Processor
30	Text File
31	Formatted File
32	Contiguous File
36	Peripheral Device Driver

2.5 DISC UTILITIES

IRIS supports three different disc utilities: the first is DISCUTILITY which copies disc-to-disc and disc-to/from-other media, etc.; the second is DDCOPY which copies disc-to-disc only; the third is BLOCKCOPY which copies selected blocks from one location to another or to another disc (see the IRIS Operations Manual for BLOCKCOPY procedures).

DISCUTILITY is available for systems using the POINT 4 MARK 3 and LOTUS 700/710 disc controllers. DDCOPY is available for most systems which do not have a POINT 4 controller.

2.5.1 DISCUTILITY

DISCUTILITY is a stand-alone utility package for the POINT 4 LOTUS 700 Disc Controller or the MARK 3 Computer System. It contains several program options depending on the computer in use. DISCUTILITY programs for POINT 4 MARK 3 Computer Systems include:

- Copy* (disc-to-disc)
- Save* (copies disc-to-tape, requires streamer tape unit)
  (copies disc-to-floppy, requires floppy disc unit)
- Restore* (copies tape-to-disc, requires streamer tape unit)
  (copies floppy-to-disc, requires floppy disc unit)
- Verify* (disc-to-disc verify)
  (floppy-to-disc verify, requires floppy disc unit)
  (tape-to-disc verify, requires streamer tape unit)
- LOTUS 700 or 710 nonzero LU-to-MARK 3 nonzero LU disc-to-disc conversion* (requires same drive type on both systems)
- Format and 8-pass analyze
- Quick format and 2-pass analyze (for specialized hardware testing only)
- Streamer tape re-tension
- Re-IPL option
- Automatic chaining of bad disc media to alternate tracks

DISCUTILITY programs for POINT 4 LOTUS 700 or 710 Disc Controller systems include:

- Copy* (disc-to-disc)
- Verify* (disc-to-disc)
- Format and 5-pass analyze
- Quick format and 2-pass analyze (for specialized hardware testing only)
- MARK 3 nonzero LU-to-LOTUS 700 or 710 nonzero LU disc-to-disc conversion* (requires same drive type on both systems)
- Re-IPL option
- Automatic chaining of bad disc media to alternate tracks

*Allows selection of starting cylinder number and number of cylinders.

These operations are performed on the basis of parameters entered by the user. The program is entirely interactive, guiding the user through the required steps. If there is any doubt as to parameter entries, etc., HELP modules can be invoked by entering an H in response to any question.

The use of DISCUTILITY requires that the system be shut down. To invoke the DISCUTILITY program enter

```
    SHUTDOWN <CTRL-E>{key}<CTRL-E>DISCUTILITY
```

where key is the password assigned by the system manager (the default is X).

Then follow the instructions displayed on the terminal. While in operation, the completion of various stages of the procedure are reported. Hardware failure is reported by displaying the status of the controller as well as any error messages.

2.5.2 DDCOPY

DDCOPY is a stand-alone utility program which copies disc-to-disc. As with DISCUTILITY, DDCOPY requires that the system be shut down. Unlike DISCUTILITY, it does not guide the user through its various phases.

Command strings entered by the user are underlined. Each command is activated by a <RETURN>. The <RETURN> is not shown unless it is the only input required.

In general terms, the procedure for using DDCOPY is as follows:

1. To invoke DDCOPY, shut down the system to DDCOPY by entering

   SHUTDOWN <CTRL-E>{key}<CTRL-E>DDCOPY

   where key is password assigned by the system manager (the default password is X).

2. In memory, location 401 is the source constant and location 402 is the destination constant unless otherwise noted in the IRIS R8 Peripherals Handbook. These constants must be entered by the user via the front panel or the virtual console (i.e., MANIP on a POINT 4 Series Computer). Refer to your DDCOPY listing for these constants.

3. DDCOPY's starting address is location 400. Use the front panel or virtual console to start DDCOPY at location 400.

4. Upon completion, DDCOPY halts and the run light goes out. Halts specific to DDCOPY are as follows:

63077	-	Good completion
67077	-	Irrecoverable Read error on source
73077	-	Irrecoverable Write error on destination
77077	-	Disc time-out

NOTE

As a stand-alone program, DDCOPY runs when the IRIS Operating System is deactivated. A halt for DDCOPY may have the same code (i.e., numbers) as a halt occurring under IRIS but the cause is different.

For *any* Halt (other than a good completion), the disc address and status word are contained in the following registers:

A0	-	Disc Status Word
A1	-	Disc Address

For a disc that is too large for a 16-bit disc address, check the following registers instead:

A0	-	Disc Status Word
A1	-	Cylinder Number
A2	-	Track and Sector Number

5. Remove the backup cartridge or disc pack.

6. An IPL <u>must</u> be performed after using DDCOPY to bring up IRIS.

CAUTION

> If the CONTinue switch is pressed after any Halt other than a 63077, up to a complete cylinder could be lost because the copy process resumes at the next cylinder of the disc.

2.6 INITIAL PROGRAM LOAD (IPL)

Initial Program Load (IPL) is a procedure that reads the IRIS Operating System from disc into memory. Several options are available that determine how the operating system is loaded.

Option   Description

0   Loads two blocks containing BZUD and BTUP. Transfers control to BTUP.

1   Brings the system up into a full configuration. Retains DBUG, BTUP, BZUD, and the BZUD buffer area in memory.

2   Brings the system up into a minimum configuration. Retains DBUG, BTUP, BZUD, and the BZUD buffer area in memory.

3   Loads REX, SIR, BTUP, DBUG, and BZUD. Transfers control to DBUG.

<RETURN>   Brings the system up into a full configuration. Does not retain DBUG, BTUP, or BZUD in memory.

Section 3
PAPER TAPE SYSTEM GENERATION PROCEDURE

The System Generation (sysgen) procedure is required only for the initial installation of a completely new IRIS system. Procedures for later additions and modifications are discussed in Section 5 (System Configuration) and in the System Programmer's Manual.

IRIS Operating Systems are supplied by POINT 4 on various media for the POINT 4 MARK Series computers.

1. MARK 5/8 and Nova-type computer systems:
   - Cassette tape (requires CTUTILITY, see Section 4.3)
   - Disc pack (see Appendix A.1)
   - Paper tape (requires sysgen)
2. MARK 3 System:
   - Cassette tape (requires CTUTILITY, see Section 4.3)
   - Disc pack (see Appendix A.1)
   - Floppy disc (see Appendix A.2)
   - Streamer tape (see Appendix A.3)

The sysgen procedure for paper tapes is described in this section and CTUTILITY procedure is described in Section 4.

3.1 PAPER TAPE SYSGEN PREPARATIONS

Before beginning the sysgen procedure make sure that all needed materials (tapes, disc packs, etc.) are at hand and that the requirements of the particular system to be installed have been specified.

3.1.1 PAPER TAPE SYSGEN MATERIALS

- Diagnostic programs
- IRIS sysgen paper tapes
- IRIS Sysgen Log
- One disc pack to set aside as a master copy
- Several disc packs for backup
- Pico-N
- Paper tape reader and controller

3.1.2 SYSTEM REQUIREMENTS

Make a list of system requirements. These include:

- CPU memory (must be 32K or more)
- Disc partitioning
- Port requirements
- Peripheral devices to be configured

3.1.3 SYSGEN LOG

A Sysgen Log serves both as a checklist and a record of the system generation procedure. It guides the user through the sequence of steps required to load the IRIS Operating System via paper tapes. See Appendix B for a sample Sysgen Log.

Fill out an IRIS Sysgen Log as the sysgen proceeds and save it for later reference. It is necessary to have a record of the sysgen in case of problems and, especially, for later modifications.

3.1.4 PAPER TAPE SYSGEN SUMMARY

The sysgen process consists of ten steps. A summary of these steps is given in Table 3-1.

TABLE 3-1. SYSGEN SUMMARY

Step	Description	Section
1	Hardware Diagnostics	3.2
2	Initial Loading of Memory Loading DBUG Initializing Memory Loading BTUP, REX, SYSGEN, SOV, and BZUD	3.3 3.3.1 3.3.2 3.3.3
3	System Loader (SYSL) Reporting Bad Blocks DISCSUBS Group #1 Initial System Files Exit SYSL and IPL	3.4 3.4.1 3.4.2 3.4.3 3.4.4
4	System Initialization Routine (SIR) Date and Time Log-on/Log-off	3.5 3.5.1 3.5.2
5	Initial System Configuration System Partition Size System Information Table Memory-Resident DISCSUBS Table Defining Logical Units Multiplexer Ports	3.6 3.6.1 3.6.2 3.6.3 3.6.4 3.6.5
6	Loading DISCSUBS Paper Tape Reader (device code 12) Master Terminal Reader (device code 10)	3.7 3.7.1 3.7.2
7	Loading Drivers and Processors	3.8
8	Loading Utilities and Text Files Loading BASIC Programs Loading Stand-alone Utilities Loading Text Files Format USERID File	3.9 3.9.1 3.9.2 3.9.3 3.9.4
9	Final Configuration	3.10
10	Backup	3.11

3.2 HARDWARE DIAGNOSTICS

The first step of the sysgen process consists of running computer diagnostics.

The diagnostics listed below are suggested aids. Not all of these programs are supplied with the computer. However, since these programs have been found to be most useful, it is wise to obtain them.

1. CPU Exerciser - If the computer is new, the Exerciser should be run overnight.

2. Memory Address Test (all memory).*

3. Memory Data Test (all memory).*

4. Disc Reliability Test - Thoroughly test all disc surfaces. If a problem occurs, it must be corrected before continuing the sysgen. The Disc Reliability Test should be left to run overnight.

5. POINT 4 Multiplexer Test (including the Q-test).

NOTE

> On a POINT 4 MARK Series Computer, use Self Test as a CPU and memory diagnostic.

*It is of particular importance that these tests be run.

3.3 INITIAL LOADING OF MEMORY BY PAPER TAPE

A number of system components have to be loaded into memory before the new system can begin to function. These components are:

DBUG	(Stand-alone Debugger/Utility)
BTUP	(Block Two Utility Package)
REX	(Real Time Executive)
SYSGEN	(System Generation)
SOV	(System Disc Driver Overlay)
BZUD	(Block Zero Utility Driver)

3.3.1 METHODS FOR LOADING DBUG

Two methods are given for this initial load process. The first method is for POINT 4 CPUs and the second method is for use with non-POINT 4 CPUs.

3.3.1.1 POINT 4 CPUs With Paper Tape MANIP PROMs

Press the APL button on the computer panel to bring MANIP into memory. Then place the DBUG object tape in the reader. At the master terminal enter

```
    Rx
```

where
x=0 or blank for the master Teletype (device code 10)
x=1 for the high-speed tape reader (device code 12)

(see also the appropriate POINT 4 CPU manual)

After the BTUP and DBUG tapes have been loaded into memory, transfer control from MANIP to DBUG by entering

```
    J73000
```

Record the ASM date and the punch date from the tape label on the Sysgen Log for future reference.

3.3.1.2 Non-POINT 4 CPUs - Special Tape Loader

A special bootstrap procedure is required to load the POINT 4 binary paper tape loader into memory.

This loader accepts tapes in standard absolute binary object format. Each record read from the tape is read into a buffer and checksummed before any data is stored in memory. This prevents data from being stored in the wrong place due to a possible read error on the address word.

The binary loader occupies the area between the bootstrap program and the top of memory. To use the loader, start execution at the top word of memory (location 77777 for 32KW) with switch zero off to select the master terminal's reader, or with the switch on to select the high-speed reader.

NOTE

> Switch zero is the left-most switch. The tape reader should stop at the last character on the tape. If it does not, the above sequence must be repeated.

The sysgen procedure requires 32KW (kilo words) of memory.

To use the special tape loader, put the paper tape labeled 'Binary' Loader' into the paper tape reader and follow the procedure shown in Figure 3-1.

SET DATA SWITCHES	THEN PRESS
77600	RESET EXAMINE
060510*	DEPOSIT
063610*	DEPOSIT NEXT
000777	DEPOSIT NEXT
001400	DEPOSIT NEXT
004774	DEPOSIT NEXT
004773	DEPOSIT NEXT
105305	DEPOSIT NEXT
000776	DEPOSIT NEXT
171000	DEPOSIT NEXT
004767	DEPOSIT NEXT
107300	DEPOSIT NEXT
045013	DEPOSIT NEXT
151400	DEPOSIT NEXT
004763	DEPOSIT NEXT
105300	DEPOSIT NEXT
010411	DEPOSIT NEXT
000771	DEPOSIT NEXT
77604	START

*CHANGE LAST DIGIT TO 2 FOR HIGH-SPEED READER (DEVICE CODE 12)

Figure 3-1. Paper Tape Loader Bootstrap Program

3.3.2 INITIALIZING MEMORY

Use DBUG's K command (see Table 2-2) to initialize memory. At the master terminal enter

```
    K0,72777,77377
```

3.3.3 LOADING BTUP, REX, SYSGEN, SOV, AND BZUD

Use DBUG's Rx command to load the remaining five components in succession. Place each tape in the reader in the following order:

1. BTUP
2. REX
3. SYSGEN
4. SOV
5. BZUD

At the master terminal enter

```
    Rx
```

where
    x=0 for the master terminal (device code 10)
    x=1 for the high-speed tape reader (device code 12)

Record ASM and punch dates from the tape labels on the Sysgen Log.

3.4 SYSTEM LOADER

The System Loader (SYSL), which resides in the Sysgen tape, builds LU/0 and loads the initial system files.

To start SYSL, enter

    J50000

3.4.1 REPORTING BAD BLOCKS

SYSL identifies itself and requests that "bad blocks" be reported with the display

```
IRIS R8.1 SYSTEM LOADER
BAD BLOCKS?
```

If the disc diagnostic/reliability tests discovered any bad blocks, enter the real disc address (RDA) of each unusable block, one at a time.

When all "bad blocks" have been reported or if there are none to report, press

    <RETURN>

3.4.2 LOADING DISCSUBS GROUP #1 TAPE

SYSL asks for the DISCSUBS Group #1 object tape to be loaded into the tape reader with the prompts

```
    PLACE DISCSUBS GROUP 1 OBJECT TAPE IN THE READER

    WHICH READER:  TTI (0) OR PTR (1)?
```

0 - for the master terminal (device code 10)
1 - for a high-speed paper tape reader (device code 12)

Enter the appropriate code (no <RETURN> is required). The DISCSUBS tape will be loaded onto the disc.

While SYSL builds LU/0 it displays

```
    PLEASE WAIT
```

3.4.3 LOADING INITIAL SYSTEM FILES

SYSL requests initial system file tapes (listed in Table 3-2). Place each tape in the reader in the sequence shown. SYSL prompts for each one.

Prompt	Enter
NAME?	{Filename}
TYPE?	{File Type}

Enter the filenames and types as they are shown in Table 3-2.

TABLE 3-2. GROUP #1 TAPES

Sequence	Name	Type
1	SCOPE	33400
2	BYE	33400
3	DSP	33400
4	FAULTPRINT	33400
5	CONFIG	77001
6	$MMUX[1]	77001
7	$DGMX[2]	77001
8	$RTC[3]	77001
9	$PTR[4]	36
10	$PTM[5]	36

NOTE

$Files are memory resident. For an efficient use of memory, enable only those files (by loading them with the $ sign) that are required for this installation.

[1] If a POINT 4 MUX is on the system, load as $MMUX. Otherwise load as MMUX.

[2] If a Data General 4060-type multiplexer is on the system, load as $DGMX. Otherwise load as DGMX.

[3] If a Data General 4060-type multiplexer is on the system, load as $RTC. Otherwise load as RTC.

[4] If a high-speed paper tape reader is on the system, load as $PTR. Otherwise load as PTR.

[5] If a Teletype paper tape reader/punch is on the system, load as $PTM. Otherwise load as PTM.

3.4.4 EXIT SYSL AND IPL

After the Group 1 tapes have been loaded, exit SYSL. At the prompt

```
    NAME?
```

enter

```
    <CTRL-C>
```

This starts the first IPL automatically.

It is necessary to transfer control to the System Initialization Routine (SIR). At the IPL prompt

```
    PRESS RETURN
```

enter

```
    2  (no <RETURN> is required)
```

For a description of IPL parameter options, see Section 2.6.

3.5 SYSTEM INITIALIZATION ROUTINE (SIR)

SIR IPLs into a minimum configuration (see Figure 3-2) and displays the following message on the terminal:

```
                     R8.1

A Licensed, unpublished, restricted and confidential work.
If and when this work is published, the following copyright
notice applies:

Copyright (c) 1982, POINT 4 DATA CORPORATION

MINIMUM CONFIGURATION REQUESTED, ONLY MASTER PORT ACTIVE!
```

3.5.1 DATE AND TIME

SIR requests date and time (time is based on the 24-hour clock):

```
ENTER YEAR,MONTH,DAY,HOUR,MINUTE
```

The date and time must be entered at this time using the format

```
YY,MM,DD,HH,MM
```

When the IPL is complete, SIR displays an exclamation mark (!).

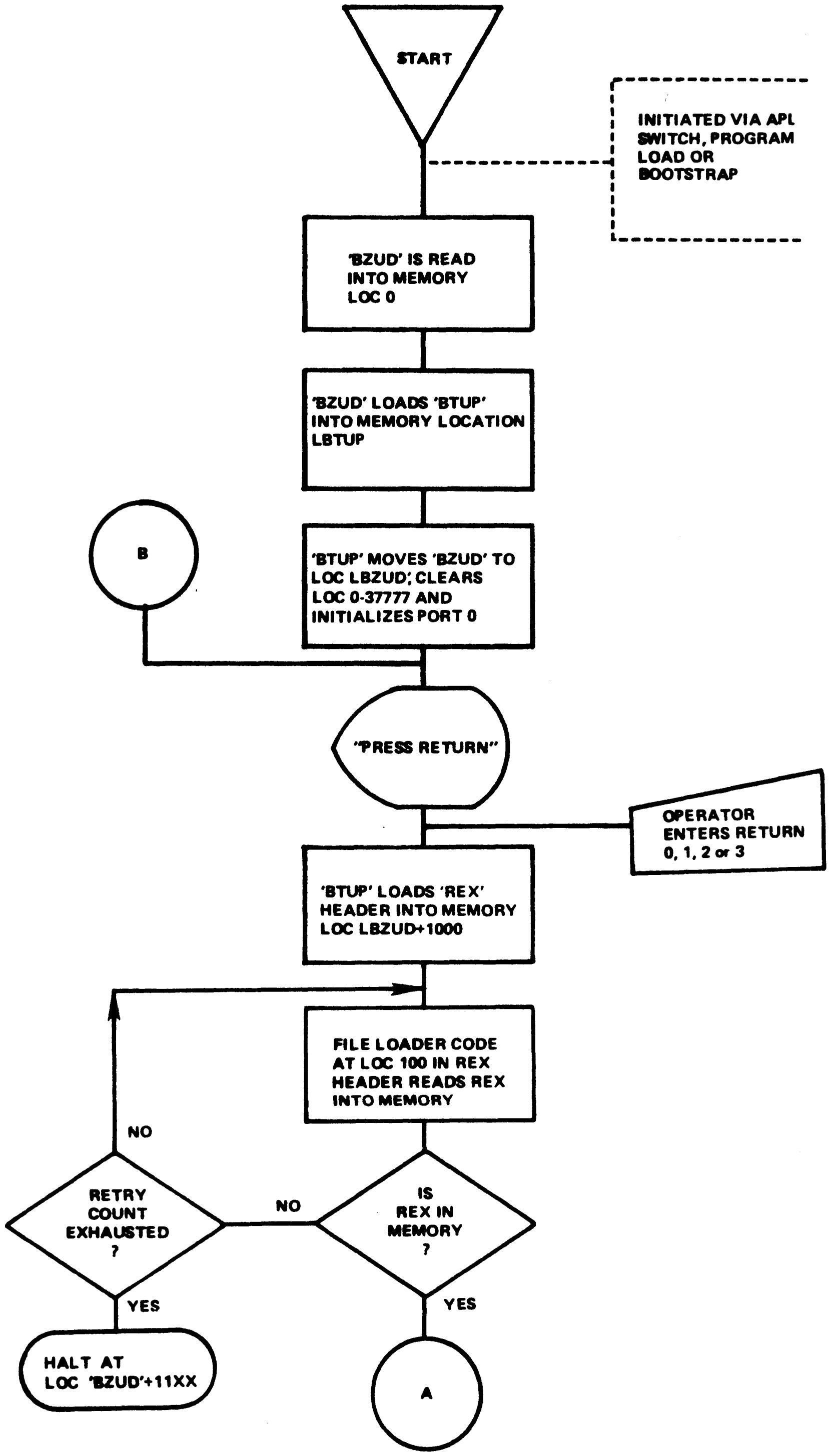

START
INITIATED VIA APL SWITCH, PROGRAM LOAD OR BOOTSTRAP
'BZUD' IS READ INTO MEMORY LOC 0
'BZUD' LOADS 'BTUP' INTO MEMORY LOCATION LBTUP
B
'BTUP' MOVES 'BZUD' TO LOC LBZUD; CLEARS LOC 0-37777 AND INITIALIZES PORT 0
"PRESS RETURN"
OPERATOR ENTERS RETURN 0, 1, 2 or 3
'BTUP' LOADS 'REX' HEADER INTO MEMORY LOC LBZUD+1000
FILE LOADER CODE AT LOC 100 IN REX HEADER READS REX INTO MEMORY
NO
RETRY COUNT EXHAUSTED ?
NO
IS REX IN MEMORY ?
YES
YES
HALT AT LOC 'BZUD'+11XX
A

Figure 3-2. IPL Flowchart (1 of 3)

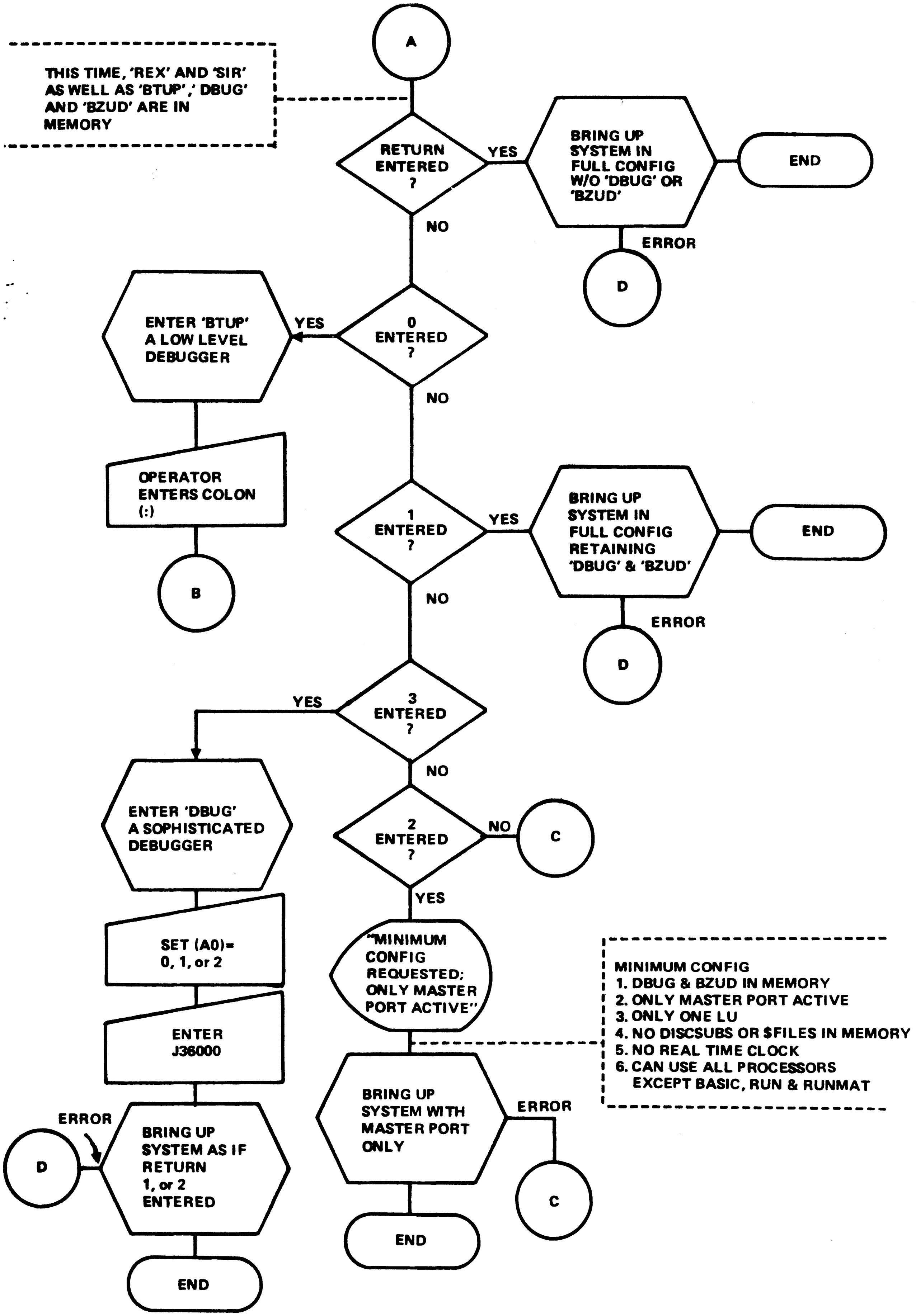

A
THIS TIME, 'REX' AND 'SIR' AS WELL AS 'BTUP',' DBUG' AND 'BZUD' ARE IN MEMORY
RETURN ENTERED ?
YES
BRING UP SYSTEM IN FULL CONFIG W/O 'DBUG' OR 'BZUD'
END
ERROR
D
NO
0 ENTERED ?
YES
ENTER 'BTUP' A LOW LEVEL DEBUGGER
OPERATOR ENTERS COLON (:)
B
NO
1 ENTERED ?
YES
BRING UP SYSTEM IN FULL CONFIG RETAINING 'DBUG' & 'BZUD'
END
ERROR
D
NO
3 ENTERED ?
YES
ENTER 'DBUG' A SOPHISTICATED DEBUGGER
SET (A0)= 0, 1, or 2
ENTER J36000
ERROR
D
BRING UP SYSTEM AS IF RETURN 1, or 2 ENTERED
END
NO
2 ENTERED ?
NO
C
YES
"MINIMUM CONFIG REQUESTED; ONLY MASTER PORT ACTIVE"
MINIMUM CONFIG
1. DBUG & BZUD IN MEMORY
2. ONLY MASTER PORT ACTIVE
3. ONLY ONE LU
4. NO DISCSUBS OR $FILES IN MEMORY
5. NO REAL TIME CLOCK
6. CAN USE ALL PROCESSORS EXCEPT BASIC, RUN & RUNMAT
BRING UP SYSTEM WITH MASTER PORT ONLY
ERROR
C
END

Figure 3-2. IPL Flowchart (2 of 3)

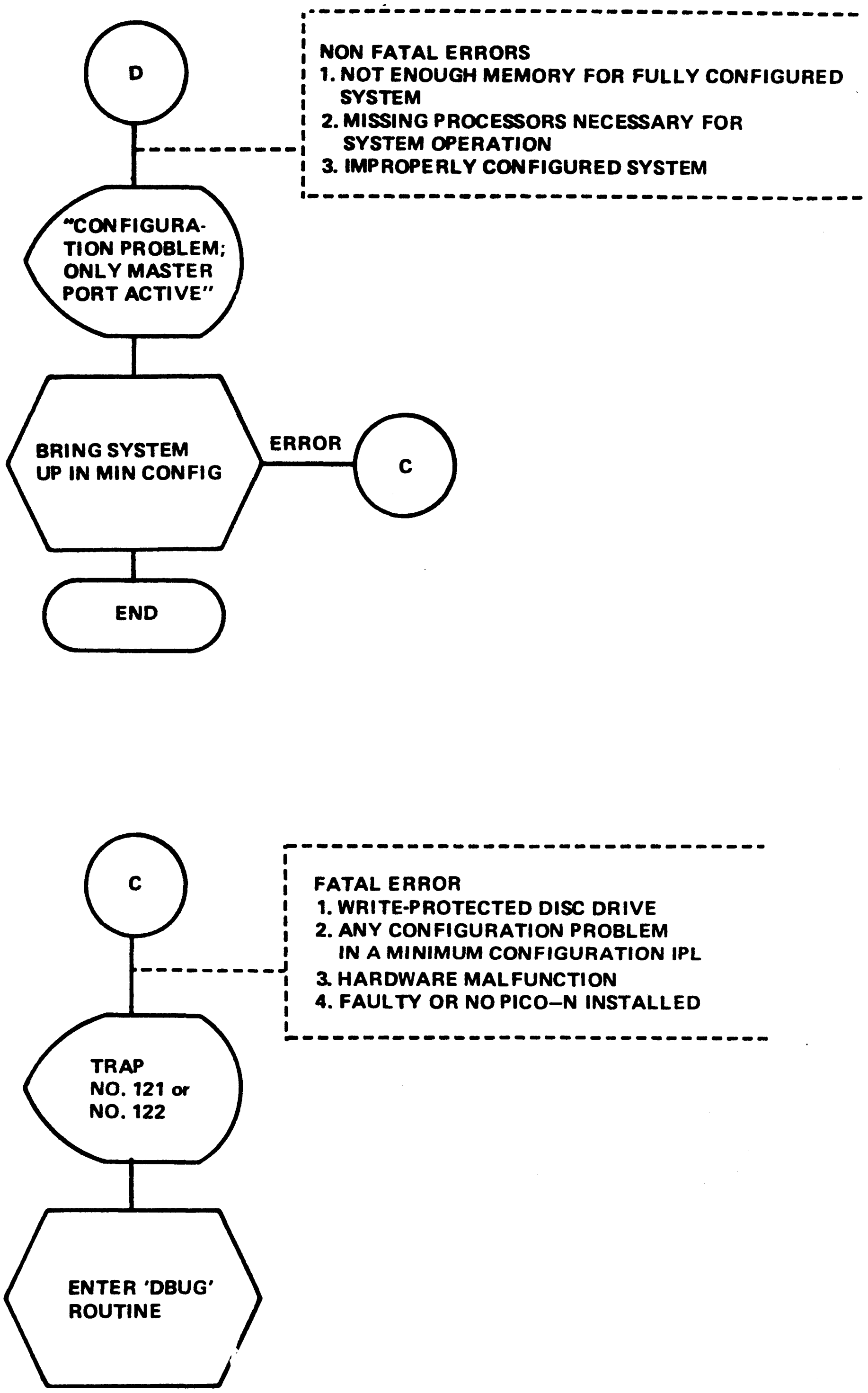

D
NON FATAL ERRORS
1. NOT ENOUGH MEMORY FOR FULLY CONFIGURED SYSTEM
2. MISSING PROCESSORS NECESSARY FOR SYSTEM OPERATION
3. IMPROPERLY CONFIGURED SYSTEM
"CONFIGURATION PROBLEM; ONLY MASTER PORT ACTIVE"
BRING SYSTEM UP IN MIN CONFIG
ERROR
C
END
C
FATAL ERROR
1. WRITE-PROTECTED DISC DRIVE
2. ANY CONFIGURATION PROBLEM IN A MINIMUM CONFIGURATION IPL
3. HARDWARE MALFUNCTION
4. FAULTY OR NO PICO–N INSTALLED
TRAP NO. 121 or NO. 122
ENTER 'DBUG' ROUTINE

Figure 3-2. IPL Flowchart (3 of 3)

3.5.2 LOG-ON/LOG-OFF PROCESSOR

BYE is the IRIS Log-on/Log-off processor (see Section 5.11.2.1). In its log-on function it requests the Account ID, which is not echoed, and displays information about the status of the account (time available, blocks available, etc.).

To invoke BYE, press

<ESC>

BYE displays

```
Welcome to "IRIS" R8.1 Timesharing
ACCOUNT ID?
```

Enter

MANAGER

The Account ID "MANAGER" must be entered in uppercase letters. It will not be echoed.

BYE activates the Manager account and places the system into command mode by displaying the system command prompt (#).

NOTE

> If there is no response, the probable cause is an incompatibility between the terminal's keyboard and the computer. To disable parity checking, enter
>
> <CTRL-P>
>
> A percent sign (%) is echoed, then press
>
> <ESC>
>
> Enter
>
> MANAGER
>
> The system prompt (#) is then displayed.

3.6 INITIAL SYSTEM CONFIGURATION

During this stage of the sysgen process the foundation is laid for the requirements of a particular system. The following subsections describe the parameters which must be set.

3.6.1 SYSTEM PARTITION SIZE

Set the System Partition Size in the General Information Table of the CONFIG file at location 400 (see Section 5.2.1).

PSIZ = System Partition Size

3.6.2 SYSTEM INFORMATION TABLE

Starting at location 600 in the System INFO Table, set the following parameters (see Section 5.2.2).

SPED = CPU Speed

MILU = Maximum Number of installed Logical Units

SPCF = Special Conditions Flag

TOPW = Highest Addressable Memory Address

ABUF = Auxiliary Buffer Size

3.6.3 MEMORY-RESIDENT DISCSUBS

The Memory-Resident DISCSUBS Table is preset to include those DISCSUBS that POINT 4 recommends should be memory resident. It is not necessary to redefine the table at this time. To modify this table, refer to Section 5.3.

3.6.4 DEFINING LOGICAL UNITS

Define the logical units for the system starting at location 1400 in the Disc Driver Table of the CONFIG file (see Section 5.4).

3.6.5 MULTIPLEXER PORTS

Multiplexer ports must be defined in the appropriate driver's Port Definition Table (see Section 5.8.1.4).

3.6.5.1 POINT 4 Multiplexer Ports

If a POINT 4 Multiplexer is on the system, define the required ports in the $MMUX Port Definition Table (see Section 5.8.3).

3.6.5.2 Data General 4060-Type Multiplexer Ports

If a Data General 4060-type multiplexer is on the system, define the required ports in the $DGMX Port Definition Table (see Section 5.8.4).

3.7 LOADING DISCSUBS

Before DISCSUBS can be loaded, a full configuration IPL is required. On a POINT 4 MARK 5 or MARK 8 Processor Mini-panel, press STOP and then at the master terminal enter

Pxx

where
xx = disc device code (see IRIS R8 Peripherals Handbook).

Respond to the IPL sequence prompt PRESS RETURN by entering

1

(<RETURN> is not required)

If the system has a POINT 4 Multiplexer, a full configuration IPL sets the real-time clock and changes the master port to Port #1.

If the CPU is not a POINT 4 MARK Series computer, then perform a standard IPL.

WARNING

> DO NOT CONTINUE if the IPL is not successful. It must complete without any traps and the CARRY indicator should be flashing at a 1Hz (once per second) rate which indicates that the system is configured correctly and the real-time clock (RTC or MMUX's RTC) is functioning.
>
> If the CARRY light is not flashing correctly, there is a configuration problem which must be rectified before continuing the sysgen.
>
> If the problem is in the hardware (i.e., a device, not the system), the sysgen may be resumed by doing another IPL after the problem is fixed.
>
> If it is a software problem, the sysgen must be repeated from the beginning.

DISCSUBS may be loaded either through a paper tape reader or the master terminal reader.

3.7.1 PAPER TAPE READER (DEVICE CODE 12)

1. Press <ESC>.

2. Log on to the MANAGER account.

3. At the system command prompt (#), enter

   DSP <CTRL-E>{key}<CTRL-E>

   where key is the password assigned by the system manager (the default is X).

4. Use the F command to select the DISCSUBS file by entering

   FDISCSUBS

5. Place the tapes (DISCSUBS #2, #3, #4, #5, and #6) in the reader.

6. Use the R command to load each Discsubs tape:

   R (for DISCSUBS #2)

   R (for DISCSUBS #3)

   R (for DISCSUBS #4)

   R (for DISCSUBS #5)

   R (for DISCSUBS #6)

NOTE

Be careful not to enter R0 or R1.

7. To exit DSP and return to the system command prompt (#), enter

   X

8. Log off by entering

   BYE

3.7.2 MASTER TERMINAL READER (DEVICE CODE 10)

If tapes are loaded through the master terminal's reader, a null code is sent to the master terminal after each tape record (about 4 inches) is checksummed. This causes the master terminal to cycle (printing nothing, but a 'clunk' is heard each time) indicating that the tape is being read without errors.

1. Press <ESC>.

2. Log on as MANAGER on a port other than the master terminal.

3. Enter

   DSP <CTRL-E>{key}<CTRL-E>

4. Load tapes onto the terminal's reader as shown in Section 3.7.1.

NOTE

> The R command may not be given from the master terminal itself, and the Teletype must have the modification that causes the tape to be fed one frame for each NIOS TTI machine code instruction.

5. Exit DSP as follows

   X

6. Log off by entering

   BYE

NOTE

> If an error occurs, reload that tape from the beginning.

3.8 LOADING DRIVERS AND PROCESSORS

It is now necessary to bring the system up into another minimum configuration IPL.

1. IPL as before (i.e., press the STOP and APL buttons).

2. Enter

   Pxx

   where xx = disc device code (see IRIS R8 Peripherals Handbook).

3. At the PRESS RETURN prompt, enter

   2

   (<RETURN> is not required).

4. Log on to the MANAGER account.

5. Use PLOAD to load the driver and processor tapes in the sequence given in Table 3-3. Enter

   PLOAD

   PLOAD will prompt for READER?, NAME?, and TYPE?. Please see Section 2.4 for a full description.

6. To exit PLOAD and return to system command mode, press

   <CTRL-C>

TABLE 3-3. DRIVERS AND PROCESSORS

Seq	Name	Type	Seq	Name	Type
1	$PHA[1]	77001	22	COPY	33401
2	$TTY[1]	77001	23	FORMAT	33401
3	$PTP[1]	36	24	INSTALL	33401
4	$CTUS[1]	77001	25	KILL	33401
5	$DEC[1]	77001	26	MAIL	33401
6	LPTM[2]	36	27	MESSAGES	77001
7	LPTP[2]	36	28	MK8[1]	77001
8	LPTP51[2]	36	29	PORT	33401
9	LPTD[2]	36	30	QUERY	33401
10	$MTA0[3]	36	31	REHASH	77401
11	$MTAS[3]	77001	32	REMOVE	33401
12	$TERMS	77001	33	SAVE	33401
13	TERM.name[4]	77001	34	PROTECT	33401
14	$LCM	77001	35	VERIFY	33401
15	$FOREIGN	36	36	SHUTDOWN	33403
16	BCONVERT	33401	37	LIBR	33401
17	LCMACTIVATE	33401	38	ASSEMBLE[5]	33401
18	LCMDEACTIVATE	33401	41	EDIT	33401
19	CHANGE	33401	40	RUNMAT	33402
20	CLEANUP	77401	41	BASIC	33702
21	CLEANUPX	77401	36	RUN	33602

[1] Load the driver <u>with</u> a $-sign for the following:

$PHA - phantom port
$TTY - secondary Teletype
$PTP - paper tape punch
$CTUS - cassette tape unit
$DEC - POINT 4 MARK 3 or 5 Computer
$MTAS - magnetic tape unit
$LCM - POINT 4 LOTUS CACHE MEMORY
$FOREIGN - read/write foreign diskette
$MK8 - POINT 4 MARK 8 Computer

Otherwise load the driver without the $-sign.

[2] Load the driver <u>without</u> a $-sign:

LPTM - Line printer on a POINT 4 310 or MARK 3 Mux
LPTP - Line printer on a parallel interface, device code 17
LPTP51 - Line printer on a parallel interface, device code 51 (included only on request)
LPTD - Line printer on a Data General 4060-type Mux

Select driver appropriate for the system and reload as $LPT (see Section 5.8.7).

[3] Load as $MTA0 if system has a magnetic tape or cassette tape unit (see also Section 5.8.10).

[4] Load all TERM.name tapes provided. See Section 5.9 for information on enabling specific Terminal Translation Modules.

[5] Filename ASM is optional.

3.9 LOADING UTILITIES AND TEXT FILES

Return to the system command prompt (#) by pressing <CTRL-C>, then shutdown the system and IPL into a full configuration as follows:

1. At the system prompt (#), enter

   ```
   SHUTDOWN <CTRL-E>{key}<CTRL-E>
   ```

   where key is the password assigned by the system manager (the default is X).

   NOTE

   > The SHUTDOWN command shuts down the system in an orderly fashion (e.g., it flushes the buffer pool). To preserve the integrity of the system, use SHUTDOWN. DO NOT PRESS THE STOP BUTTON!

2. Respond to the PRESS RETURN prompt by entering

   ```
   1
   ```

3. Log on to the UTILITY account.

4. At the system command prompt, enter

   ```
   BASIC
   SIZE
   EXIT
   ```

   As a result, BASIC is linked with RUN and RUNMAT. This also displays the maximum BASIC program size.

3.9.1 LOADING BASIC PROGRAMS

The procedure for loading text files of BASIC programs is as follows:

1. Place the paper tape in the reader.

2. At the system command prompt (#), enter

   ```
   BASIC
   NEW
   LOAD $PTR (use $PTM for Teletype)
   EXIT
   ```

   This reads the BASIC program into the port's active file and then exits to command mode.

3. To save the BASIC program on disc, enter

   ```
   SAVE <pp> {filename}
   ```

   where pp is the protection level given for the program

   Table 3-4 gives a list of BASIC program filenames and protection levels. These programs must be loaded with the name shown in the table. BASIC programs may be added as they become available. Refer to the Sysgen Log for a complete list.

TABLE 3-4. BASIC UTILITY PROGRAMS

Seq	PP	Name	Seq	PP	Name
1	77	ACTUTIL.1	12	77	GUIDE.LPT
2	77	ACCOUNTUTILITY	13	77	GUIDE.LU
3	77	BASICTEST[1]	14	77	GUIDE.BLOCKCPY
4	33	BUILDXF	15	77	RETRY
5	77	BUILDPFERR[2]	16	33	SETTIME
6	33	BUILDPF	17	77	R7TO8ACTCONV
7	33	QUERYPF	18	33	U.CONVERT
8	33	KILLPF	19	33	U.CONVERT1
9	33	ANALYPF	20	77	CHECKSUM
10	77	EXTRAPORT	21	33	TRANSMIT
11	77	GUIDE	22	33	RECEIVE

[1] BASICTEST is a BASIC Readiness Test. To run BASICTEST it must be copied to LU/1. When LU/1 is installed and serviceable, log on to the UTILITY account. At the system command prompt (#), enter

```
    BASIC BASICTEST
    DUMP 1/BT.
    EXIT
```

BASICTEST is now available for use.

[2] BUILDPFERR is a BASIC program loaded as source text. BUILDPFERR builds the POLYFILERRORS file which is used by the BUILDPF, KILLPF, and QUERYPF programs. It is created once only on LU/0. If BUILDPF, etc. cannot find POLYFILERRORS, error codes for CALL 91 and SEARCH statements are printed. Refer to the IRIS Business BASIC Manual for error code definitions.

To create the POLYFILERRORS file, at the system command prompt (#), enter

```
    BUILDPFERR
```

BUILDPFERR uses approximately eight disc blocks. The program may be deleted after the POLYFILERRORS file has been created.

3.9.2 LOADING STAND-ALONE UTILITIES

The procedure for loading stand-alone utility files differs from that of the text files. The format is

```
COPY {filename} *A=$PTR
```

(Use $PTM for a Teletype)

Stand-alone utility files include

DISCDIAG*
DISCFORMAT*
DDCOPY or DISCUTILITY
BLOCKCOPY
LCMDIAG1.3
M8EXERCISER
DC700**
MUX310DP

*These files must be supplied by the disc controller vendor.

**DC700 is the diagnostic program for the POINT 4 LOTUS 700 Disc Controller. To load DC700, enter

```
COPY DC700*A<UP-ARROW>$PTR,<UP-ARROW>$PTR
```

Other stand-alone utilities will be added as they are developed.

3.9.3 LOADING TEXT FILES

The text files PZ, SYMBOLS, L.LCMDEFS and DEFS should be loaded at this time.

1. PZ, SYMBOLS and L.LCMDEFS are loaded with the format

   ```
   COPY {filename} *T=$PTR
   ```

   (Use $PTM for a Teletype)

2. The DEFS file is supplied on two tapes, DEFS #1 of 2, and DEFS #2 of 2. These are loaded with the format

   ```
   COPY DEFS *T=<up-arrow>$PTR,<up-arrow>$PTR
   ```

   (Use $PTM for a Teletype)

3.9.4 FORMAT USERID FILE

Use FORMAT to create the USERID file. At the system command prompt, enter

```
#FORMAT USERID
D2,S14
<RETURN>
```

3.10 FINAL CONFIGURATION

The initial system configuration is described in Section 3.6. The final configuration makes the system unique to the requirements of a particular installation. The following is a checklist for those areas of configuration which are largely optional.

Configuration Area	Section
Log-On Restrictions	5.5
Automatic Program Start	5.6
User Accounts	5.7
Phantom Ports	5.8.6
Line Printer(s)	5.8.7
Terminal Translator	5.9
Processor Passwords (Options and restrictions)	5.11.1

3.11 SYSTEM DISC BACKUP

Once the sysgen procedure has been completed, it is absolutely essential that a master backup copy of the system disc be made. This backup copy should be dated and kept in a safe place, and should never be used to IPL the system.

3.11.1 BACKUP PREPARATION

- Mount a fresh disc pack on the destination drive
- Log on to the MANAGER account

3.11.2 BACKUP WITH A POINT 4 LOTUS 700 DISC CONTROLLER

Shutdown to DISCUTILITY with the command

```
    SHUTDOWN <CTRL-E>{key}<CTRL-E> DISCUTILITY
```

where key is the password assigned by the system manager (the default is X).

At the computer front panel press

```
    RESET/START at location 2
```

The DISCUTILITY program guides the user through the necessary backup steps.

3.11.3 BACKUP WITH OTHER DISC CONTROLLERS

Shutdown to DDCOPY with the command

```
    SHUTDOWN <CTRL-E>{key}<CTRL-E> DDCOPY
```

DDCOPY loads into location 400 (octal).

Obtain the source and destination constants for the particular disc drives from the DDCOPY listing.

Use the computer front panel to enter the source constant at location 401 and the destination constant at location 402.

To start the copy process, press

```
    RESET/START at location 400
```

A normal HALT (63077 in Data Lights) indicates a good completion.

For more information on backups, refer to the IRIS Operations Manual.

Section 4
CTUTILITY
Sysgen and Transfer of Nonzero LUs

CTUTILITY may be used to install a completely new IRIS system or to transfer any nonzero logical unit between two drives.

When used for a sysgen, CTUTILITY is a disc image copy procedure that requires minimal manual intervention. The IRIS system generated by this process is a standard operating system. To customize this system for a particular installation, please refer to Section 5.

CTUTILITY may be used to transfer any nonzero logical unit between the same or different drives, disc controllers, and MARK 3, MARK 5, and MARK 8 systems.

4.1 INTRODUCTION TO CTUTILITY

CTUTILITY boots into memory from a special CTUTILITY cassette and runs on MARK 3 and MARK 5/8 Computer Systems. It provides a method for transfers from MARK 3 to MARK 5/8 (or vice versa) because it can transfer nonzero logical units between different drive-controller configurations supported under IRIS.

Any CPU which runs IRIS may be used. A POINT 4 310 or MARK 3 Mux is required. A CRT must be connected to Mux Port 0 and a CTU drive to Mux Port 1. Most standard CRT cables may be used.

NOTE

> When the power is turned on for the CTU drive, the read heads should load as though reading tape and then unload (retract) with a noticeable sound. If it does not, the CTU drive does not have the most up-to-date components. Please call POINT 4 hardware customer support to request an update. The older components do not support those features of the new software that enhance reliability.

The CTUTILITY tape-set can be used on any disc drive and controller supported under IRIS. Therefore, the tapes for LU/0 are desi̇gnated as a universal template.

The following subsections describe CTUTILITY loading instructions, sysgen procedure (i.e., tape-to-disc transfer), transfer of nonzero LUs (i.e., disc-to-tape transfers), and loading procedures for stand-alone and utility programs.

NOTES

- CTUTILITY replaces the MULTIBLK cassette tape utility. MULTIBLK tapes cannot be used with CTUTILITY and are no longer necessary.
- CTUTILITY may not be used on muxes with more than 32 (decimal) ports.

4.2 USING CASSETTE TAPES

Proper cassette mounting and removal are important for accurate data transfer. Use the following procedures for cassette handling.

4.2.1 MOUNTING CASSETTE

To mount the cassette into the "cassette well" of the CTU drive, use the following procedure.

1. Grasp sides of cassette, A-side up, with thumb and third finger (see Figure 4-1).

2. Gently slide the back edge of the cassette up to the lower edge of the two stainless steel positioning springs in the back of the cassette well.

3. Holding the sides of the cassette with thumb and third finger, press the front edge of the cassette down with the first finger. A click should be heard when the cassette is engaged. If resistance is encountered, do not use force. Check the alignment between the high spots on the PHI-deck black plastic drive hubs and the drive studs on the cassette tape reel holes. If necessary, adjust alignment by moving the hubs slightly by hand.

4. If the cassette is properly installed, its front edge will be level with its back edge (horizontal).

4.2.2 REMOVING CASSETTE

1. Press the black plastic lever in the right front corner of the cassette well firmly to the right.

2. Lift out by the left-hand corner of the cassette. Do not touch the tape! Touching the tape may eventually cause read errors due to tape and head contamination.

IMPORTANT!

> To avoid damaging or stretching the tape, allow the cassette head to disengage from the tape before removing the cassette.

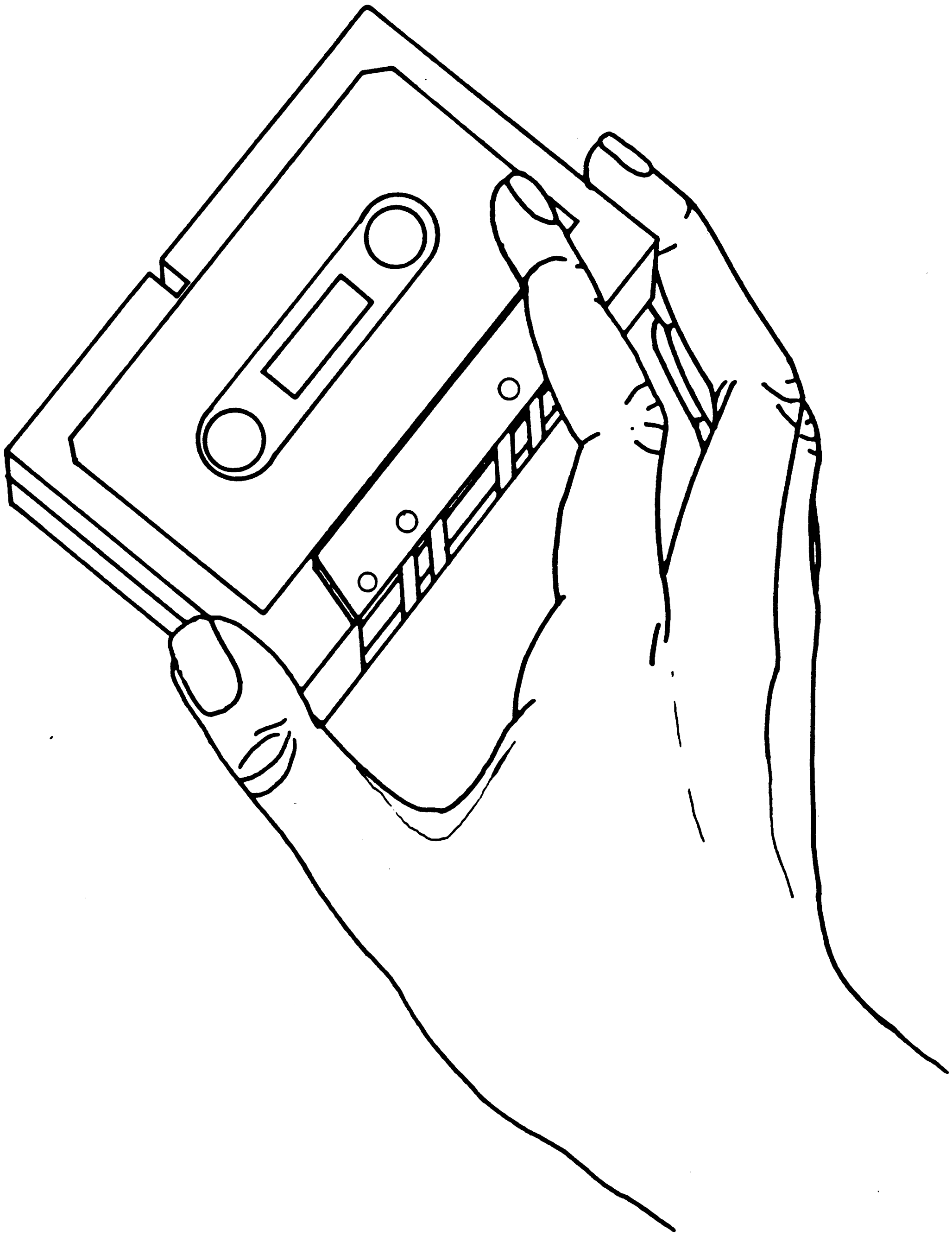


Figure 4-1. Mounting the Cassette

4.2.3 CTU CLEANING

The tape transport of the CTU may accumulate oxide from the recording media and dust from the environment, which may cause errors. If a lot of "soft errors" are occurring, the following procedure is recommended for cleaning the tape transport.

1. Remove the tape cassette.
2. Clean the tape guides, pinch roller, capstan and read/write head with a cotton swab moistened with a mixture of 70% isopropyl alcohol (rubbing alcohol) and water. A cotton pad presaturated with this mixture such as Texwipe "Alcopad" is excellent for this.
3. After all of the oxide and dust have been removed, let the alcohol mixture evaporate from the tape transport. Do not wipe dry or touch with fingers.

4.3 CTUTILITY SYSGEN PREPARATIONS

Before beginning the sysgen procedure make sure that all needed materials (tapes, disc packs, etc.) are at hand, that the requirements of the particular system to be installed have been specified (see Section 3.1.2), and that hardware diagnostics have been run as discussed in Section 3.2.

The following are required:

- Diagnostic programs
- IRIS Cassette tapes appropriate for the particular computer
- One disc pack to set aside as a master copy
- Several disc packs for backup
- Pico-N
- CPU memory (must be 32K or more)

4.3.1 CTUTILITY SYSGEN PROCEDURE

The template for LU/0 contains a disc image copy of a standard IRIS Operating System. The system is loaded as follows

1. Determine the type of PROMs that are on the CPU (see Sections 4.4.1 and 4.4.2).
2. Load CTUTILITY by the method appropriate for the type of CPU and PROMs:
    - MARK 5/8 CPU with CTU PROMs - Section 4.5.2
    - MARK 3 CPU with CTU PROMs - Section 4.5.3
    - MARK 3 CPU without CTU PROMs - Section 4.5.4
    - All other CPUs without CTU PROMs - Section 4.5.5
3. Perform a sysgen as described in Section 4.6.1.
4. Load the appropriate utility programs as described in Section 4.8.
5. Back up the system disc (LU/0).

4.3.2 CUSTOMIZING THE IRIS OPERATING SYSTEM

The standard IRIS Operating System may be configured to suit the requirements of a particular installation. Refer to Section 5 for configuration procedures.

4.4 PREPARATION FOR LOADING CTUTILITY

POINT 4 CPUs are equipped with different types of PROMs. CTUTILITY loading procedures differ according to the PROMs in use.

The following tests may be used to identify the PROMs on POINT 4 CPUs.

4.4.1 CHECKING POINT 4 MARK 5/8 CPU PROMs

The POINT 4 MARK 5 or MARK 8 CPU may have either CTU PROMs or PTP PROMs. To identify which type the CPU has, use the following procedure:

1. Press STOP, then APL on the computer front panel.

2. Connect the CTU drive to the second port on the Mux.

3. Insert cassette tape into the CTU drive, and power on the CTU.

4. Enter <CTRL-Z><RETURN>

   - If the system displays

     ```
     ^Z
     ```

     the CPU has CTU PROMs. Follow the loading procedure in Section 4.5.2.

   - If the system displays

     ```
     \
     ```

     the CPU has PTP PROMs. Follow the loading procedure in Section 4.5.5.

4.4.2 CHECKING POINT 4 MARK 3 CPU PROMs

The POINT 4 MARK 3 CPU may have either CTU PROMs or Archive PROMs. To identify which type the CPU has, use the following procedure:

1. Press RESET on the computer front panel.
2. Connect the CTU drive to the second port on the Mux.
3. Insert cassette tape into the CTU drive, and power on the CTU.
4. Enter R

   - If the system begins reading the tape, the CPU has CTU PROMs. Follow the loading procedure in Section 4.5.3.
   - If the system displays

     ```
     \
     ```

     the CPU has Archive PROMs. Follow the loading procedure in Section 4.5.4.

4.5 CTUTILITY LOADING PROCEDURES

The five procedures given in this section load the CTUTILITY memory-resident executive into memory from the CTUTILITY cassette. Each procedure applies to a different configuration:

1. Any CPU currently running IRIS

2. POINT 4 MARK 5 or MARK 8 CPU with CTU PROMs

3. POINT 4 MARK 3 CPU with CTU PROMs

4. POINT 4 MARK 3 CPU without CTU PROMs

5. Any CPU without CTU PROMs

User input is underlined. All entries are followed by <RETURN>. The <RETURN> is not shown unless it is the only response, or more than one is required.

4.5.1 LOADING CTUTILITY ON CPUs CURRENTLY RUNNING IRIS

For an IRIS R8.1 (or later) system, use the following procedure to load CTUTILITY.

1. Mount the CTUTILITY cassette into the CTU drive, following the instructions given in Section 4.2.1.

2. Enter

   ```
   SHUTDOWN <CTRL-E>{key}<CTRL-E> @73000
   ```

3. Start CPU execution at location 73000. On a POINT 4 MARK 5 CPU press APL and enter

   ```
   J73000
   ```

   On other CPUs, it may be necessary to use the front panel switches.

4. Rewind the tape by entering

   ```
   <CTRL-Z><RETURN>
   ```

   A caret and a Z should be displayed on the screen.

5. Select Track 0 by entering

   ```
   <CTRL-T>0
   ```

6. Enter

    0:

   The system will display the current contents of location 0. This is not significant to this procedure.

7. Enter

    <CTRL-R>20,156

   where 20 is the decimal value of the beginning block on tape, and 156 is the number of cassette tape blocks (minus 1) to be read.

   It takes approximately two minutes for the tape to be read. It is finished when the tape stops and the cursor moves to the following line. If the cursor does not move down to the next line, an error has occurred. Remove the cassette and go back to step 1.

8. To begin CTUTILITY, enter

    J2

9. Now follow the instructions for either the disc-to-tape or tape-to-disc transfer, as appropriate. These transfer procedures are described in Section 4.6.

4.5.2 LOADING CTUTILITY ON POINT 4 MARK 5/8 CPUs WITH CTU PROMs

1. Mount the CTUTILITY cassette into the CTU drive, following the instructions given in Section 4.2.1.

2. Press STOP then APL on the computer front panel.

3. Rewind the tape by entering

   <CTRL-Z><RETURN>

   A caret and a Z should be displayed on the screen. (The system displays a backslash if the CPU does not have CTU PROMs. In that case, follow the procedure in Section 4.5.5.) Wait for the cursor to move down to the next line.

4. Select Track 0 by entering

   <CTRL-T>0

5. Enter

   0:

   The system will display the current contents of location 0. This is not significant to this procedure.

6. Enter

   <CTRL-R>20,156

   where 20 is the decimal value of the beginning block on tape, and 156 is the number of cassette tape blocks (minus 1) to be read.

   It takes approximately two minutes for the tape to be read. It is finished when the tape stops and the cursor moves to the following line. If the cursor does not move down to the next line, an error has occurred. Remove the cassette and go back to step 1.

7. To begin CTUTILITY, enter

   J2

8. Now follow the instructions for either the disc-to-tape or tape-to-disc transfer, as appropriate. These transfer procedures are described in Section 4.6.

4.5.3 LOADING CTUTILITY ON POINT 4 MARK 3 CPUs WITH CTU PROMs

1. Insert the CTUTILITY tape into the drive, following the instructions given in Section 4.2.1.

2. Press the RESET button on the computer front panel.

3. To read in the MARK 3 boot program, enter

   R

   The system displays a backslash if the CPU does not have CTU PROMs. In that case, follow the procedure in Section 4.5.4.

4. Rewind the tape by entering

   <CTRL-Z><RETURN>

   A caret and a Z should be displayed on the screen. Wait for the cursor to move down to the next line.

5. Select Track 0 by entering

   <CTRL-T>0

6. Enter

   0:

   The system will display the current contents of location 0. This is not significant to this procedure.

7. Enter

   <CTRL-R>20,156

   where 20 is the decimal value of the beginning block on tape, and 156 is the number of cassette tape blocks (minus 1) to be read.

   It takes approximately two minutes for the tape to be read. It is finished when the tape stops and the cursor moves to the following line. If the cursor does not move down to the next line, an error has occurred. Remove the cassette and go back to step 1.

8. To begin CTUTILITY, enter

   J2

9. Now follow the instructions for either the disc-to-tape or tape-to-disc transfer, as appropriate. These transfer procedures are described in Section 4.6.

4.5.4 LOADING CTUTILITY ON POINT 4 MARK 3 CPUs WITHOUT CTU PROMs

For this procedure, you will need the streamer tape cartridge containing DISCUTILITY and DBUG.

1. Insert the tape cartridge in the streamer tape drive.

2. Press the RESET button on the computer front panel. This brings the MANIP program into memory.

3. To load the streamer tape, enter

   H

   If a program other than DBUG starts running, press RESET and enter the address of CTU DBUG

   J64000

4. Mount the CTUTILITY cassette into the CTU drive, following the instructions given in Section 4.2.1.

5. Rewind the tape by entering

   <CTRL-Z><RETURN>

   A caret and a Z should be displayed on the screen. Wait for the cursor to move down to the next line.

6. Sélect Track 0 by entering

   <CTRL-T>0

7. Enter

   0:

   The system will display the current contents of location 0. This is not significant to this procedure.

8. Enter

   <CTRL-R>20,156

   where 20 is the decimal value of the beginning block on tape, and 156 is the number of cassette tape blocks (minus 1) to be read.

   It takes aproximately two minutes for the tape to be read. It is finished when the tape stops and the cursor moves to the following line. If the cursor does not move down to the next line, an error has occurred. Remove the cassette and go back to step 4.

9. To begin CTUTILITY, enter

   J2

10. Now follow the instructions for either the disc-to-tape or tape-to-disc transfer, as appropriate. These transfer procedures are described in Section 4.6.

4.5.5 LOADING CTUTILITY ON CPUs WITHOUT CTU PROMs

This procedure applies to non-POINT 4 CPUs and POINT 4 MARK 5 or MARK 8 CPUs without CTU PROMs. The procedure involves entering octal values into memory. On a POINT 4 CPU, this is done by using the ":" command. On other computers, it may be necessary to use the front panel switches to EXAMine the address, and DEPosit or STORE the subsequent values into memory.

The CPU must allow execution to be started at a specified address. On a POINT 4 CPU, this is done by using the "J" command. On other computers, it may be necessary to use the front panel switches and the START switch.

1. Mount the CTUTILITY tape into the drive, following the instructions given in Section 4.2.1.

2. Starting at location 66040, enter the following into memory:

Location	Contents
66040	0
41	50377
42	20000
43	0
44	157765
45	154151
46	161576
47	154160
66050	20414
51	61025
52	63025
53	20766
54	101113
55	776
56	20764
57	101120
66060	40760
61	20761
62	40757
63	400
64	66000
65	111260
66	130662
67	126263
66070	106615

3. Start CPU execution at location 66050 to read in MARK 5 boot program from the CTUTILITY tape.

4. When the tape stops, start CPU execution at location 70000.

5. Rewind the tape by entering

   <CTRL-Z><RETURN>

   A caret and a Z should be displayed on the screen. Wait for the cursor to move down to the next line.

6. Select Track 0 by entering

   <CTRL-T>0

7. Enter

   0:

   The system will display the current contents of memory location 0. This value is not significant to this procedure.

8. Enter

   <CTRL-R>20,156

   where 20 is the decimal value of the beginning block on tape, and 156 is the number of cassette tape blocks (minus 1) to be read.

   It takes aproximately two minutes for the tape to be read. It is finished when the tape stops and the cursor moves to the following line. If the cursor does not move down to the next line, an error has occurred. Remove the tape and go back to step 1.

9. To begin CTUTILITY, enter

   J2

10. Now follow the instructions for either the disc-to-tape or tape-to-disc transfer, as appropriate. These transfer procedures are described in Section 4.6.

4.6 TRANSFER PROCEDURES

Logical unit zero may be transferred via tape only to equivalent configurations. For a generalized LU/0 transfer, use the LU/0 template supplied by POINT 4 and follow the tape-to-disc transfer procedure described in Section 4.6.1.

NOTE

> The entire logical unit on the disc must be scratch because it will be overwritten.

To transfer a nonzero logical unit, first use the disc-to-tape transfer procedure in Section 4.6.2 on the source system. This copies the logical unit onto cassette tape. Then follow the tape-to-disc transfer procedure described in Section 4.6.1 to copy the tape to the destination system.

4.6.1 TAPE-TO-DISC

The tape-to-disc copy procedure may be used for the following purposes:

- With an LU/0 template, to sysgen IRIS on any supported drive and controller.
- To install a nonzero LU by moving a CTUTILITY-created LU from tape to disc.

The dialogue in this section is an example illustrating a sysgen of LU/0 from tape to disc. Some values shown here are dependent on the configuration. Values may vary.

NOTE

> This procedure requires a formatted disc pack. If the disc format program is included on the Stand-Alone Utilities cassette, use the procedure in Section 4.8. If the disc format program is not included on the cassette, an already-formatted disc pack must be available before sysgening from CTU.

When CTUTILITY is loaded and executing, the system displays:

```
INITIALIZING CTUTILITY.....

CTUTILITY IS LOADED.

THIS REVISION OF CTUTILITY USES MARK 3 DISC ENTRY NUMBERS
301 AND GREATER.  IF YOUR MARK 3 SPEC SHEETS START AT 1,
THEN ADD 300 TO THEM TO USE THIS REVISION.

PLEASE LOCATE YOUR PARTICULAR DISC CONTROLLER, DEVICE CODE
AND DRIVE IN THE R8.1 PERIPHERALS HANDBOOK. (NOTE: IF NOT
FOUND THEN IT IS NOT SUPPORTED BY POINT 4.)

PLEASE ENTER THE DISC SPECIFICATION ENTRY NUMBER
FOR THE SYSTEM YOU ARE "CURRENTLY" ON:
```

Locate the appropriate disc controller, device code and drive in the IRIS R8 Peripherals Handbook. The entry number is printed in the upper right corner of the page, as shown in Figure 4-2. Enter this number and press <RETURN>.

The system displays

```
READING SOV AND BZUD, PLEASE WAIT.....
```

while it reads SOV (System Overlay Disc Driver) and BZUD (Block Zero Utility Driver). When completed, the system displays

```
USING THE DISC SPECIFICATION SHEET, CALCULATE PHYU AND FCYL
FOR THE LOGICAL UNIT ON DISC TO BE USED FOR CTU TRANSFERS
(EITHER AS SOURCE OR DESTINATION).

PHYU=
```

The formula for calculating PHYU (Physical Unit Select Word) is given on the appropriate Disc Specification sheet in the IRIS R8 Peripherals Handbook.

After the value of PHYU is entered, the system requests

```
FCYL=
```

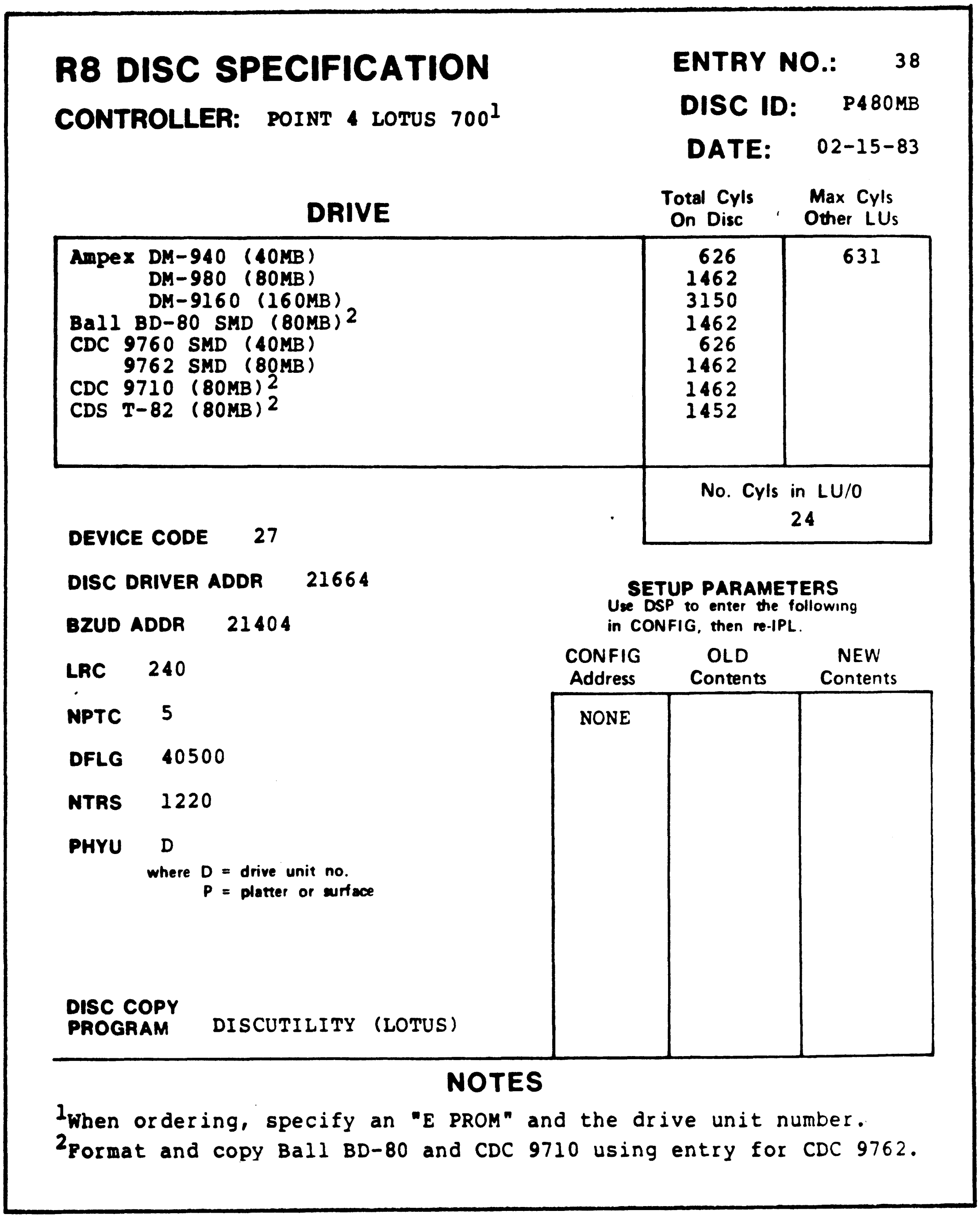
After FCYL (first physical cylinder) is entered, the system asks

```
IS THIS A TRANSFER FROM TAPE TO DISC? (Y/N)
```

Enter Y for yes. The system displays

```
REMOVE THE CURRENT TAPE AND LOAD TAPE NUMBER 001 OF THE SET.

PRESS CR WHEN READY!
```



R8 DISC SPECIFICATION

ENTRY NO.: 38

CONTROLLER: POINT 4 LOTUS 700[1]

DISC ID: P480MB

DATE: 02-15-83

DRIVE	Total Cyls On Disc	Max Cyls Other LUs
Ampex DM-940 (40MB)	626	631
DM-980 (80MB)	1462	
DM-9160 (160MB)	3150	
Ball BD-80 SMD (80MB)[2]	1462	
CDC 9760 SMD (40MB)	626	
9762 SMD (80MB)	1462	
CDC 9710 (80MB)[2]	1462	
CDS T-82 (80MB)[2]	1452	

No. Cyls in LU/0
24

DEVICE CODE 27

DISC DRIVER ADDR 21664

BZUD ADDR 21404

LRC 240

NPTC 5

DFLG 40500

NTRS 1220

PHYU D

where D = drive unit no.
P = platter or surface

DISC COPY PROGRAM DISCUTILITY (LOTUS)

SETUP PARAMETERS
Use DSP to enter the following
in CONFIG, then re-IPL.

CONFIG Address	OLD Contents	NEW Contents
NONE		

NOTES

[1] When ordering, specify an "E PROM" and the drive unit number.

[2] Format and copy Ball BD-80 and CDC 9710 using entry for CDC 9762.

Figure 4-2. Sample Disc Specification Sheet

Mount Tape 1 of the appropriate MARK 5 or MARK 3 IRIS R8.n template, following the instructions in Section 2. When the tape is mounted, press <RETURN>.

```
NOTE: A "SCRATCH" LOGICAL UNIT MUST BE SUPPLIED ON THE DESTINATION DISC.
ALSO, PHYU AND FCYL MUST AGREE WITH THE DISC DRIVER TABLE SETTINGS
IN THE CONFIG FILE.

REQUIRED NUMBER OF CYLINDERS (OCTAL) FOR DESTINATION LOGICAL UNIT IS xxxxxx

ARE YOU SURE THE LOGICAL UNIT BEING WRITTEN TO IS A SCRATCH UNIT?  (Y/N)
```

This question and the next one are intended to help guard against inadvertently writing over important disc data. A "scratch" logical unit is one which may be completely overwritten.

Ensure that the logical unit is a scratch unit and enter Y. The system then asks

```
ARE YOU SURE IT CONTAINS AT LEAST xxxxxxx CYLINDERS (OCTAL)? (Y/N)
```

In this prompt, the system displays the number of cylinders to be written from tape to disc (i.e., the number of cylinders on disc to be overwritten).

Ensure that the scratch logical unit contains enough cylinders and then enter Y.

```
PLEASE WAIT..........
APPROXIMATELY 25 MINUTES IS REQUIRED FOR A WHOLE TAPE UNTIL NEXT TAPE CHANGE.
```

The system reads the tape, displaying the following progress messages as it does.

```
READING TRACK 0.....
READING TRACK 1.....
```

If an error is encountered, the system displays an error message. See Section 4.7 for examples.

When finished, the system displays

```
REMOVE THE CURRENT TAPE AND LOAD TAPE NUMBER 002 OF xxx OF THE SET.

PRESS CR WHEN READY!
```

Remove Tape 1 and insert Tape 2. The system reads the tape, displaying progress messages as it does.

```
PLEASE WAIT.........
APPROXIMATELY 25 MINUTES IS REQUIRED FOR A WHOLE TAPE UNTIL NEXT TAPE CHANGE.

READING TRACK 0.....
READING TRACK 1.....
```

When finished, the system displays

```
DISC AND TAPE CHECKSUMS AGREE, AND LOGICAL UNIT ADJUSTMENT IS DONE.
```

In addition to the individual block checksums, a four-word checksum for the whole LU has been stored as part of the data on tape. This additional check is done automatically and ensures that the disc has been correctly written.

If this is an LU/0 sysgen, IPL into a minimum configuration and complete system configuration, as described in Section 5. IPL into a full configuration and run CLEANUP (not CLEANUPX) on LU/0.

4.6.2 DISC-TO-TAPE

CTUTILITY allows the transfer of any nonzero logical unit (LU) between the same or different drives, different disc controllers, and between MARK 3, MARK 5 and MARK 8 systems.

The disc-to-tape procedure requires the following:

- A backup disc pack
- Several initialized cassette tapes (see Section 4.9)
- A scratch LU

The disc-to-tape procedure is accomplished in three steps:

1. Run CLEANUPX on the LU to be transferred as described in Section 4.6.2.1.
2. Copy the LU using the disc-to-tape procedure given in Section 4.6.2.2.
3. Transfer the LU to the destination system using the tape-to-disc procedure given in Section 4.6.1.

4.6.2.1 Running CLEANUPX

Before running CLEANUPX, back up the system disc packs. Use the backup copy for running CLEANUPX. DO NOT USE the original disc. CLEANUPX requires a scratch LU as a work file to clean up the DMAP and reorganize the files. The number of disc blocks required for the scratch LU is

```
    X = 1 + nnnn/256
```

where

```
    X    - number of blocks required
    1    - one block for the header block
    nnnn - number of blocks on the LU to be transferred
```

If the total number of disc blocks on the LU to be transferred is not equally divisible by 256, the result must be rounded upward. For example, assume there are 1000 blocks on the LU:

```
    1 + 1000/256 = 5 blocks for the scratch LU
```

The command format is

```
    CLEANUPX <CTRL-E>{key}<CTRL-E> {LU1} USING {LU2}
```

where

```
    key - password assigned to CLEANUPX (the default is X)
    LU1 - number of logical unit to be transferred
    LU2 - number of logical unit to be used as a workfile
```

4.6.2.2 Disc-To-Tape Procedure

After the LU is cleaned up, load CTUTILITY.

When it is loaded and executing, the system displays

```
  INITIALIZING UTILITY.....

  CTUTILITY IS LOADED.

  THIS REVISION OF CTUTILITY USES MARK 3 DISC ENTRY NUMBERS
  301 AND GREATER.  IF YOUR MARK 3 SPEC SHEETS START AT 1,
  THEN ADD 300 TO THEM TO USE THIS REVISION.

  PLEASE LOCATE YOUR PARTICULAR DISC CONTROLLER, DEVICE CODE
  AND DRIVE IN THE R8.1 PERIPHERALS HANDBOOK. (NOTE: IF NOT
  FOUND THEN IT IS NOT SUPPORTED BY POINT 4.)

  PLEASE ENTER THE DISC SPECIFICATION ENTRY NUMBER
  FOR THE SYSTEM YOU ARE "CURRENTLY" ON:
```

Locate the appropriate disc controller, device code and drive in the IRIS R8 Peripherals Handbook. The entry number is printed in the upper right corner of the disc specification sheet. Enter this number and press <RETURN>.

While the system reads SOV (System Overlay disc driver) and BZUD (Block Zero Utility Driver), it displays

```
READING SOV AND BZUD, PLEASE WAIT.....
```

When completed, the system displays

```
USING THE DISC SPECIFICATION SHEET, CALCULATE PHYU AND FCYL
FOR THE LOGICAL UNIT ON DISC TO BE USED FOR CTU TRANSFERS
(EITHER AS SOURCE OR DESTINATION).

PHYU=
```

The formula for calculating PHYU is given on the appropriate Disc Specification sheet, refer to the IRIS R8 Peripherals Handbook.

After the value of PHYU is entered, the system requests

```
FCYL=
```

After FCYL (first physical cylinder) is entered, the system asks

```
IS THIS A TRANSFER FROM TAPE TO DISC? (Y/N)
```

Enter N for no.

The system then asks

```
IS THIS A TRANSFER FROM DISC TO TAPE? (Y/N)
```

Enter Y for yes. The system then displays

```
ENTER THE CTU ORDER NUMBER TO BE ASSOCIATED WITH THESE TAPES
```

The order number is an arbitrary control number which may be assigned to this set of tapes. Enter a number between 1 and 65,000.

Next, the system displays

```
IT IS NECESSARY TO RUN CLEANUPX ON THE LOGICAL UNIT FIRST.  IF YOU HAVE NOT
DONE SO, IPL IRIS, RUN CLEANUPX AND THEN RESTART CTUTILITY FROM
THE BEGINNING.

PRESS CR WHEN READY!
```

If CLEANUPX has not yet been run on the desired LU, IPL IRIS and run CLEANUPX (see Section 4.6.2.1). CTUTILITY will have to be started again from the beginning after CLEANUPX has been run.

If CLEANUPX has already been run, press <RETURN> to continue.

The system calculates the highest RDA used on the logical unit being transferred and the number of tapes needed to contain the logical unit, while it displays

```
PLEASE WAIT...
```

After completing the calculation, the appropriate information is displayed

```
THE HIGHEST RDA USED ON THIS LOGICAL UNIT IS:xxxxxx
THIS LU WILL REQUIRE xxx ALREADY INITIALIZED SCRATCH TAPE(S).

LABEL THE NEXT INITIALIZED WRITE ENABLED "SCRATCH" TAPE AS NUMBER 001 OF xxx
PLACE IT IN THE CASSETTE DRIVE.

PRESS CR WHEN READY!
```

Label a scratch cassette tape as "001 of xxx" and mount it in the CTU drive, following the instructions in Section 4.2.1. When the tape is mounted, press <RETURN>.

The system displays

```
ARE YOU SURE THAT THE TAPE BEING WRITTEN TO IS A SCRATCH TAPE? (Y/N)
```

Double-check that the tape to be written on is really scratch. Then enter Y. The system displays

```
PLEASE WAIT.........
APPROXIMATELY 25 MINUTES IS REQUIRED FOR A WHOLE TAPE TILL NEXT TAPE CHANGE.
```

The system writes the specified logical unit from the disc to the tape, displaying progress messages as it proceeds.

```
WRITING TRACK 0.....
WRITING TRACK 1.....
```

When finished, the system displays

```
LABEL THE NEXT INITIALIZED WRITE ENABLED "SCRATCH" TAPE AS NUMBER 002 OF xxx
PLACE IT IN THE CASSETTE DRIVE.

PRESS CR WHEN READY!
```

The system displays a message to mount a new tape in the drive when a tape has been filled. This process continues until the entire logical unit has been written. The system then displays

```
DISC TO TAPE TRANSFER COMPLETE.
REMOVE CURRENT TAPE FROM CTU AND REPLACE IT WITH CTUTILITY AND PRESS CR.
```

NOTE

> Small logical units may require only one cassette tape; large logical units may require many tapes. Unused blocks at the end of a logical unit are not written to tape.

4.7 ERROR MESSAGES

Generally, errors will be displayed in the form:

```
ERROR: function FAILED! ON UNIT# 000000
   RDA = 000000, TRACK = 000000, COUNT = 000000, STATUS = 000000
WILL RETRY TO SEE IF ERROR IS CORRECTABLE.......
```

The "function" may be READ, WRITE, REWIND, or TRACK SELECT. All values are given in octal.

If the error is correctable, the program automatically resumes as if there had been no error. The system displays

```
RETRY WORKED! OPERATION IS PROCEEDING WITHOUT ERROR.
```

If the error is not correctable, the system displays

```
AN IRRECOVERABLE ERROR HAS BEEN ENCOUNTERED...PROGRAM ABORTED!!!
CHECK HARDWARE.
```

4.7.1 TAPE-TO-DISC ERROR EXAMPLE

If an error is encountered during the tape-to-disc transfer procedure, the system displays

```
ERROR:  READ FAILED! ON UNIT# 000001
   RDA = 000530, TRACK = 000000, COUNT = 000050, STATUS = 000122
WILL RETRY TO SEE IF ERROR IS CORRECTABLE......
```

The system will retry 16 times before deciding the error is not correctable. If so, it then displays

```
AN IRRECOVERABLE ERROR HAS BEEN ENCOUNTERED...PROGRAM ABORTED!!!
CHECK HARDWARE.
```

If the error is correctable, the program automatically resumes as if there had been no error.

4.7.2 DISC-TO-TAPE ERROR EXAMPLE

If an error is encountered during the disc-to-tape transfer procedure, the system displays

```
ERROR: WRITE FAILED! ON UNIT# 000001
   RDA = 000010, TRACK = 000001, COUNT = 000050, STATUS = 000005
WILL RETRY TO SEE IF ERROR IS CORRECTABLE......
```

The system will retry 16 times before deciding the error is not correctable. It then displays

```
AN IRRECOVERABLE ERROR HAS BEEN ENCOUNTERED...PROGRAM ABORTED!!!
CHECK HARDWARE.
```

If the error is correctable, the program automatically resumes as if there had been no error.

4.8 AUXILIARY CASSETTES

Auxiliary cassettes are available for a variety of utility programs. The types of programs and loading instructions are discussed in the following subsections.

4.8.1 UTILITIES - DDCOPY

The tape marked "UTILITIES/DDCOPIES" is a disc image copy of a nonzero logical unit containing user-oriented utilities and a number of system utilities. This tape contains DDCOPY programs for most of the controller/drive combinations which are not supported by DISCUTILITY. The nonzero logical unit should be copied to disc using the tape-to-disc transfer procedure given in Section 4.6.1. Transfer the appropriate DDCOPY program to LU/0 using the COPY processor. DDCOPY program names have the format, DDCOPY.nn, where nn corresponds to the entry number in the IRIS R8 Peripherals Handbook.

See Table 4-1 for the appropriate disc copy program. The ID shown in the table corresponds to that given for the controller/drive combination in the IRIS R8 Peripherals Handbook.

TABLE 4-1. COPY PROGRAMS

Entry #	ID	Copy Program
1	P410MB	DDCOPY.1
2	SI10MB	DDCOPY.2
3	DG4019	BLOCKCOPY
4	AMMEGA	BLOCKCOPY
5	DCC446	DDCOPY.5
6	MCQT50	DDCOPY.6
7	MCQT25	DDCOPY.7
8	MCQT80	DDCOPY.8
9	MCT200	DDCOPY.9
10	MCT300	DDCOPY.10
11	BA3170	DDCOPY.11
12	MC9Q40	DDCOPY.12
13	MC9Q80	DDCOPY.13
14	MC9CMD	DDCOPY.14
15	BABD50	DDCOPY.15
16	BA3150	DDCOPY.16
17	TF3380	DDCOPY.17
18	TF3350	DDCOPY.18
19	AE3100	DDCOPY.19
20	AE6200	DDCOPY.20
21	SI4050	DDCOPY.21
22	DGFL33	DDCOPY.22
23	P41040	DDCOPY.23
24	DGFL40	DDCOPY.24
25	SI8050	DDCOPY.25
26	DG20MB	DDCOPY.26
27	AE6240	DDCOPY.27
28	AE3140	DDCOPY.28
29	SMC12C	DDCOPY.29
30	S12S80	DDCOPY.30
31	SI05MB	DDCOPY.31
32	DG2533	DDCOPY.32
33	SI8073	DDCOPY.33
34	MC9F50	DDCOPY.34
35	QUECMD	*
36	700CMD	DISCUTILITY
37	P40K80	DISCUTILITY
38	P480MB	DISCUTILITY
39	P4300M	DISCUTILITY
40	S12300	DDCOPY.40
41	P4F135	DISCUTILITY
42	DG2073	DDCOPY.42
43	DG6067	BLOCKCOPY*
44	MC9202	DDCOPY.44
45	700LMD	DISCUTILITY
46	P4F168	DISCUTILITY
47	P41073	DDCOPY.47
48	DGFL73	DDCOPY.48
49	DG2540	DDCOPY.49
50	RN1033	DDCOPY.50
51	DG2573	DDCOPY.51
52	RN1040	DDCOPY.52
53	RN1073	DDCOPY.53

*Not supplied by Point 4.

4.8.2 STAND-ALONE UTILITIES

The CTUTILITY tape set may include a tape labeled Stand-Alone Utilities. It contains several disc format, copy, and diagnostics programs. The following sections provide a procedure for reading any of these programs from tape into memory.

User input is underlined. All user entries are followed by <RETURN>. The <RETURN> is not shown unless it is the only response, or more than one is required.

4.8.3 PREPARATIONS FOR LOADING STAND-ALONE UTILITIES

The method for loading stand-alone utilities depends on the type of PROMs on the CPU. Follow the procedure appropriate to the system.

4.8.3.1 Preparing to Load Utilities on CPUs Currently Running IRIS R8

For an IRIS R8.1 (or later) system, use the following procedure to load CTUTILITY.

1. Mount the CTUTILITY cassette into the CTU drive, following the instructions given in Section 4.2.1.

2. Enter

   ```
   SHUTDOWN <CTRL-E>{key}<CTRL-E> @73000
   ```

   where key is the password assigned to SHUTDOWN (the default is X).

3. Start CPU execution at location 73000. On a POINT 4 MARK 5/8 CPU, press APL and enter J73000. On other CPUs, it may be necessary to use the front panel switches.

4. Proceed to the loading procedure given in Section 4.8.4.

4.8.3.2 Preparing to Load Utilities on POINT 4 MARK 5/8 CPUs With CTU PROMs

1. Mount the Stand-Alone Utilities tape in the CTU drive, following the instructions given in Section 4.2.1.

2. Press APL on the computer front panel to enter MANIP.

3. Press <ESC> on the terminal. This should be echoed on the terminal as a backslash (\).

4. Proceed to the loading procedure given in Section 4.8.4.

4.8.3.3 Preparing to Load Utilities on POINT 4 MARK 3 CPUs With CTU PROMs

1. Mount the Stand-Alone Utilities tape in the CTU drive, following the instructions given in Section 4.2.1.

2. Press the RESET button on the computer front panel.

3. Enter R on the terminal.

4. When the tape stops and the cursor moves to the next line, use the loading procedure given in Section 4.8.4.

4.8.3.4 Preparing to Load Utilities on POINT 4 MARK 3 CPUs Without CTU PROMs

1. Insert the Stand-Alone Streamer tape cartridge in the streamer tape driver.

2. Press the RESET button on the computer front panel.

3. Enter H to load the streamer tape. If a program other than DBUG starts running, press RESET and enter the address of CTU DBUG

   ```
   J64000
   ```

4. Mount the CTUTILITY cassette into the CTU drive, following the instructions given in Section 4.2.1.

5. Use the loading procedure given in Section 4.8.4.

4.8.3.5 Preparing to Load Utilities on CPUs Without CTU PROMs

This procedure applies to non-POINT 4 CPUs and to POINT 4 MARK 5 and MARK 8 CPUs without CTU PROMs.

1. Mount the Stand-Alone Utilities tape in the CTU drive, following the instructions given in Section 4.2.1.

2. Starting at location 66040, enter the following into memory:

Location	Contents
66040	0
41	50377
42	20000
43	0
44	157765
45	154151
46	161576
47	154160
66050	20414
51	61025
52	63025
53	20766
54	101113
55	776
56	20764
57	101120
66060	40760
61	20761
62	40757
63	400
64	66000
65	111260
66	130662
67	126263
66070	106615

3. Start CPU execution at location 66050 to read in MARK 5 boot program from the CTUTILITY tape.

4. When the tape stops, start CPU execution at location 70000.

5. Use the loading procedure given in Section 4.8.4.

4.8.4 LOADING PROCEDURE FOR STAND-ALONE UTILITIES

1. Rewind the tape by entering

   <CTRL-Z><RETURN>

   A caret and a Z should be displayed on the screen. Wait for the cursor to move down to the next line.

2. To select track zero, enter

   <CTRL-T>0

3. Enter

   <CTRL-D>

   The system displays a directory similar to the one shown in Figure 4-3. The directory indicates the program ID, followed by the program starting block address and length. Explanatory comments are provided in the figure for clarification, and will not appear on the screen.

4. To select track one, enter

   <CTRL-T>1

5. Enter

   <CTRL-D>

   The system again displays a directory like that shown in Figure 4-3. The directory indicates the program ID, followed by the program starting block address and length. Explanatory comments are provided in the figure for clarification, and will not appear on the screen.

6. Choose a program from the directory and note its starting address and program length.

7. Enter

   0:

   The current contents of location 0 will be displayed followed by a colon.

8. Enter

   <CTRL-R>n1,n2

   where

   n1 = the program starting block address from the directory

   n2 = the program length in blocks from the directory

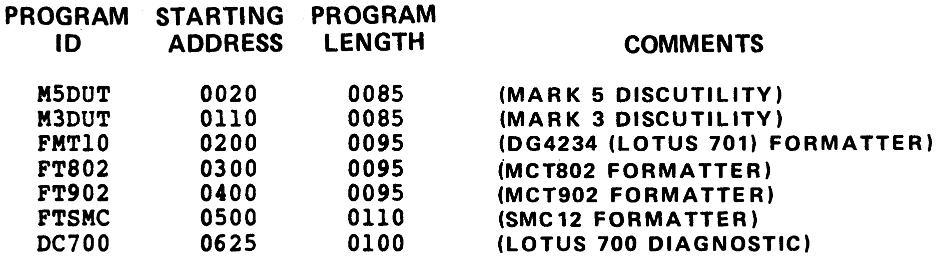
9. When the tape stops and the cursor moves to the next line, start the program by entering

   Jn

   where
   n = 2 for DISCUTILITY 1.4/2.5, or DC700
     or SMC 12 Formatter
   n = 400 for DG 4234 Formatter
   n = 1000 for MCT 802/902 Formatter

NOTE

The formatting programs for the DG 4234, SMC 12, and MCT 802/902 are not products of POINT 4 Data Corporation and are provided on this tape for convenience.



```
PROGRAM  STARTING  PROGRAM
   ID     ADDRESS   LENGTH               COMMENTS

 M5DUT     0020      0085     (MARK 5 DISCUTILITY)
 M3DUT     0110      0085     (MARK 3 DISCUTILITY)
 FMT10     0200      0095     (DG4234 (LOTUS 701) FORMATTER)
 FT802     0300      0095     (MCT802 FORMATTER)
 FT902     0400      0095     (MCT902 FORMATTER)
 FTSMC     0500      0110     (SMC12 FORMATTER)
 DC700     0625      0100     (LOTUS 700 DIAGNOSTIC)
```

NOTE: COLUMN HEADINGS AND COMMENTS ARE PROVIDED FOR CLARIFICATION AND WILL NOT APPEAR ON THE SCREEN.

Figure 4-3. Sample Cassette Tape Directory

4.9 CASSETTE TAPE INITIALIZATION

Initialization of a cassette tape requires that DBUG is in memory or the CPU is equipped with CTU PROMs. During the initialization procedure, each block on the cassette tape is written with a predetermined data pattern. It takes approximately eleven minutes to perform this procedure.

The procedure for cassette tape initialization is as follows:

1. Select track zero by entering

```
<CTRL-T>0<RETURN>
```

2. Initialize track zero by entering

```
<CTRL-I>9999<RETURN>
```

3. Rewind the tape by entering

```
<CTRL-Z><RETURN>
```

4. Select track one by entering

```
<CTRL-T>1<RETURN>
```

5. Initialize track one by entering

```
<CTRL-I>999<RETURN>
```

6. Rewind the tape by entering

```
<CTRL-Z><RETURN>
```

7. Return to normal DBUG mode or to CTUTILITY; press

```
<ESC>
```

Section 5
A GUIDE TO CONFIGURATION

This section contains the information required for setting up an IRIS system. It describes configuration aids, system files that may require modification, configuration of peripheral devices, time-sharing, processor options, user accounts, and other factors involved in configuring a system.

5.1 AIDS FOR CONFIGURATION

POINT 4 provides a GUIDE menu program and GUIDE modules (including GUIDE.LU and GUIDE.LPT) to assist in the configuration of a system. POINT 4 also provides an R8 Peripherals Handbook.

5.1.1 GUIDE

GUIDE is a menu program which provides access to the various guide programs for system configuration and setup. At the system prompt (#), enter

```
GUIDE
```

The GUIDE Menu program then displays topics for selection.

5.1.1.1 GUIDE.LU

GUIDE.LU is a program which provides directions for partitioning and for configuring logical units. It may be accessed from the GUIDE Menu or, at the system prompt (#), enter

```
GUIDE.LU
```

The program then guides the user through the necessary steps.

5.1.1.2 GUIDE.LPT

GUIDE.LPT is a program which assists the user in configuring the line printers for a system. It may be accessed from the GUIDE Menu or, at the system prompt (#), enter

```
GUIDE.LPT
```

The program then guides the user through the necessary steps.

5.1.1.3 GUIDE.MAGTAPE*

GUIDE.MAGTAPE is a program which assists the user in increasing the magnetic tape buffer size and in making the appropriate adjustments to the $MTAS driver. It may be accessed from the GUIDE Menu or, at the system command prompt (#), enter

<u>GUIDE.MAGTAPE</u>

The program then guides the user through the necessary steps.

*Available in the near future.

5.1.2 IRIS R8 PERIPHERALS HANDBOOK

The IRIS R8 Peripherals Handbook provides the information necessary for the configuration of disc controllers and terminals supported under IRIS R8.

5.2 SYSTEM CONFIGURATION FILE (CONFIG)

The CONFIG file holds general system information used by the entire system. It may be examined and modified by the use of DSP (see Section 2.3). After changes have been made, an IPL must be performed to load the newly configured system into memory.

Table 5-1 shows the parameters and contents of the CONFIG file for the IRIS R8 system. Locations are given in octal.

TABLE 5-1. CONFIG FILE

Location (octal)	Description
0-277	Reserved.
300-377	Initialization Table, reserved for use by SIR. DO NOT CHANGE!
400-577	General Information Table. See Section 5.2.1.
600-777	System Information Table. See Section 5.2.2.
1000-1177	Memory-resident Discsub Table. See Section 5.3.
1200-1377	Reserved.
1400-1777	Disc Driver Table. See Section 5.4.1.
2000-15777	Reserved.
16000-16377	Log-on Restrictions Table. See Section 5.5.
16400-17377	Log-on Program Startup Table. See Section 5.6.2.
17400-17777	IPL Program Startup Table. See Section 5.6.1.
20000-77777	Disc Drivers.

5.2.1 GENERAL INFORMATION TABLE (PSIZ)

The General Information Table contains data that is referenced during the IPL process. Its location is 400 (octal) in the CONFIG file.

Currently, the General Information Table consists of the system Partition Size (PSIZ) only.

Location (octal)	Label	Description
400	PSIZ	Partition Size. The size of the dynamic memory partition shared by all jobs.

For information on BASIC program partition requirements, refer to Section 5.12.1.

5.2.2 SYSTEM INFORMATION (INFO) TABLE

The System Information (INFO) Table contains system parameters starting at location 600 in the CONFIG file. Some of these parameters are set at Sysgen time, others may be modified to reflect the requirements of a particular system configuration. The locations (in octal) of the various parameters are shown in Table 5-2.

TABLE 5-2. INFO TABLE

<table>
<tr><th>Location (octal)</th><th>Label</th><th>Description</th></tr>
<tr><td>600</td><td>SDAT</td><td>System creation date (hours after BASEYEAR). DO NOT CHANGE!</td></tr>
<tr><td>601</td><td>SPED</td><td>Average CPU speed in instructions per millisecond:
<table>
<tr><th>Computer</th><th>Speed (octal)</th></tr>
<tr><td>POINT 4 MARK 5 and 8</td><td>2000</td></tr>
<tr><td>POINT 4 MARK 3</td><td>1200</td></tr>
<tr><td>NOVA</td><td>302</td></tr>
<tr><td>NOVA 1200 or D-116</td><td>653</td></tr>
<tr><td>NOVA 2 or D-116H</td><td>770</td></tr>
<tr><td>NOVA 800</td><td>1325</td></tr>
<tr><td>NOVA 3</td><td>770</td></tr>
<tr><td>SUPER NOVA</td><td>1255</td></tr>
<tr><td>SUPER NOVA SC</td><td>1762</td></tr>
</table></td></tr>
<tr><td>602</td><td>MILU</td><td>Maximum number of installed logical units - The total number of physical disc partitions defined in the Disc Driver Table. See Section 5.4.1.</td></tr>
<tr><td>603</td><td>NDCH</td><td>Number of data channels per port - Each data channel occupies eight words of memory. NDCH is usually set to 12 (decimal 10). Minimum NDCH is 2.</td></tr>
</table>

(Table continues on next page)

TABLE 5-2. INFO TABLE (Cont)

Location (octal)	Label	Description
604	LPCA	Location of port control area - Contains the address of port control block (PCB) for Port 0. It is automatically modified by SIR if any driver's attributes table specifies a PCB location.
605	TNAP	Total number of interactive ports - If the value in TNAP represents less than the total number of interactive ports contained in all driver's attributes tables, SIR <u>increases</u> the value automatically. **NOTE** This value is NEVER decreased automatically by the system - If the number of ports on the system is decreased, set TNAP to 1. SIR will then <u>increase</u> the number of interactive ports automatically.
606.	SPCF	Special conditions flags - These are flags which control certain system functions and options: Bit 14 - BASIC program error message flag (BEMF). Set to 0, error message text is printed. Set to 1 (40000 octal), messages are suppressed. Bit 15 - No Dirty Page flag (NDPF). Set bit 15 to 1 (100000 octal) to force a write-to-disc of any dirty buffer pool page. (Refer to Sections 5.13 and 5.14.) Bit 13 – defer buffer write until end of time slice All other bits are reserved.

(Table continues on next page)

TABLE 5-2. INFO TABLE (Cont)

Location (octal)	Label	Description
607	LEPS	Location of end of processor storage - This cell indicates the first available memory space above the processor overlay area. LEPS must be a multiple of 400 octal greater than the beginning of processor storage (BPS). DO NOT CHANGE LEPS unless RUN is modified accordingly!
610	TOPW	Highest addressable word in memory - IRIS ignores any memory above this address. The memory available above 77777 octal is used for user partitions and buffer pooling. Do not set TOPW above 77777 unless the CPU and all disc controllers on the system use a 16-bit memory address. All other devices use lower (<32K) memory.
611	ABUF	Size of auxiliary buffer area (number of words) - Must be at least 1004 words octal if indexed data files are to be used.
612		Reserved.
613	NCQN	Number of extra character queue nodes - SIR allocates two nodes per interactive port plus this number of extra nodes. Extra nodes are required to handle peak input rates if extra heavy character processing is required. Each node occupies two words of memory. Minimum value is two.
614	NNOD	Minimum number of free nodes - Each node occupies 32 words (decimal).
615	NSIG	Number of signal buffer nodes - This is the maximum number of signals which can be waiting to be received. Each node occupies 4 words of memory. Minimum value is 1.

TABLE 5-2. INFO TABLE (Cont)

Location (octal)	Label	Description
616	NSUB	Number of discsubs - The minimum value is one greater than the largest subroutine number in the DISCSUBS file.
617	KTSL	Time slice parameters - Used by the scheduler for determining the time slice (Long Time Slice * 400 + Short Time Slice). See Section 5.10.
620	TLOK	~~Minimum time (in tenths of a second) a locked partition is kept locked.~~ NA
621	KSCH	Scheduler parameters - For use by the time-sharing algorithm in job scheduling (APRI * 4000 + SINT.BIAS * 400 + SINT.AOI). See Section 5.10.
622 to 631		Reserved.
632 to 777		Reserved but not memory-resident.

5.3 IRIS DISCSUBS

Discsubs are subroutines which normally reside on disc. Some are standard and some are optional. Standard discsubs are those subroutines used by the IRIS Operating System in general. Optional discsubs are subroutines required by optional hardware and application packages. Some discsubs are more important and used more frequently than others. These should be made memory resident to ensure optimum system performance. Some discsubs are extended files requiring two disc blocks; some may include another discsub when memory resident.

Each discsub has an identifying number. Numbers of discsubs that are to be memory resident are entered into the DISCSUB Table starting at location 1000 in the CONFIG file. Based on the DISCSUB Table, selected discsubs are brought into memory during an IPL. POINT 4 presets the DISCSUB Table with standard discsubs that are most important and most frequently used by the IRIS Operating System (see Section 5.3.2). Depending on the memory available, this list may be expanded and/or modified by the user. A complete list of IRIS discsubs is given in Appendix C.

5.3.1 MAKING DISCSUBS MEMORY RESIDENT

In order to make a discsub memory resident, it is necessary to modify the memory-resident DISCSUB Table by the use of DSP. Some discsubs are included within another discsub. Appendix C (page C-10) gives a complete list of discsub numbers and their relationship to each other. If a discsub is a part of another, it cannot be loaded into memory by itself; it must go with the "parent". Alternatively, if a desired subroutine is such a "parent", subroutines contained within it are automatically transferred.

For example: OPEN is to be moved to memory. Appendix C-10 shows its number is 22+x (x=40000, an extended subroutine). Enter 22, since only the lower three digits of the discsub number may be entered into CONFIG's DISCSUB Table. Because Discsub 22 includes Discsubs 23, 24, and 25, they are included in the transfer.

Discsubs do not have to be entered in any particular order but the table must terminate with a -1 (177777).

The sequence of commands is as follows (user response is underlined):

```
    #DSP<CTRL-E>{key}<CTRL-E>CONFIG (where X is default for key)

    D1000

    1000:  1  3  15  30  177777 <ESC>

    E1004

    1004: 22  (DISCSUB number)

    1005: 177777

    1006: <ESC>

    D1000 (to check the entry)

    1000: 1  3  15  30  22  177777 <ESC>
```

To make the discsubs memory resident, shut down the system with the SHUTDOWN command and re-IPL.

SIR attempts to load all discsubs which have been included in the list at location 1000. If this process exceeds the space currently available in memory, then a minimum IPL is done automatically. This brings up the system with only the master terminal active. DSP may then be used to get the necessary space by modifying the memory-resident DISCSUB Table or by making other changes to accommodate the needed subroutines.

5.3.2 STANDARD DISCSUBS

Standard discsubs are subroutines that are required by the IRIS Operating System for its general functions. Some discsubs are more important and used more frequently than others. Based on the requirements of a normal 64KW system, POINT 4 presets the DISCSUB Table starting at location 1000 (octal) in the CONFIG file. Table 5-3 shows a list of the preset, standard discsubs arranged in order of priority, giving location, discsub number, and description.

The preset list of discsubs may be modified for system configuration requirements. Discsubs should be replaced starting with the lowest priority discsub (i.e., priority 18 - discsub number 41).

TABLE 5-3. PRESET STANDARD DISCSUB LIST

Priority	Location (octal)	Discsub Number	Name/Description
1	1000	67	AFSET
2	1001	100	LINKP&377 (2 blocks)
3	1002	3	FFILE
4	1003	15	ACNTLOOKUP
5	1004	22	OPEN&377 (2 blocks)
6	1005	26	CLOSE
7	1006	30	GETRR&377 (2 blocks)
8	1007	33	READITEM
9	1010	1	ALLOCATE
10	1011	40	CHARGE
11	1012	36	READCONTIG
12	1013	61	SEARCH&377 (2 blocks)
13	1014	62	SHUFFLE
14	1015	63	DEKEY
15	1016	27	CLEAR
16	1017	46	SPECIAL
17	1020	57	SIGPAUSE
18	1021	41	SYSCO
	1022	177777	-1

Table 5-4 is a listing of all standard discsubs arranged by discsub number.

Discsub numbers are assigned in the following ranges:

0-167	Reserved for use by POINT 4
170-177	Reserved for use by OEMs
200-237	Reserved for use by POINT 4
240-777	Not used at this time

The system manager determines which discsubs are most important to the particular system requirements and modifies the DISCSUB Table accordingly. Any discsubs may be made memory resident provided that enough memory is available to accommodate the discsubs and other system requirements such as a buffer pool, etc.

No priority is indicated except for the discsubs that are included in the preset configuration. These are identified with the priority number assigned by POINT 4.

TABLE 5-4. STANDARD DISCSUB LIST

Preset Prior-ity	Name	Number	Description
	FAULT	0+D	Abort task
9	ALLOC	1	Allocate disc blocks
	DALLC	2	Deallocate disc blocks
3	FFILE	3	Find file in index
	EXTEN	4	Change to extended file
	ALCON	5	Allocate contiguous file
	CDTA	6	Convert DRATSAB to ASCII
	CIA	7	Convert integer to ASCII (any Radix)
	CSTR	10	Compare strings
	PASSC	11	Password compare
	ERROR	12	Error routine for BASIC
	MESSA	13	Canned message to I/O buffer

TABLE 5-4. STANDARD DISCSUB LIST (Cont)

Preset Prior-ity	Name	Number	Description
	BREAK	14	Breakpoint setup for DSP
4	ACNTL	15	Account lookup
	DELET	16	Delete file
	PDELE	17+N	Delete processor or driver (included with #16)
	BUILD	20+X	Build new file
	BILDD	21+N+X	Build "$" file
5	OPEN	22+X	Open file or a device
	OPENU	23+N+X	Open file or device for update (included with #22)
	OPENL	24+N+X	Open and lock file or device (included with #22)
	OPENR	25+N+X	Open file or device for reference (included with #22)
6	CLOSE	26	Close channel
15	CLEAR	27	Clear channel
7	GETRR	30+X	Get record for read
	GETRW	31+N+X	Get record for write (included with #30)
	FINDI	32	Find item (not implemented)
8	READI	33	Read item
	WRITI	34+N	Write item (included with #30)
	WRITN	35	Write new item
11	READC	36	Read from contiguous file
	WRITC	37+N	Write into contiguous file (included with #36)

TABLE 5-4. STANDARD DISCSUB LIST (Cont)

Preset Prior-ity	Name	Number	Description
10	CHARG	40	Charge for file access
18	SYSCO	41	Transmit system command (CALL 98)
	CNVDA	42	Convert date to ASCII
	CNVAD	43	Convert ASCII to date
	CNVDT	44	Convert date and time (CALL 99)
	RDFHI	45	Read file header info (CALL 97)
16	SPECI	46	Special functions
	RECOV	47+D	Recover from stall or crash
	PATNF	50	Pseudo divide arc tangent function
	PLOGF	51	Pseudo divide natural log function
	PSQRF	52	Pseudo divide square root function
	PEXPF	53+X	Pseudo divide exponential function
	PSINF	54+X	Pseudo divide sine function
	PCOSF	55+N+X	Pseudo divide cosine function (included with #54)
	PTANF	56	Pseudo divide tangent function
17	SIGPA	57	Signal or pause
	DIREC	60	Set up directories for indexed file
12	SEARC	61+X	Search indexed file directory
13	SHUFF	62	Shuffle directory blocks
14	DEKEY	63	Delete key from directory
	RELEA	64+N	Release directory block (included with #63)
	FIXDI	65+X	Fix directories of moved indexed file

TABLE 5-4. STANDARD DISCSUB LIST (Cont)

Preset Priority	Name	Number	Description
	REOPT	66+X	Re-optimize indexed file directory
1	AFSET	67	Set up active file for swap-out
2	LINKP	100+X	Link programs (BASIC's "CHAIN")
	LOADP	101+N+X	Load BASIC program (included with #100)
	FINDF	110	Find file (CALL 96)
	RDISC	111	Read or write word to disc (CALL 95)
	CHFLT	112	Change file type (CALL 94)
	WRWRD	113	Write word to memory (CALL 93)
	JULIA	145	Julian date routine
	ETOA	167	Convert EBCDIC to ASCII

where

X - Extended subroutine (two blocks)

N - Included with another; causes a trap if made memory resident by itself

D - Version is disc-resident only

5.3.3 OPTIONAL DISCSUBS

Optional discsubs are associated with many of POINT 4's optional utilities and software packages. For maximum performance, at least some of the appropriate discsubs should be made memory resident. The following subsections provide the priorities for each of the utilities and packages.

Discsubs are made memory resident by entering the specific discsub numbers in the CONFIG File DISCSUB Table starting at location 1000. Use DSP to make these changes. For the procedure, see Section 5.3.1.

5.3.3.1 MAGTAPE Discsubs

Although none of the discsubs for MAGTAPE needs to be made memory resident, it is recommended that some be transferred to memory, based on their priority, to increase system efficiency.

Priority	Name	Discsub Number	Description
1	MTASK	72	'Other' post processing
2	MNEXT	76	Go to next drive
3	MTAPA	77	Read/Write functions
4	MRFIL	74	File input post processing
5	MTFPE	75	Read/Write transfers
6	MRC3	71	Read status, write EOF, and initialize media
7	MRFHD	73	Read file header

5.3.3.2 CTU Discsubs

CTU does not require that its discsubs be memory resident. The following list of CTU discsubs includes a priority designation for those users who wish to make some discsubs memory resident.

Priority	Name	Discsub Number	Description
1	CTNXT	102	CTU post processing task
2	CTUSR	103	Extended CTU Directory Search routine
3	CTUWENTRY	104	CTU Write Directory entry

5.3.3.3 Polyfile Discsubs

None of the polyfile discsubs are required to be memory resident. The following list of the discsubs includes their priority designation for those users who wish to make some discsubs memory resident.

Priority	Name	Discsub Number
1	PFSEA	122
2	READP	142
	WRITP	143 (included in Discsub #142)
3	PFSCN	133
4	VOLRE	134
5	DIRFN	136
6	MODE4	123
7	MODE5	127
8	PFABL	130
	PFALL	131 (included in Discsub #130)
9	PFRLS	132
10	PRCOM	124
11	PFSHF	125
12	PFSHX	126
13	DATCK	140
14	MAPBU	135
15	SZMAP	137
16	OPENP	141
17	MODE1	121
18	MODE0	120
19	CALLP	144

NOTE

Discsubs #142 and 143 are data access discsubs. If index access only is desired, efficiency will not be affected if neither discsub is memory resident.

5.3.3.4 STYLUS Discsubs

STYLUS does not require that its discsubs be made memory resident. However, its performance can be optimized by making as many discsubs as possible memory resident. It is strongly recommended that Discsub 105 which occupies only 8 words of CPU memory space be made memory resident as it is the "switchboard" for other STYLUS discsubs. If it is not memory resident, a double disc swap is required to access other subroutines.

A list of STYLUS discsubs in order of priority is given below.

Priority	Discsub Number
1	105
2	157 (includes #106)
3	155
4	151
5	156
6	153
7	154
8	107
9	152

Discsub #152 is used infrequently; there is no advantage in making it memory resident.

STYLUS is largely I/O-oriented and makes extensive use of text files and indexed contiguous files. It is important that the standard IRIS discsubs which process these files be made memory resident.

5.3.3.5 TYPIST Discsubs

TYPIST does not require that its discsubs be made memory resident. However, its performance can be optimized by making as many discsubs as possible memory resident.

All TYPIST discsubs are non-extended and none are nested. Discsubs number 163 and 165 are used infrequently, there is no advantage in making them memory resident.

A list of TYPIST discsubs in order of priority is given below.

Priority	Discsub Number
1	160
2	164
3	162
4	161
5	163

TYPIST does not require that other discsubs be made memory resident to improve performance.

5.4 DISC PARTITIONING

Disc partitioning is based on the type of hardware (e.g., disc drive(s) and disc controller) and the software (e.g., disc drivers) that are used at a particular installation. Generally, the disc drive and the disc controller are supplied by different vendors.

A. Hardware

A disc controller interfaces between the disc drive and the CPU. An IRIS system may contain multiple disc controllers.

A disc drive unit is usually one of three types:

1. Contains a removable cartridge or disc pack (e.g., SMD drives).

2. Contains one or more enclosed nonremovable discs (e.g., fixed-head or Winchester drives).

3. Contains a removable cartridge and one or more fixed platters (e.g., CMD or Diablo 44-type drive).

Each disc drive is a physical unit. Under IRIS, a physical unit contains one or more physical partitions. Each physical partition maps to a logical unit (i.e., physical partition = logical unit). LU/0 is always the partition assigned to the IRIS system files. Other logical units may be numbered from 1 to 127. Logical unit numbers need not be tied to physical partition numbers.

There are two reasons for partitioning a physical unit:

1. To minimize head travel time.

2. To separate files into functional groups.

POINT 4 recommends that

- Partition 0.0 (LU/0) be made small and contain only the IRIS system modules (i.e., no user program or data files).

- Most frequently accessed data files be combined in a central partition.

- Extreme partitions be used for archival files, backup files, and most BASIC programs.

B. Software

A disc driver is the software that accesses the disc system (i.e., the disc drive and controller). Under IRIS, two disc drivers are required for each disc system:

1. SOV (Disc driver sometimes called System Overlay)

2. BZUD (Block Zero Utility Driver)

These drivers are specific to each disc system (the disc drive and controller combination).

The size of a physical partition is determined by the number of cylinders. The maximum number of cylinders that can be configured in any one partition is a parameter dependent on the particular disc drive and controller.

While the parameters for IRIS system files (LU/0) are preset, all other LUs have to be defined by creating the Disc Driver Table(s) in the CONFIG file.

5.4.1 DISC DRIVER TABLE

The Disc Driver Table (see Figure 5-1) is set up in the CONFIG file. It consists of a Disc Controller Table for each controller, followed by a number of Disc Partition Tables. There is one Disc Partition Table for each partition on each drive. The Disc Driver Table must terminate with a -1 (177777).

Each Disc Controller Table contains information about a specific disc controller such as its device code and the appropriate software drivers (see Table 5-5).

Each Disc Partition Table defines one partition by specifying the drive number and other platter information in PHYU, the starting cylinder number in FCYL, and the total number of cylinders in NCYL (see Section 5.4.2).

The first Disc Controller Table is for the system disc controller. The first Disc Partition Table is for LU/0 (partition 0.0, the system logical unit). This table is a place holder. The information defining LU/0 is contained in SOV.

Values for partition 0.0 may be entered in one of two ways:

1. Enter all zeroes.

2. Enter nothing (i.e., leave whatever values are currently in the first Disc Partition Table).

When the system is IPLed, IRIS determines the parameters for partition 0.0 from information contained in SOV. The parameters are then stored in the eight words for partition 0.0 in the CONFIG file.

IRIS sets Word 1 of partition 0.0 to NPTC*400_8+MINB. The user sets all other partitions to NPTC.

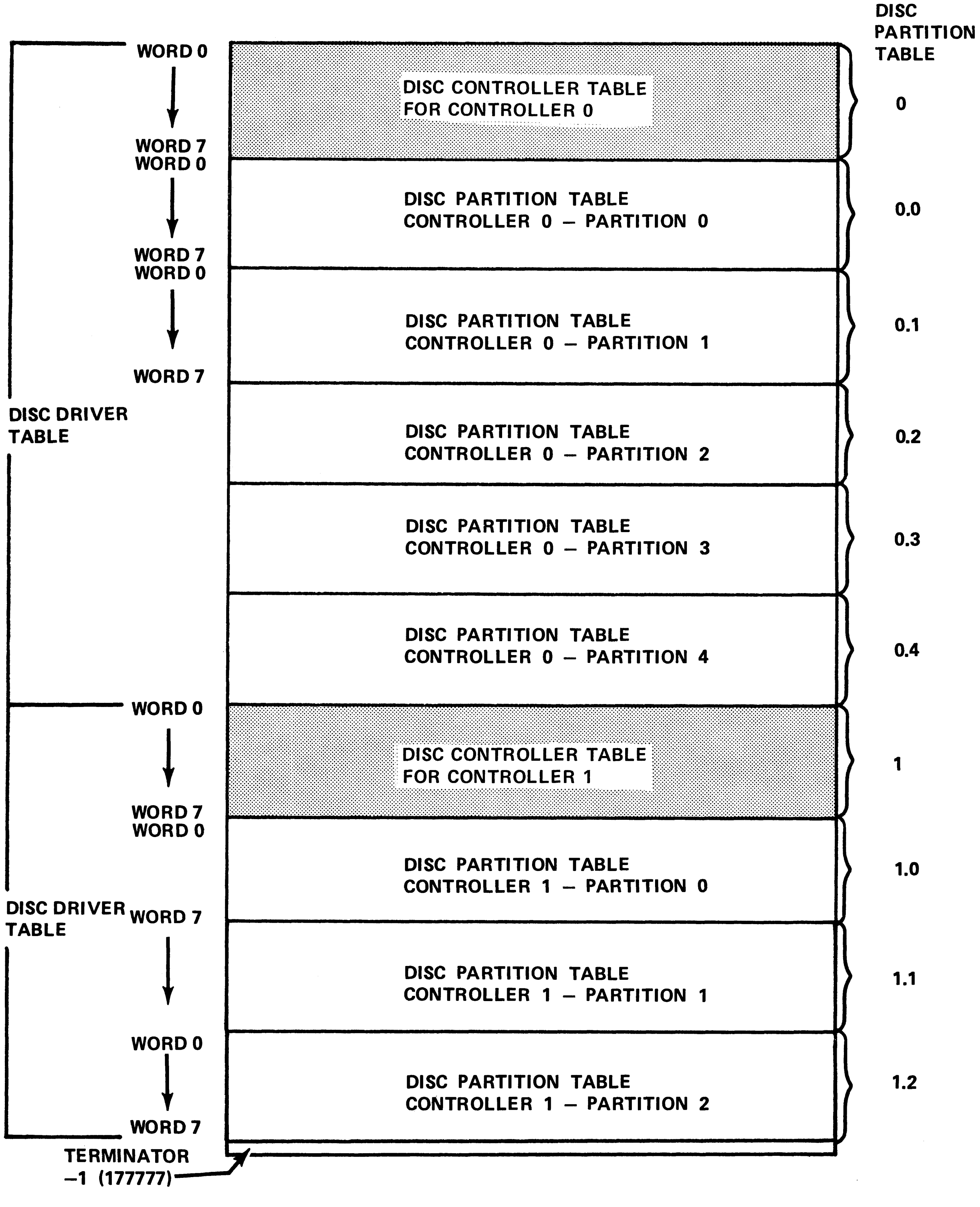

DISC PARTITION TABLE
WORD 0
DISC CONTROLLER TABLE FOR CONTROLLER 0
0
WORD 7
WORD 0
DISC PARTITION TABLE CONTROLLER 0 – PARTITION 0
0.0
WORD 7
WORD 0
DISC PARTITION TABLE CONTROLLER 0 – PARTITION 1
0.1
WORD 7
DISC DRIVER TABLE
DISC PARTITION TABLE CONTROLLER 0 – PARTITION 2
0.2
DISC PARTITION TABLE CONTROLLER 0 – PARTITION 3
0.3
DISC PARTITION TABLE CONTROLLER 0 – PARTITION 4
0.4
WORD 0
DISC CONTROLLER TABLE FOR CONTROLLER 1
1
WORD 7
WORD 0
DISC PARTITION TABLE CONTROLLER 1 – PARTITION 0
1.0
DISC DRIVER TABLE
WORD 7
DISC PARTITION TABLE CONTROLLER 1 – PARTITION 1
1.1
WORD 0
DISC PARTITION TABLE CONTROLLER 1 – PARTITION 2
1.2
WORD 7
TERMINATOR
–1 (177777)

Figure 5-1. Example of a Disc Driver Table

TABLE 5-5. DISC CONTROLLER TABLE

Word	Contents
0	Address of LUFIX (set by SIR)
1	Virtual (listing) address of read/write (R/W) entry of system disc driver
2	Virtual (listing) address of read/block (R/B) entry in BZUD driver
3	Number of disc partitions for this driver
4	Disc Controller's device code
5	Ratio for minimum number of blocks for this LU
6	Reserved
7	Reserved

Word 0 - SIR replaces the value in Word 0 with a pointer to the driver's actual location in memory; i.e., the LUFIX pointer.

Words 1 and 2 - The next two words point to the appropriate system disc driver and BZUD driver in the CONFIG file. Obtain the appropriate addresses from the IRIS R8 Peripherals Handbook.

Word 3 - Designates the number of partitions handled by this disc controller, i.e., the number of Disc Partition Tables that follow. (In the example shown in Figure 5-1, controller 0 has 5 partitions; controller 1 has 3.)

Word 4 - Contains the true device code for the controller. SIR uses this value to modify all I/O instructions in the system driver contained in CONFIG so that they use the given device code. INSTALL does the same for the BZUD driver in CONFIG. Thus, if the system has more than one disc controller, any controller (other than controller 0) may use any device code not in use, even if the device code is not listed in the IRIS R8 Peripherals Handbook.

Word 5 - Contains a ratio for calculating MINB, i.e., the minimum number of blocks that must be available in a logical unit to permit the building and saving of a new file. MINB is calculated as the quotient of the total number of blocks in the physical partition divided by the value of Word 5.

Words 6 and 7 - Reserved.

5.4.2 DISC PARTITION TABLE

Immediately following the Disc Controller Table (shown in Table 5-5) are the Disc Partition Tables. One Disc Partition Table is required for each partition contained in Word 3 of the Disc Controller Table. The format of the Disc Partition Table is shown in Table 5-6.

TABLE 5-6. DISC PARTITION TABLE

Word	Contents
0	Real memory address of LUVAR (set by SIR) OR INSTALL [illegible]
1	NPTC - Number of physical tracks per cylinder SPEC
2	DFLG - Disc flag word (see Figure 5-2) SPEC
3	Reserved
4	PHYU - Physical unit select word
5	FCYL - First cylinder of this partition
6	NCYL - Number of cylinders in this partition
7	NTRS - Number of IRIS tracks (NT*100_8+NS) SPEC

Word 0 - SIR replaces the value in Word 0 with a pointer to the logical unit's LUVAR table in memory.

Word 1 - The number of physical tracks per physical cylinder - used by IRIS to compute the physical cylinder track and sector for a given RDA.

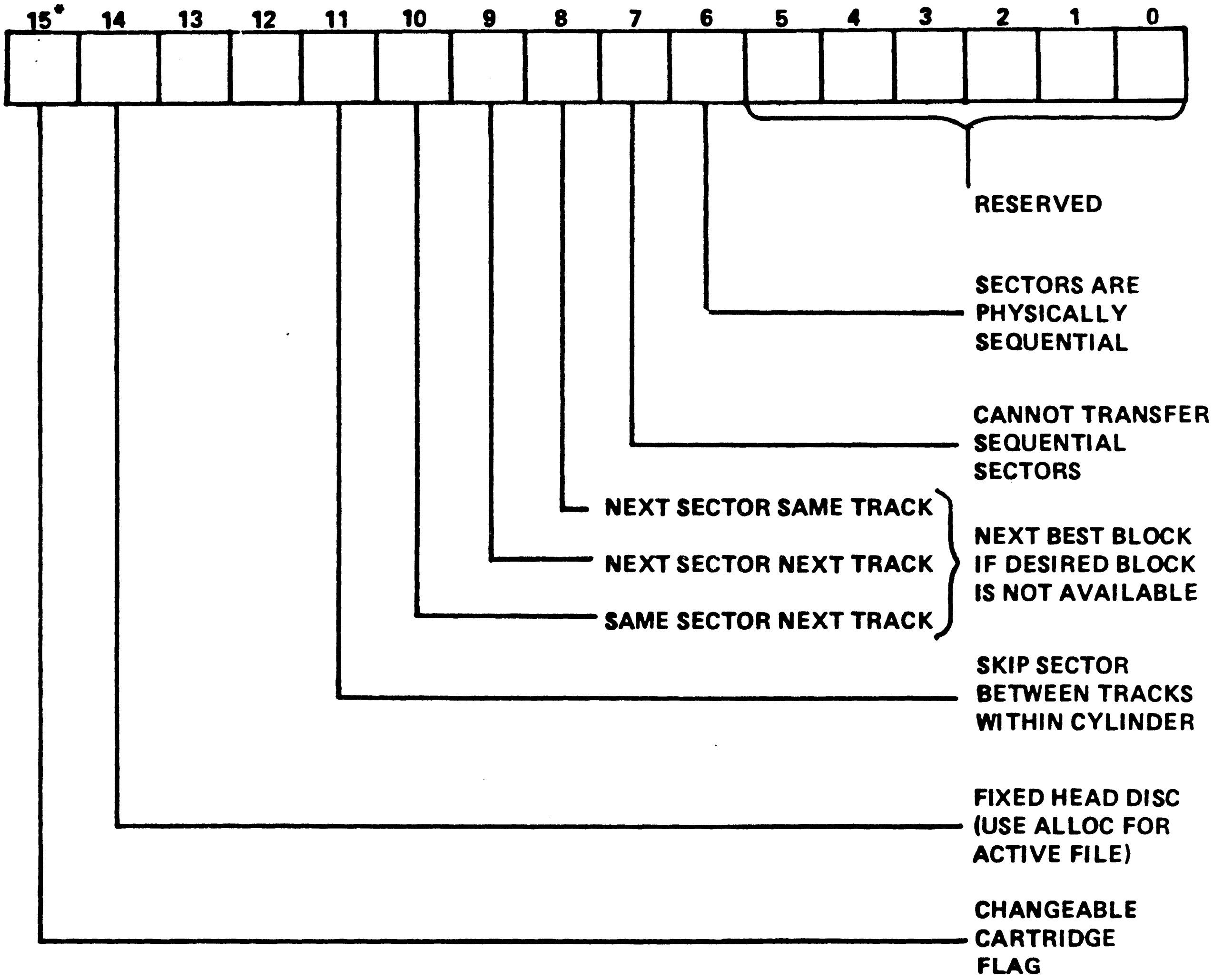
NOTE

> Word 1 in the table is always 1 for a CMD drive.

Word 2 - A flag word specified in the R8 Software Definitions - used to govern disc transfers.

Word 3 - Reserved

Word 4 - A physical unit select word used by the SOV to select the drive and the proper physical area of the drive (specified by DRIV) for access. The value is calculated with the formulas given in the IRIS R8 Peripherals Handbook.


15*
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
RESERVED
SECTORS ARE PHYSICALLY SEQUENTIAL
CANNOT TRANSFER SEQUENTIAL SECTORS
NEXT SECTOR SAME TRACK
NEXT SECTOR NEXT TRACK
SAME SECTOR NEXT TRACK
NEXT BEST BLOCK IF DESIRED BLOCK IS NOT AVAILABLE
SKIP SECTOR BETWEEN TRACKS WITHIN CYLINDER
FIXED HEAD DISC (USE ALLOC FOR ACTIVE FILE)
CHANGEABLE CARTRIDGE FLAG

*NOTE: BIT 15 IS THE MOST SIGNIFICANT BIT

Figure 5-2. Disc Flag Word (DFLG)

Word 5 - The starting cylinder number of the partition.

NOTE

Cylinders are numbered starting at 0.

Word 6 - The total number of cylinders contained in the physical disc partition. The value of NCYL may not exceed the maximum allowed. Refer to the IRIS R8 Peripherals Handbook for maximum number of cylinders.

Word 7 - The number of IRIS tracks and sectors computed as

$NT*100_8+NS$

where
NT - IRIS (logical) number of tracks/cylinder
NS - IRIS (logical) number of sectors/track

Refer to the IRIS R8 Peripherals Handbook for the value of NTRS.

The last Disc Partition Table for the last Disc Driver Table must be followed by a -1 (177777). This is entered where the next Word 0 would have been.

5.5 LOG-ON RESTRICTIONS

Log-on of selected users (identified by account group-user number) may be restricted to certain ports and/or certain times of day. This is controlled by a table starting at location 16000 (octal) in CONFIG. This block of CONFIG does not normally exist, so before entering any log-on restrictions it is necessary to allocate and zero out this block by giving DSP the commands:

```
FCONFIG
A16000
K16000,16377,0
```

The log-on restrictions table has four words per entry:

Word 0 - has an account number in the lower 14 bits (group-user number). The top 2 bits define a mode as follows:

00 Entry applies only to the group-user number given.

01 Entry applies to all users in given group with user number greater than the user number given.

10 Entry applies to all account numbers greater than the account number given as a 14-bit number (i.e., group = G and user ≥ U, or group > G, where G-U is the group-user number given).

11 Same as mode 10, but log-on is allowed if any entry in the whole table both matches and allows log-on. In all other modes, scan stops with the first match, e.g.,

$$041140 = \underbrace{01}_{\text{mode } 1}\underbrace{00\ 001\ 001}_{\text{group (9 decimal)}}\ \underbrace{100\ 000}_{\text{user (32 decimal)}} \quad \text{in binary}$$

Any restrictions on a user are determined by the first table entry where a match occurs; if no match is found, there are no restrictions on the particular user. When a match is found, Words 1 through 3 are used as follows:

Word 1 - has the form nnnppp in octal, where nnn ≤ 177 (octal). Any account selected by Word 0 may log on only if the port number falls within the range ppp thru ppp+nnn.

Words 2 and 3 - each have the form 00aabb, where aa<bb and bb≤60 (octal), and each of aa and bb is a half-hour since midnight (in octal). Any account selected by Word 0 may log on only if the current time t is in the range aa ≤ t < bb. The two words allow two time ranges for each day.

The value 000060 (octal) in Word 2 means "any time of day", and Word 3 is ignored. Words 2 or 3 are ignored if Word 1 indicates that the user is not on an allowable port.

For Word 2 or 3, the time of day values (based on a 24-hour clock) are as follows:

Time	aa or bb	Time	aa or bb
00:00	00	12:00	30
00:30	01	12:30	31
01:00	02	13:00	32
01:30	03	13:30	33
02:00	04	14:00	34
02:30	05	14:30	35
03:00	06	15:00	36
03:30	07	15:30	37
04:00	10	16:00	40
04:30	11	16:30	41
05:00	12	17:00	42
05:30	13	17:30	43
06:00	14	18:00	44
06:30	15	18:30	45
07:00	16	19:00	46
07:30	17	19:30	47
08:00	20	20:00	50
08:30	21	20:30	51
09:00	22	21:00	52
09:30	23	21:30	53
10:00	24	22:00	54
10:30	25	22:30	55
11:00	26	23:00	56
11:30	27	23:30	57
12:00	30	24:00	60

The table is terminated by a zero where Word 0 of the next entry would be, unless the block is full (64 entries) in which case a terminating zero word is not used. The value 100000 (octal) in Word 0 of an entry means "any account", and 177000 (octal) in Word 1 means "any port". For example,

```
000314  Word 0 (mode=0, group 3, user 14 octal)
002004  Word 1 (n=2, p=4 =>  ports 4, 5, 6)
004160  Word 2 (a=41, b=60 => 4:30 PM to midnight)
000016  Word 3 (a=0, b=16 => midnight to 7 AM)

041140  Word 0 (mode=01, group 11 octal, user 40 octal)
177000  Word 1 (n=177, p=0 => ports 0 thru 127 decimal (all))
002030  Word 2 (a=20, b=30 => 8 AM to noon)
003242  Word 3 (a=32, b=42 => 1 PM to 5 PM)

000000  Word 0 (terminator)
```

The first group of entries allows users to log on to account group 3 user 12 (decimal), only on ports 4 through 6, and only between 4:30 PM and 7:00 AM. The second group of entries allows users to log on to any account in group 9 with user number ≥32 to log on to any port but only during the hours 8:00 AM to noon or 1:00 PM to 5:00 PM.

5.6 AUTOMATIC PROGRAM START

The IRIS Operating System (R8.1 or later) permits two methods for specifying user programs to run automatically:

1. Selected initialization program at IPL time

2. Selected user program at log-on time

5.6.1 INITIALIZATION PROGRAM AT IPL TIME

An initialization program may be set up to run automatically upon completion of an IPL without operator intervention. Such a program may include:

- Automatic installation of logical units.

- Automatic log-on of selected ports to eliminate the distribution of restricted Account IDs.

- Automatic startup of user programs, provided the programs have been entered as described in Section 5.6.2.

The utility uses a table that must be set up at location 17400 (octal) in CONFIG. It occupies one block (17400 through 17777 octal) and contains a port number, an account number, and one program entry. The specified program must reside on LU/0.

After an IPL, the utility checks for the table. If it is found and the specified program resides on LU/0, SIR logs on the port for the specified account and copies the program name into the port's intermediate input buffer (IIB). The specified program starts up automatically. If the table is not found, the port is not logged on. If the program is not found on LU/0, the port is logged on but the program is not initiated.

The block occupied by the table does not normally exist. DSP is used to allocate the required space, and to enter the port number of the port from which the program is to run, the account number, and the filename as follows:

Word 0	- (17400) the logical system port number
Word 1	- (17401) account number (group and user)
Words 2-7	- (17402-17407) not used
Word 8	- (17410) program name

It is recommended that such a program be run from the first Mux port and the utility account. An example of the DSP commands required follows:

Command	Description
FCONFIG	Find the CONFIG file
A17400	
K17400,17777,0	Allocate a block for the table
17400:1	Usually the first Mux port
17401:100002	Account #0,2 = Utility account
I17410:STARTUP	Run the program named STARTUP

The name of the specified program, the port, and user account may be changed by using the DSP commands as shown in the example.

To log the port off in case the program is not found, an automatic log-off may be included with the program name at location 17410 in CONFIG:

```
STARTUP<CTRL-Z>BYE
```

To eliminate the display of the program name (stop echo), enter it at location 17410 as follows:

```
<CTRL-E><CTRL-X>STARTUP
```

These features may be combined when entering the program name to allow both options to take effect, e.g.,

```
<CTRL-E><CTRL-X>STARTUP<CTRL-Z>BYE
```

Automatic startup of the program will not occur under any of the following conditions:

- A minimum configuration IPL was done
- There is no CONFIG file
- CONFIG does not have a block at location 17400
- The word at 17400 is not a legal port number
- The word at 17401 is not a legal account number
- The IIB is less than 30 bytes in size
- The program does not reside on LU/0

5.6.2 USER PROGRAMS AT LOG-ON TIME

Selected users (identified by account group-user numbers) may have a specified BASIC program started automatically after log-on. The program to be started can be port-dependent.

The user's account number and specified program name are entered into a table starting at 16400 (octal) in CONFIG. This table may be up to two blocks long (16400 through 17377 octal), but the blocks do not normally exist. DSP is used to allocate the required blocks and the commands are similar to those for the log-on restrictions block (see Section 5.5).

Each block of the table holds up to 16 entries of 16 (octal 20) words each. Words 0 to 1 have the same form as in the log-on restrictions table; if these words indicate that a selected user is on a selected port, then Words 2-11 are assumed to be a BASIC program filename string, and that program is started running. The string may be in the form lu/filename; otherwise, the user's assigned logical unit is assumed. The following example shows the use of DSP commands for the required entries:

Command	Description
FCONFIG	Find the CONFIG file
A16400	
K16400,16777,0	Allocate block(s) for the table
16400:000201	User 2,1
16401:000000	Port 0
I16402:MENU	Program name
16420:140000	Any user
16421:001003	Ports 3 or 4
I16422:INVENTORY	Program name
16440:0	

In the first entry, if user 2,1 logs on to port 0, the program MENU will start automatically, following printout of log-on information and any messages.

In the second entry, if any user logs on to port 3 or 4, the program INVENTORY will start automatically, following printout of log-on information and any messages.

5.7 USER ACCOUNTS

An account utility program (ACCOUNTUTILITY) is provided for the purpose of setting up and maintaining user accounts. The program is entirely interactive and guides the user through the required functions by the display of various menu selections and appropriate prompts.

The information required for the ACCOUNTS file which resides on each logical unit is described in Table 5-7. The program adds a creation date and record number automatically.

TABLE 5-7. ACCOUNT FILE FIELDS

Description	Type	Range
Account ID	Alphanumeric	up to 12 char
User Name	Alphanumeric	up to 14 char
Privilege Level	Numeric	0-2
Account number - Group - User	Numeric Numeric	0-255 0-63
Assigned priority	Numeric	1-7
Connect Time	Numeric/U	0-1000 or U
CPU Time	Numeric/U	0-1000 or U
Assigned Logical Unit	Numeric	0-127
Disc Blocks (on assigned LU)	Numeric/U	0-65535 or U

Privilege Level may be set at three different levels:

0 - Lowest level - may access own files and those of other level 0 users not protected against such use

1 - Median level - may access level 1 and 0 account files not protected against such use

2 - Privileged level - may examine and modify other level account files; has access to certain system files not protected against such use (includes the utility account 0,2)

Privilege Level 3, the MANAGER account, is preset by POINT 4 and restricted.

Entry of the character U indicates a request for unlimited value.

The display of user account information for LU/0 is more comprehensive (see Figure 5-3) than the information displayed for a nonzero LU (see Figure 5-4).

```
         ACCOUNT STATUS ON LU#0

         ACCOUNT CREATION DATE:  mm/dd/yy
         RECORD NUMBER:          nnnnn
    (I)  ACCOUNT ID:             actid
    (N)  USER NAME:              name
    (L)  PRIVILEGE LEVEL:        p
    (A)  ACCOUNT GROUP,USER:     ggg,uu
    (P)  ASSIGNED PRIORITY:      p
    (M)  CONNECT TIME REMAINING: hhhh:mm:ss
    (S)  CPU TIME REMAINING:     hhhh:mm:ss
    (U)  ASSIGNED UNIT:          lu
    (D)  DISC BLOCKS ALLOTTED:   ddddd
         DISC BLOCKS IN USE:     ddddd
    (C)  TOTAL FILE USE CHARGE:  $nnn.nn
```

Figure 5-3. User Account Status On LU/0

```
         ACCOUNT STATUS ON LU#u

         RECORD NUMBER:           nnnnn
    (L)  PRIVILEGE LEVEL:         p
    (A)  ACCOUNT GROUP,USER       ggg,uu
    (D)  DISC BLOCKS ALLOTTED     ddddd
         DISC BLOCKS IN USE       ddddd
```

Figure 5-4. Account Status On A User Logical Unit

Entry of the number or letter displayed in the menu invokes the associated module or field.

The <ESC> key may be used for the following purposes:

- To exit a menu or program module and return to the previous menu (from the first input field of the screen)
- To back up to the previous entry field on the screen
- To exit from the Accounts File Maintenance Menu

The <RETURN> key may be used for the following purposes:

- To signal completed entry of data and move to next input field (when applicable)
- To signal entry of the default value of a field (when applicable)

To invoke the ACCOUNTUTILITY program and the Accounts File Maintenance Menu, at the system command prompt (#), enter

ACCOUNTUTILITY <RETURN>

The Accounts File Maintenance Menu will be displayed:

```
          ACCOUNTS FILE MAINTENANCE

     (0)  EXIT THE SYSTEM
     (1)  ADD NEW ACCOUNT
     (2)  MODIFY ACCOUNT
     (3)  DELETE ACCOUNT
     (4)  INQUIRE ACCOUNT
     (5)  LIST THE ACCOUNTS

          ENTER FUNCTION NUMBER:
```

0 - Chains back to the system command prompt (#)

1 - Allows addition of a new account to the system

2 - Allows modification of an account

3 - Allows deletion of an account

4 - Allows examination of an account

5 - Allows the listing of accounts on a Logical Unit

5.7.1 NEW ACCOUNTS

Selection 1 from the Accounts File Maintenance Menu invokes the new accounts module. User accounts are entered in two places. They must be entered on LU/0 and on the user's assigned logical unit. As the required information for all fields shown in Table 5-7 is entered, the program automatically assigns the input to the ACCOUNTS files on both LU/0 and the LU specified for the user. However, the allotted disc blocks are assigned to the specified LU only.

The program prompts for input, one field at a time:

```
    ENTER ACCOUNT ID:
```

After the input has been checked, it is followed by

```
    ENTER USER NAME:
```

Sequential prompting continues until the input for all the fields is complete.

The Escape <ESC> key may be used to back up to the previous entry field.

After the last field has been entered, the program redisplays all the fields on the screen and asks

```
    UPDATE THE ACCOUNT FIELDS ? (Y/N):
```

Enter Y to add the account. Enter N to reenter the fields and make any necessary corrections.

After Y has been entered the program responds

```
    UPDATING ACCOUNT FIELDS ON LU#n
    UPDATING ACCOUNT FIELDS ON LU#0
```

where n is the requested LU number.

The program then prompts for disc block allotments on other LUs

```
    ALLOT DISC BLOCKS ON OTHER LU
    ENTER LU/DISC BLOCKS (U=UNLIMITED):
```

where U is the total number of disc blocks available on the specified LU.

After the unit number and the number of disc blocks are entered, the program updates the account while displaying

```
    UPDATING ACCOUNT FIELDS ON LU#u
```

where u is the specified logical unit number.

If the account specified already exists on that Logical Unit, the program displays

```
    ACCOUNT EXISTS ON LU#u, NOT UPDATED
```

The program then repeats the LU/DISC BLOCKS question.

Press <ESC> to return to the Accounts File Maintenance Menu.

5.7.2 MODIFY AN EXISTING ACCOUNT

Selection 2 from the Accounts File Maintenance Menu invokes the change module which allows modification of an existing account. The program first prompts for the logical unit number:

```
    ENTER LOGICAL UNIT:
```

The program then displays the Account Modification Menu:

```
         ACCOUNT MODIFICATION ON LU#u

    (0)  RETURN TO MAIN MENU
    (1)  SELECT BY RECORD NUMBER
    (2)  SELECT BY ACCOUNT GROUP, USER
    (3)  SELECT BY ACCOUNT ID
    (4)  SELECT BY USER NAME

         ENTER FUNCTION NUMBER:
```

0 - Chains back to the Accounts File Maintenance Menu.

1 - Allows retrieval of an account by specifying its record number. The program displays the fields for modifications. If the record is not in the file (e.g., the record number given was wrong or it had been deleted) the program responds

```
         RECORD nnn NOT FOUND, TRY AGAIN !
```

2 - Allows retrieval of an account by specifying its Account Group, User. The program displays the fields for modifications. If the account is not found on the specified LU but the account exists on LU/0, the program responds

```
         g,u NOT FOUND
         ACCOUNT FOUND ON LU#0
         ADD THE ACCOUNT TO LU#n ? (Y/N):
```

Enter Y to add the account. Enter N to select another account. If the account does not exist on LU/0, the program responds

```
         g,u NOT FOUND, TRY AGAIN !
```

3 - Allows retrieval of an account by specifying its account ID. The program displays the fields for modifications. If the account is not found, the program responds

```
         {account id} NOT FOUND, TRY AGAIN !
```

4 - Allows retrieval of an account by specifying its user name. The program retrieves the account which matches the user name and displays the fields for modifications. If the user name is not found, the program responds

```
{user name} NOT FOUND, TRY AGAIN !
```

Selections 3 and 4 are available only when retrieving information for logical unit zero.

After the account information is displayed, the program asks

```
ENTER FIELD LETTER, <RETURN> WHEN DONE:
```

Enter the letter of the field you wish to modify. The program displays the field content. Enter new information or press <RETURN> for no change. After all modifications are entered, press <RETURN> at the Field Letter question. The program asks

```
UPDATE THE ACCOUNT FIELDS ? (Y/N):
```

Enter Y to accept the modifications. Enter N to retain the account as is. The program then asks for another account to be modified. Press Escape <ESC> to return to the Accounts File Maintenance Menu.

5.7.3 DELETE AN EXISTING ACCOUNT

Selection 3 from the Accounts File Maintenance Menu invokes the module which allows deletion of an account. The program first prompts for the user's assigned LU:

```
ENTER LOGICAL UNIT:
```

Enter the logical unit from which you wish to delete the account. The program then displays the Account Deletion Menu:

```
        ACCOUNT DELETION ON LU#n

(0)     RETURN TO MAIN MENU
(1)     SELECT BY RECORD NUMBER
(2)     SELECT BY ACCOUNT GROUP, USER
(3)     SELECT BY ACCOUNT ID
(4)     SELECT BY USER NAME

        ENTER FUNCTION NUMBER:
```

0 - Chains back to the Accounts File Maintenance Menu.

1 - Allows retrieval of an account by specifying its record number.

2 - Allows retrieval of an account by specifying its Account Group, User.

3 - Allows retrieval of an account by specifying its account ID.

4 - Allows retrieval of an account by specifying its user name.

Selections 3 and 4 are available only when retrieving information for logical unit zero.

After the account fields are displayed the program confirms the existence of the account and prompts

```
ACCOUNT g,u EXISTS ON THE FOLLOWING ACTIVE UNITS:
n; n; n; ...

DELETE THE ACCOUNT FROM ALL ACTIVE UNITS ? (Y/N):
```

Enter Y to delete the account from all active units. Enter N to retain the account on the listed units. If N is entered, the program asks

```
DELETE THE ACCOUNT FROM LU#n ? (Y/N):
```

Enter Y to delete the account from the specified unit. Enter N to retain the account. The program then repeats the prompt for each active unit to which the account is assigned space.

The program then requests another account number. If there are no further accounts to delete, press <ESC> to return to the Accounts File Maintenance Menu.

5.7.4 QUERY AN ACCOUNT ON A LOGICAL UNIT

Selection 4 from the Accounts File Maintenance Menu invokes the module which allows examination of a user account on a specified logical unit. Thus the first prompt is for an LU number:

```
    ENTER LOGICAL UNIT:
```

The program then displays the Account Enquiry Menu:

```
        ACCOUNT ENQUIRY ON LU#u

   (0)  RETURN TO MAIN MENU
   (1)  SELECT BY RECORD NUMBER
   (2)  SELECT BY ACCOUNT GROUP, USER
   (3)  SELECT BY ACCOUNT ID
   (4)  SELECT BY USER NAME

        ENTER FUNCTION NUMBER:
```

0 - Chains back to the Accounts File Maintenance Menu.

1 - Allows retrieval of an account by specifying its record number.

2 - Allows retrieval of an account by specifying its Account Group, User.

3 - Allows retrieval of an account by specifying its account ID.

4 - Allows retrieval of an account by specifying its user name.

Selections 3 and 4 are available only when retrieving information for logical unit zero.

After the account fields are displayed the program asks if you want to examine any other account:

```
    PRESS <RETURN> TO CONTINUE
```

The program then requests another account.

Press Escape <ESC> to return to the Accounts File Maintenance Menu.

5.7.5 LISTING ACCOUNTS ON A LOGICAL UNIT

Selection 5 from the Accounts File Maintenance Menu allows the listing of all the accounts on a specific LU. It gives the option to print or display the listing. The first prompt is for logical unit:

```
    ENTER LOGICAL UNIT:
```

Specify the appropriate LU; the program then prompts for the type of listing or display

```
    SELECT OUTPUT 1=DEVICE 2=FILE <RETURN>=CRT:
```

1 - outputs the report to a specified device. The program asks

```
        ENTER DEVICE NAME, <RETURN>=$LPT:
```

The input must begin with a dollar sign ($). If an error occurs it will be reported and the program will repeat the device name question. Press <ESC> to return to the output selection screen.

2 - outputs the report to a specified text file. The program asks

```
        ENTER LU/FILENAME:
```

The program will try to build the file. If it already exists, the user must include an exclamation point (!) at the end of the filename to overwrite the existing file. Press <ESC> to return to the output selection screen.

<RETURN> - outputs the report to the user terminal.

When printing is completed, the program responds

```
    PRESS <RETURN> TO CONTINUE
```

Press <RETURN> to select another logical unit.

Press Escape <ESC> to return to the Accounts File Maintenance Menu.

5.8 INTERACTIVE AND PERIPHERAL DRIVERS

A driver is enabled only if it is on LU/0 and there is a $-sign at the beginning of its name at IPL time. The CHANGE processor may be used to enable or disable drivers.

If there is not enough available space at IPL-time, a memory overflow occurs, and the system is automatically brought up into a minimum configuration. The user must either change the memory allocation in the CONFIG file, disable any unnecessary driver, or disable the new driver by removing the $-sign from its name.

Two steps are involved in adding a driver to a system:

1. Enable the driver. This is done by adding the $-sign to the filename (e.g., change PHA to $PHA).

2. Define a port for the device. Any device requiring a port must have that port defined in its Port Definition Table.

Each $-sign file has four tables:

- Entry Table

- Attribute Table (ATRIB)

- Linkage Table

- Port Definition Table (PDT)

The first three tables are preset and must not be changed.

When all necessary modifications have been made to the driver file's Port Definition Table, IPL the system. SIR makes all $ files memory-resident; if the driver specifies that it needs to be linked and/or initialized, SIR does that at IPL time.

5.8.1 DEVICE DRIVER FILE TABLE LOCATIONS

While the first three tables in a driver file are preset and must not be changed, it is necessary to know where they are located before the Port Definition Table can be modified.

5.8.1.1 Entry Table

The Entry Table is located at the beginning of the driver file and contains five words. The Entry Table must not be changed.

All driver files begin at location BPS in the CONFIG file. BPS is currently defined as location 32200 by IRIS DEFS.

Word 1 of the Entry Table contains the pointer to the ATRIB Table. Thus, location 32201 will give the location of the ATRIB Table.

32640

5.8.1.2 ATRIB Table

The ATRIB Table is located at the end of the driver file. It always contains three words. Thus, the location of ATRIB+3 will give the address of the Linkage Table (see Figure 5-5). The ATRIB Table must not be changed.

5.8.1.3 Linkage Table

The Linkage Table starts at ATRIB+3 and consists of two words per entry. It may have zero or more Linkage Table entries. If there is more than one, the Linkage Table increments by two words for each entry and terminates with -1 (177777). If there is no Linkage Table, the -1 is found at ATRIB+3. The Port Definition Table immediately follows. The Linkage Table must not be changed.

TABLE	DISPLACEMENT	CONTENTS		COMMENTS
ATRIB	0	X		ATRIB IS LOCATED AT THE END
	1	X		OF THE DRIVER FILE. ITS ADDRESS
	2	X		IS FOUND THROUGH THE POINTER
				IN WORD 1 OF THE ENTRY TABLE.
LINKAGE TABLE	0	X	1	THE LINKAGE TABLE FOLLOWS THE
	1	X		ATRIB TABLE. EACH DRIVER MAY
	2	X	2	HAVE 0 OR MORE LINKAGE TABLE ENTRIES
	3	X		(2 WORDS PER ENTRY). THE LINKAGE
	4	X	3	TABLE TERMINATES WITH A –1(177777).
	5	X		IN THIS EXAMPLE THERE ARE 3 LINKAGE
	6	177777		TABLE ENTRIES.
PDT	0	PORTS	1	THE PORT DEFINITION TABLE FOLLOWS
	1	PCW		THE LINKAGE TABLE. IT MAY HAVE 0 OR
	2	BUFFER		MORE PART DEFINITION ENTRIES (8 WORDS
	3	RDE/TTC		PER ENTRY). THE PORT DEFINITION TABLE
	4	RESERVED		TERMINATES WITH A –1(177777).
	5	RESERVED		
	6	AF		
	7	RESERVED		
	10	PORTS	2	
	11	PCW		
	12	BUFFER		
	13	RDE/TTC		
	14	RESERVED		
	15	RESERVED		
	16	AF		
	17	RESERVED		
	20	177777		

Figure 5-5. Driver File Tables

5.8.1.4 Port Definition Table (PDT)

The Port Definition Table (PDT) follows the Linkage Table's terminator. It consists of zero or more entries and must be terminated by a -1 (177777). PDT is located at ATRIB+3+Linkage Table. If there is no PDT, there will be a terminator at its location.

The Port Definition Table consists of eight words per entry as follows:

Word 0 - Number of ports (with the characteristics described in words 1-7).

Word 1 - Port Control Word (PCW) in the Port Definition Table (PDT) and in the port control block (PCB) controls various characteristics of the port such as baud rate, modem control, parity checking, etc., provided that the hardware allows these parameters to be controlled by software. (For example, with the POINT 4 MARK 3, baud rate is hardware controlled.) PCW should be zero for any device which cannot control any of these characteristics. The general format of the PCW is shown in Figure 5-6. Values that may be entered into PCW for a MARK 3 System are shown in Table 5-8.

Word 2 - Input/output buffer size (bytes)

Word 3 - Return delay (RDE) and terminal type code (TTC)

RDE - Carriage return delay. For a port on a POINT 4 Mux, the delay is in fiftieths of a second. For ports on all other devices, the delay is the number of null codes before the next character. RDE is given in the upper (left-hand) byte.

TTC - The number assigned to a Terminal Translation Module (see Section 5.9). TTC is given in the lower (right-hand) byte.

Word 4 - Reserved.

Word 5 - Reserved.

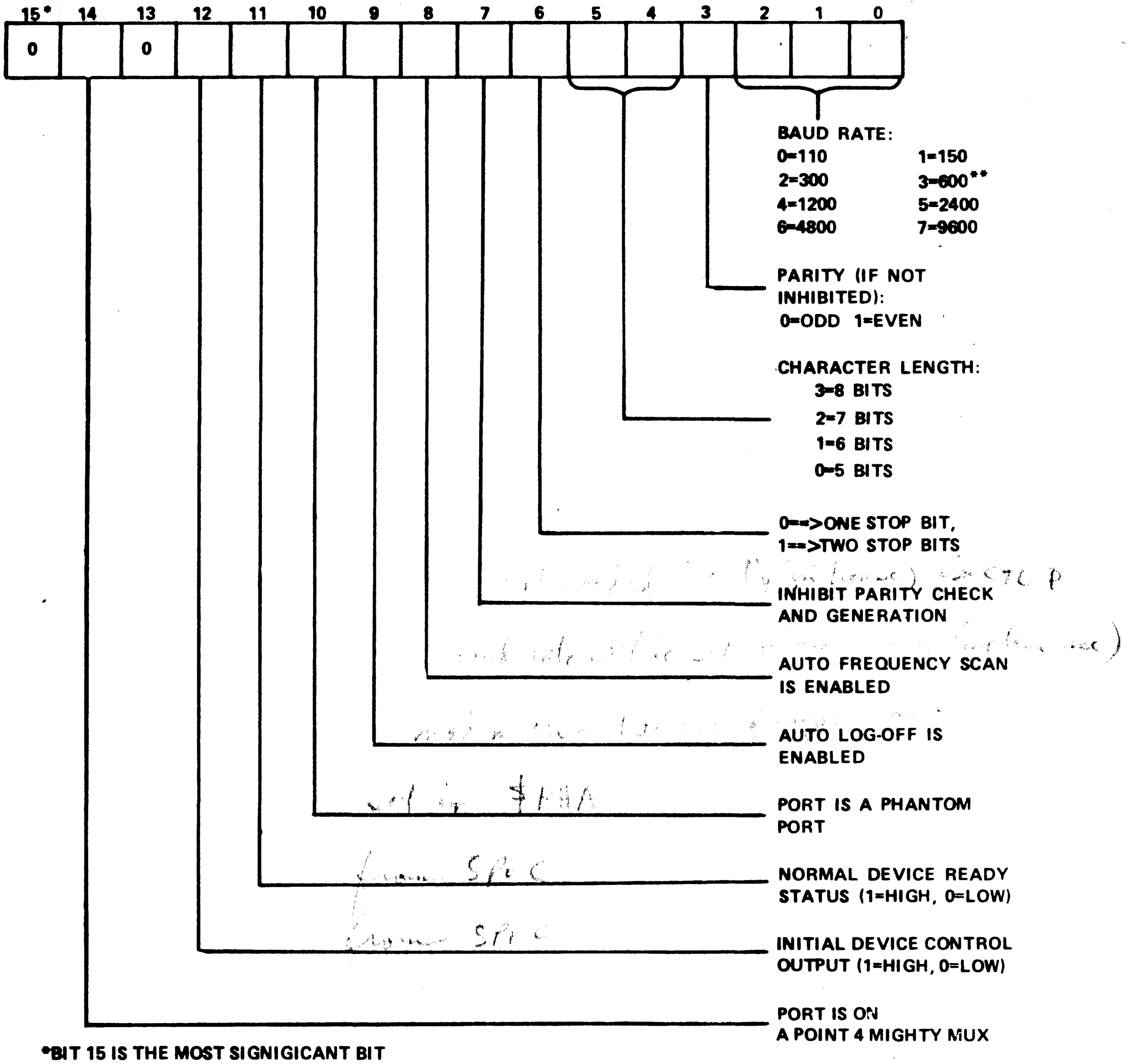
Word 6 - Size of active file on disc in blocks. The recommended size is 40 (octal) blocks.

Word 7 - Reserved.

The port entries must be terminated by a -1; the table may be empty, but the -1 terminator is required. A Port Control Block (PCB) is assigned for each port listed.

NOTE

PDT cannot extend over a block boundary.



BAUD RATE:
0=110 1=150
2=300 3=600**
4=1200 5=2400
6=4800 7=9600

PARITY (IF NOT INHIBITED):
0=ODD 1=EVEN

CHARACTER LENGTH:
3=8 BITS
2=7 BITS
1=6 BITS
0=5 BITS

0==>ONE STOP BIT,
1==>TWO STOP BITS

INHIBIT PARITY CHECK AND GENERATION

AUTO FREQUENCY SCAN IS ENABLED

AUTO LOG-OFF IS ENABLED

PORT IS A PHANTOM PORT

NORMAL DEVICE READY STATUS (1=HIGH, 0=LOW)

INITIAL DEVICE CONTROL OUTPUT (1=HIGH, 0=LOW)

PORT IS ON A POINT 4 MIGHTY MUX

*BIT 15 IS THE MOST SIGNIGICANT BIT

**ON A POINT 4 MUX WITH THE 19200 BAUD OPTION, 3=19200, NOT 600

Figure 5-6. Port Control Word Format

TABLE 5-8. PCW VALUES FOR A MARK 3 SYSTEM

No. of Data Bits	Parity	No. of Stop Bits	PCW Value
7	Even	2	140201
7	Odd	2	140205
7	Even	1	140211
7	Odd	1	140215
8	Inhibited	2	140221
8	Inhibited	1	140225
8	Even	1	140231
8	Odd	1	140235

5.8.2 MASTER PORT

The master port's main function is to access the system during an IPL. It is always port 0.

A master port is one of the following:

- A terminal on a POINT 4 MARK 3 Mux port 0
- A terminal on a POINT 4 310 Mux port 0 (master terminal mode)
- A terminal (e.g., Teletype or CRT) on a separate controller (device code 10/11)

In any case, the master port driver is a Teletype-type driver, device code 10/11, residing in REX. The address of the driver's Port Definition Table (PDT) is to be found at location 200 (octal) in REX. The PDT may be modified in accordance with the instructions given in Section 5.8.1.4.

Port 0 is the only interactive port on the system when one of the following conditions exists:

- The system is operating under a minimum configuration IPL
- An IPL is in progress
- The system is executing in DBUG or another stand-alone program

Port 0 becomes one of a number of interactive real ports when the system is operating under a full configuration IPL, has a separate Teletype controller (device code 10/11) and has one of the following:

- A Mux other than a POINT 4
- A POINT 4 310 Mux without master terminal mode

Port 0 becomes a phantom port (see Section 5.8.6) when the system is operating under a full configuration IPL using either a POINT 4 310 Mux with master terminal mode or a POINT 4 MARK 3 Mux.

The terminal assigned to port 0 reverts from the control of the Teletype driver to the control of the Mux driver (device code 25) and is automatically assigned another port number.

5.8.3 POINT 4 310 OR MARK 3 MULTIPLEXER ($MMUX)

$MMUX is the system interface for the POINT 4 310 or MARK 3 Multiplexer. It can be configured to accommodate any combination of CRT terminals, printers, modems, and other RS-232 devices on a port-by-port basis.

Each port, including its characteristics, is defined in the Port Definition Table. The word just before ATRIB must be set to reflect the total number of ports (in octal) physically present in the Mux system, even if all ports are not actually used. If the number of ports defined in the Port Definition Table is less than the number of physical ports on the Mux system, a halt (77277) occurs at IPL-time. This prevents possible destruction of data on disc or in memory.

For example, if a POINT 4 301 expansion board with 16 ports is connected to the basic 310 board (which has 8 ports), then the total number of ports defined in the $MMUX Port Definition Table must be exactly 24, and the word at ATRIB-1 must contain the value 30 octal.

The Port Control Word (the second word in each set of eight words in the Port Definition Table) must be set up in accordance with Section 5.8.1.4. In particular, be sure to set the "POINT 4 MIGHTY MUX" bit 14 (i.e., octal 40000 bit).

The following example of a Port Definition Table (PDT) for $MMUX assumes a POINT 4 MIGHTY MUX with a 301-A8 expansion board (16 ports total):

1. Ten interactive ports with CRTs (ports 1-10)

   - 9600-baud
   - 7-bit character plus even parity bit
   - 135-byte I/O buffer
   - 32-block active file

2. Two interactive ports with CRTs (ports 11 and 12)

   - 4800-baud
   - 7-bit character plus odd parity bit
   - 135-byte I/O buffer
   - 32-block active file

3. One interactive port for a modem (port 13)

   - 300-baud
   - 7-bit character plus even parity bit
   - 1 stop bit
   - 80-byte I/O buffer
   - 32-block active file
   - auto frequency scan enabled
   - data terminal ready set high

4. One Cassette Tape Unit (port 14)

   - 9600-baud
   - 8-bit character without parity
   - 512-byte data buffer with an extra 25 bytes of data buffer for CTU commands.

5. One unused port (port 15)

6. One non-interactive port used for a line printer (port 16)

   - 9600-baud
   - 8-bit character without parity
   - normal device ready status is high
   - 512-byte I/O buffer

The PDT for this sample configuration is shown in Figure 5-7.

```
ATRIB- 1=20

ATRIB TABLE
 •  )
 •  } 3 WORDS
 •  )

LINKAGE TABLE
 •  } 2 WORDS
 •
177777

PORT DEFINITION TABLE
   12   50057    207   0   0   0   40    0
    2   50046    207   0   0   0   40    0
    1   55452    120   0   0   0   40    0
    1   40367   1031   0   0   0    0    0
    1   40000      0   0   0   0    0    0
    1   44277   1000   0   0   0    0    0
   177777
```

Figure 5-7. Sample Port Definition Table For $MMUX

For easy reference, some of the most commonly used Port Control Words are listed below:

For CRT terminals:

50277 = 8-bit character, no parity, 9600 baud

50077 = 8-bit character, even parity, 9600 baud

50067 = 8-bit character, odd parity, 9600 baud

50057 = 7-bit character, even parity, 9600 baud

50047 = 7-bit character, odd parity, 9600 baud

For Teletype:

40360 = 8-bit character, no parity, 2 stop bits, 110 baud

40150 = 7-bit character, even parity, 2 stop bits, 110 baud

For Modems:

55452 = 7-bit character, even parity, 1 stop bit, 300 baud with modem control (Data Terminal Ready set high, Auto Log-Off and Auto Frequency Scan enabled)

55054 = 7-bit character, even parity, 1 stop bit, 1200 baud, Data Terminal Ready set high, Auto Log-Off, but no Auto Frequency Scan

For Line Printers:

44277 = 8-bit character, no parity, 9600 baud, printer "ready" status high (pin 20 mux connector)

40277 = same as 44277 but "ready" status low

For Cassette Tape Units:

40367 = 8-bit character, no parity, 2 stop bits, 9600 baud

5.8.4 DATA GENERAL 4060-TYPE MULTIPLEXER ($DGMX)

POINT 4 provides a system interface named $DGMX for those installations that use a Data General 4060-type Mux with device code 30.

Any combination of CRT terminals and line printers may be configured on a port-by-port basis. Each port, including its characteristics, is defined by hardware options on the Mux and system parameters in $DGMX's Port Definition Table (PDT).

Characteristics defined by the hardware (refer to the manufacturer's specifications) are:

- Baud rate
- Character length
- Parity generation and checking (may be done by software)
- Number of stop bits
- Device ready status

Characteristics defined in the PDT (refer to Section 5.8.1.4) are:

- Number of ports
- Parity generation and checking (may be done by hardware)
- Carriage return delay
- Terminal type code
- Size of Active File

To define the total number of ports on the system, the word just before ATRIB (i.e., ATRIB-1, see Section 5.8.1.2) must be set equal to the total number of ports (in octal). In general, this number should reflect the total number of physical ports even if not all ports are to be used.

Where a Data General 4060-type multiplexer is used, parity checking is done by the system after a character is input. Word 1 (Port Control Word) in the PDT is set to the type of parity checking desired.

The following example of a PDT for $DGMX configuration assumes a Data General 4060-type Mux with eight ports:

1. Five interactive ports with five CRTs (ports 1-5)

   - Even parity (handled by software)
   - 135-byte I/O buffer
   - 32-block active file

2. One interactive port with a modem (port 6)

   - Odd parity (handled by software)
   - 80-byte I/O buffer
   - 32-block active file

3. One unused port (port 7)

4. One non-interactive port running a line printer with a 512-byte I/O buffer (port 8)

The PDT for this sample configuration is shown in Figure 5-8.

```
ATRIB-1 = 10

ATRIB TABLE

●
●     3 WORDS
●

LINKAGE TABLE

177777    (No Linkage Table entries)

PORT DEFINITION TABLE

  5   10   207   0   0   0   40   0
  1    0   120   0   0   0   40   0
  1    0     0   0   0   0    0   0
  1    0  1000   0   0   0    0   0
177777
```

Figure 5-8. Sample Port Definition Table For $DGMX

5.8.5 REAL-TIME CLOCK ($RTC)

If a Real-Time Clock is required and the system does not have the POINT 4 310 MIGHTY MUX, enable the RTC file by changing its name to $RTC.

NOTE

> The POINT 4 MARK 3 Mux has a built-in real-time clock.

5.8.6 PHANTOM PORT

A phantom port is similar to an interactive port in that it has an Active File, a Port Control Block, and a Data File Table. User programs may be run on it and it will accept system commands. A phantom port differs from an interactive port in that it has no I/O interface and cannot be accessed via a terminal. For information on how to access a phantom port, refer to the IRIS R8 User and Business BASIC manuals. There may be any number of phantom ports on a system since they are not limited by physical hardware ports.

The phantom port driver file is named PHA. A phantom port may be set up as follows:

1. Enable the driver by using the CHANGE command:

```
#CHANGE PHA

IF NO CHANGE, PRESS RETURN
NEW NAME: $PHA

COST = $0.00
NEW COST? <ESC>
#
```

2. Set up the PDT for $PHA using DSP and the instructions given in Section 5.8.1.4.

3. SHUTDOWN the system.

4. Re-IPL.

5.8.7 LINE PRINTERS

Selection of the correct driver for a line printer is not based on the particular type of printer; the selection of the driver is based on the interface between the line printer and the computer. The following controllers provide appropriate interfaces:

- POINT 4 310 MIGHTY MUX
- POINT 4 MARK 3 PIB
- Data General 4060-type multiplexer
- Device Code 17 Controller Board
- Device Code 51 Controller Board

IRIS supports several types of universal line printer drivers. They are 'universal' because they can be customized to support almost any particular make or model of line printer.

POINT 4 supplies all the line printer driver files supported under IRIS. They are read onto the system disc at sysgen time (see Section 3.8) or are included on the prepared system disc, floppy, etc. The names for the drivers and their specifications are given in Table 5-9.

As an aid to configuring line printers, POINT 4 supplies the GUIDE module, GUIDE.LPT. The program can be accessed from the GUIDE Menu or from the system prompt at any time since it makes no changes to any file by itself. It provides specific information on how to set up a line printer.

POINT 4 recommends that the system be backed up before using DSP.

Setting up a line printer requires seven steps:

1. Select the appropriate driver from Table 5-9.

2. Copy the driver.

3. Run GUIDE.LPT to customize the line printer.

4. Enter GUIDE.LPT's output in the line printer driver file.

5. Set up the line printer in the appropriate mux driver.

6. Enable the driver.

7. Test the line printer and make adjustments if necessary.

5.8.7.1 Select Appropriate Driver

IRIS offers four different drivers; select the driver appropriate for the particular system using the information given in Table 5-9.

TABLE 5-9. IRIS LINE PRINTER DRIVERS AND SPECIFICATIONS

Driver Name	Specifications
LPTM	Line printers using a POINT 4 310 or MARK 3 Multiplexer. If the line printer has an RS-232 serial option, it may be plugged into a port on the POINT 4 310 or MARK 3 Multiplexer. The Mux outputs to the printer using Direct Memory Access (DMA). Since the CPU does not have to handle individual characters, the result is better performance for all users while print jobs are running.
LPTP	Line printers using a device code 17 controller. If the line printer does not have an RS-232 serial option, it requires a parallel interface. Generally, this is provided by a device code 17 controller using PIO. The CPU must handle each character using such PIO instructions as DOA or SKPBZ resulting in a slower rate of data transfer than with DMA. Usually, the data is transmitted to the printer in 7 or 8 parallel lines. (Not used on a MARK 3)
LPTP51	Line printers using a device code 51 controller. Same as LPTP. (Not used on a MARK 3)
LPTD	Line printers using a Data General 4060-type multiplexer. Requires the RS-232 serial option but does not have the DMA advantage. **NOTE** POINT 4's GUIDE.LPT does not provide instructions for the installation of an LPTD driver. A listing of the LPTD driver file is given in Appendix D.

5.8.7.2 Copy The Line Printer Driver

It is necessary to make a copy of the driver file to ensure that a valid (unmodified) file of that driver remains on the system. POINT 4 recommends that the names of line printer drivers start with "LPT" followed by a digit as some IRIS programs use that form. The name may not include periods, other letters, or symbols. Examples of legal and illegal line printer driver names are as follows:

Legal Name	Illegal Name
LPT	LPTA
LPT1	LPTP
LPT2	LPTP51 (Reserved)
.	LPT.1
.	
.	LPT/3
LPT99	

To copy an IRIS line printer driver, at the system command prompt (#), enter

```
    COPY <00>LPT=LPTM
```

When the line printer file has been copied, the system displays

```
    COPIED!
```

5.8.7.3 GUIDE.LPT

GUIDE.LPT is an interactive BASIC program provided by POINT 4 to assist the user in setting up a line printer. The program asks questions about the line printer and then gives instructions for making the appropriate changes using DSP.

5.8.7.3.1 USING GUIDE.LPT

GUIDE.LPT does not make any changes to the driver file. It provides the information that needs to be entered into the driver file. It will not interfere with any system processes and can be run at any time. If a mistake is made in entering answers to GUIDE.LPT's questions, press <ESC> and restart the program.

5.8.7.3.2 NOTES ON GUIDE.LPT QUESTIONS

GUIDE.LPT asks the user to check certain values in the driver file to make sure it is the correct version.

Consider the following when answering the questions asked by GUIDE.LPT:

- If the driver is LPTP51, answer all the questions as if it were an LPTP (device code 17).

- For LPTP questions on 'DIA' and 'interrupt after any busy', answer NO if you are not sure of what is required. If an inappropriate YES answer is given, the printer may hang up while printing.

- If the printer specifications require a motor-on character in the OPEN list, enter that as the first character in the OPEN list. The motor-off character should be the last character in the CLOSE list.

- For an automatic formfeed on OPEN and CLOSE, enter 14 in both lists. (For a word processing printer, consult the appropriate installation document.)

- The following lists are recommended for a system line printer:

CR LIST:	15 0 12 0 -1
MULTIPLE CR LIST:	0 12 0 -1
DELAY AFTER SPECIAL CHARS:	0 0 0 -1

- The following lists are recommended for a word processing printer:

CR LIST:	15 -1
MULTIPLE CR LIST:	15 -1
DELAY AFTER SPECIAL CHARS:	-1

- Some line printers slash zeros and others slash the letter O. GUIDE.LPT asks

  DO YOU WISH TO PRINT ZERO IN PLACE OF OH AND VICE VERSA?

  Answer YES or NO depending on the line printer and/or requirements.

- When GUIDE.LPT asks

  OUTPUT WHERE?

  Press <RETURN> to display the output on the screen. Write down the output and use the information to modify the driver file.

- After $LPT is functional, run GUIDE.LPT again. When the question 'OUTPUT WHERE?' is asked, press L to get a printout of the line printer configuration.

5.8.7.4 Use DSP To Make Appropriate Changes In The Driver Files

Before using DSP, there are two important things to remember:

1. Always backup the system.
2. Never modify the original line printer drivers. Work with a copy of the driver (see Section 5.8.7.2).

5.8.7.5 Change The Port Definition Table

Changes must be made to the Port Definition Table (PDT) in the appropriate driver file:

- For LPTM - change $MMUX → 33600
- For LPTD - change $DGMX
- For LPTP - change $LPTP

Changes to the PDT consist of:

- Entering the appropriate PCW word in the driver's PDT.
- Setting up a large I/O buffer size for the port (e.g., 500 characters at 9600 baud). A large I/O buffer reduces overhead because line printers have a circular buffer.
- Setting active file size to zero.

5.8.7.6 Enable The Driver

To enable a driver file that has been copied and modified, its name must be preceded by a $-sign. Use the CHANGE command as follows

```
#CHANGE {filename}
NEW NAME: {$filename}
COST = $0.00
NEW COST? <ESC>
#
```

5.8.7.7 Test and Customize The Line Printer

To test the current setup, first do a SHUTDOWN and IPL, then run the following BASIC program:

```
10 OPEN #0,"$LPT"
20 PRINT #0;"ABCD"
30 PRINT "*";
40 GOTO 20
```

If an error results from the OPEN statement, it usually indicates a mistake in the setup. A common mistake is to give the wrong port number to GUIDE.LPT. "Logical, IRIS System Port#" refers to the decimal number of the port assigned to the line printer. This number is always different from the "octal, origin zero" port number for the same physical unit. Port numbers in octal start at zero:

0 - The first possible Mux port

7 - The last port on an 8-port Mux

10 - The first port on the Mux extender

If a serial line printer is used and no real errors result from running the BASIC program but the printer output is wrong or nothing prints, the problem may be caused by either software or hardware.

1. Software - Check the Port Control Word (PCW) for the line printer port in $MMUX.

   a. If the line printer continues printing every time ON or OFF-line is selected, the PCW bit 11 (ready status) may be set incorrectly.

   b. If garbage prints on the line printer or the printer slews the paper, the PCW may have the wrong number of data bits, stop bits, or the wrong parity.

   c. If the asterisks (*) stop printing on the screen, the ready status (bit 11) in the PCW may be incorrectly defined. The asterisks will not resume printing until the printer is able to return to a ready state.

2. Hardware - If asterisks continue printing but no data is output by the printer, then the data is transferred to the Mux port.

   a. Check that the printer is plugged into the proper Mux port.

   b. If the printer requires special jumpers at the printer end of the cable, make sure they have been installed.

Figure 5-9 illustrates some line printer problems. The circled numbers refer to the errors listed below.

Error 1 - An error at the top of the page may mean:

a. Not enough delay characters in the DELAY AFTER SPECIAL CHARS list

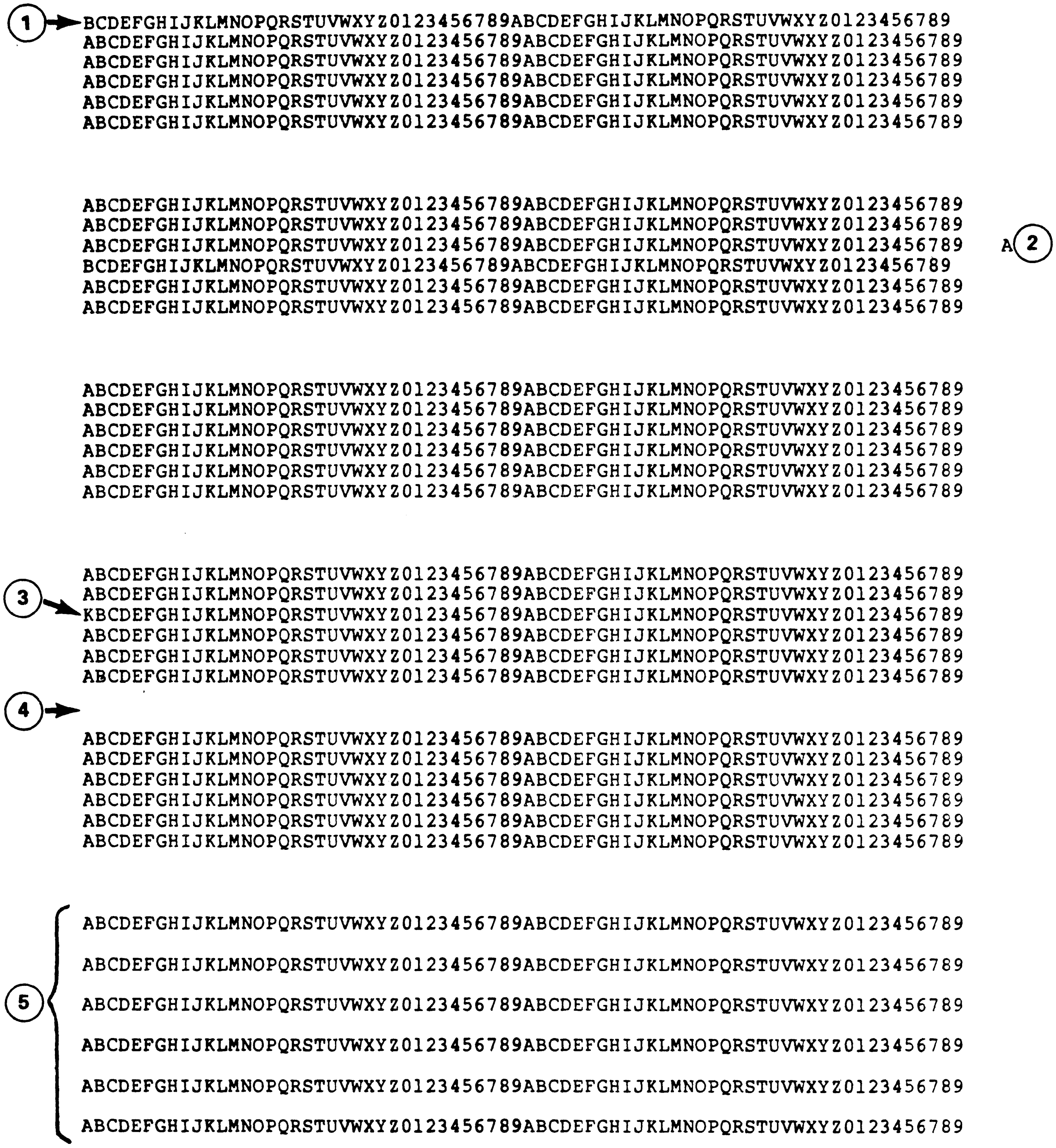
b. If this is the first page to be printed, not enough delay in the OPEN list

Error 2,3 - An error at the beginning or at the end of a line often indicates that there are not enough delay characters in the CR list

Error 4 - Notice that one blank line is missing. This is often caused by not enough delay characters either at the start or at the end of the MULTIPLE CR list

Error 5 - If the whole printout is double spaced, it may be caused by having a 12 (linefeed) in the CR list <u>and</u> having the automatic linefeed option set in the hardware. If that is the case, run GUIDE.LPT again to remove the 12 from the CR list

These problems can be solved by rerunning GUIDE.LPT to make the appropriate changes, and then entering the new values using DSP.



```
BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789


ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789   A
BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789


ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789


ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
KBCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789


ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
```

Figure 5-9. Printer Output

5.8.8 TELETYPE

Two Teletypes may be operated under IRIS. They may use device codes 10/11 and 50/51.

1. Master port Teletype (device code 10/11). It is always port 0. The master port driver resides in REX and remains memory-resident. The address of the driver's Port Definition Table (PDT) is found at location 200 (octal) in REX. The PDT may be modified as described in Section 5.8.1.4.

   A Teletype (device code 10/11) may be used as an interactive port or as a reader/punch. However, it cannot be used for both purposes at the same time (i.e., a tape cannot be read or punched while a user is logged onto the port).

   If the system has a POINT 4 310 Mux, the Teletype must have the RS232 interface and may be used as an interactive device during an IPL and while operating under a minimum configuration IPL.

   The read/punch options on this Teletype are controlled by a Teletype reader/punch driver ($PTM) as a non-interactive device. $PTM does not have a PDT.

   If the paper tape reader/punch driver was loaded as PTM, change its name to $PTM as follows:

   ```
   #CHANGE PTM

   IF NO CHANGE, PRESS RETURN
   NEW NAME? $PTM

   COST = $0.00
   NEW COST? <ESC>
   #
   ```

   When the change procedure is finished, SHUTDOWN and reIPL.

2. A second Teletype (device code 50/51) may be added to the system but it must be used as an interactive device only. It cannot be used as a reader/punch.

   This Teletype is controlled by a secondary Teletype driver ($TTY) which must be made memory-resident. If the driver was loaded as TTY during the sysgen process, change its name as follows:

   ```
       #CHANGE TTY

       IF NO CHANGE, PRESS RETURN.
       NEW NAME? $TTY

       COST = $0.00
       NEW COST? <ESC>
       #
   ```

   The driver's PDT must be set up as described in Section 5.8.1.4.

   To activate the driver, SHUTDOWN the system and reIPL.

5.8.9 HIGH-SPEED PAPER TAPE READER/PUNCH

Under IRIS, the high speed paper tape reader driver is $PTR (device code 12) and the paper tape punch driver is $PTP (device code 13). Neither driver has a Port Definition Table.

To make the reader and/or punch functional, the drivers must be enabled. If the drivers were loaded as $PTR and $PTP during the sysgen process, no further action is required.

If the drivers were loaded as PTR and PTP, change the names to $PTR and $PTP as follows:

```
    #CHANGE PTR

    IF NO CHANGE, PRESS RETURN
    NEW NAME: $PTR

    COST = $0.00
    NEW COST? <ESC>
```

Repeat that procedure for the paper tape punch driver (PTP).

SHUTDOWN the system and reIPL.

5.8.10 MAGNETIC TAPE AND CASSETTE TAPE UNITS

An IRIS installation may operate both magnetic tape and cassette tape units. The magnetic tape subsystem under IRIS supports Data General-compatible magnetic tape (i.e., reel-to-reel) drives and POINT 4-supplied cassette tape units.

The subsystem contains two physical interface drivers:

1. MTAS is the physical interface driver for the magnetic tape system controller. Magnetic tape transfers use the buffer contained within the MTAS file. This buffer replaces the magnetic tape buffer area (TBUF) located in the CONFIG file in earlier IRIS releases. The size of the buffer is 512 bytes. It precedes the ATRIB table. GUIDE.MAGTAPE* may be used to increase the buffer size and to make the appropriate adjustments to MTAS.

2. CTUS is the physical interface driver for the POINT 4 supplied cassette tape unit on a POINT 4 310 MUX. Cassette tape transfers use the buffer defined in the Port Definition Table in $MMUX. A buffer size of 512 bytes plus 25 bytes of data buffer for CTU commands is required.

The system interface driver file is MTA0. It must be copied for each magnetic tape drive and cassette tape unit on the system. These driver files are used by the magnetic tape subsystem to direct the flow of data between a user and a particular magnetic tape drive or cassette tape unit.

To a user, both media are functionally the same since the BASIC commands (such as OPEN, CLOSE, READ, and WRITE) address either the magnetic tape or cassette tape units.

A list of MAGTAPE discsubs is given in Section 5.3.3.1 and CTU discsubs in Section 5.3.3.2. POINT 4 recommends that some discsubs, based on the priority given, be made memory resident to increase system efficiency.

*GUIDE.MAGTAPE will be available in the near future.

5.8.10.1 CONFIGURING A MAGNETIC TAPE DRIVE

If the physical interface driver (MTAS) was not loaded as $MTAS, use the CHANGE command as follows:

```
    #CHANGE MTAS

    IF NO CHANGE, PRESS RETURN
    NEW NAME: $MTAS

    COST = $0.00
    NEW COST? <ESC>
    #
```

Each magnetic tape drive on the system is made functional as follows:

1. Copy the system interface driver with the command

   ```
   COPY MTAn=MTA0
   ```

   where n = the drive number (i.e., 1, 2, 3, etc).

2. Use DSP to enter the following constants into the MTAn driver file:

Location	Enter
ATRIB-2	177777
ATRIB-1	100000+n

   where n = drive 1, 2, 3, etc. corresponding to the driver filename. For the location of ATRIB, refer to Section 5.8.1.

3. To activate the driver, it must be given a $-filename. Use CHANGE as follows:

   ```
   #CHANGE MTAn

   IF NO CHANGE, PRESS RETURN
   NEW NAME: $MTAn

   COST = $0.00
   NEW COST? <ESC>
   #
   ```

4. IPL the system.

5.8.10.2 CONFIGURING A CASSETTE TAPE UNIT

If the physical interface driver (CTUS) was not loaded as $CTUS, use the CHANGE command as follows:

```
#CHANGE CTUS

IF NO CHANGE, PRESS RETURN
NEW NAME: $CTUS
COST = $0.00
NEW COST? <ESC>
#
```

Each cassette tape unit on the system is made functional as follows:

1. Copy the system interface driver with the command

   ```
   COPY CTUn=MTA0
   ```

   where n = the cassette tape unit number (i.e., 0, 1, 2, 3, etc).

2. Use DSP to enter the following constants into the CTUn driver file:

Location	Enter
ATRIB-1	{Port #}

   To find the location of ATRIB, refer to Section 5.8.1.2. Port # is the logical system port number in octal.

   NOTE

   The logical system port number may not be the same as the physical Mux port number on some systems. To find a logical port number, use BASIC's SPC(6).

3. To activate the driver, it must be given a $-filename. Use CHANGE as follows:

   ```
   #CHANGE CTUn

   IF NO CHANGE, PRESS RETURN
   NEW NAME: $CTUn

   COST = $0.00
   NEW COST? <ESC>
   #
   ```

   where n = the cassette tape unit number.

4. Set up the Port Definition Table entry in $MMUX for the CTU port. Refer to Section 5.8.3.

5. IPL the system.

5.9 TERMINAL TRANSLATOR

A Terminal Translation Module ($TERM.name) is the interface between terminal-independent IRIS terminal control functions and a specific type of interactive terminal.

The module is a standard IRIS System Subroutine module ($TERMS) but it does not have an initialization routine.

Terminal Translation Modules are reentrant. Therefore, any number of ports may be linked to a single module.

The system accepts up to 15 enabled $TERM.name modules, but each enabled module must have a unique terminal type code (TTC). Acceptable numbers for TTC range from 1 to 144. A zero indicates that no driver was selected and default processing is desired. All ports are type zero until linked to an enabled module.

TTC is the lower (right) byte of Word 3 in the Port Definition Table (see Section 5.8.1.4).

5.9.1 INSTALLATION OF A TERMINAL TRANSLATION MODULE

Three steps are required to install a TERM.name file:

1. Obtain the correct Terminal Translation Module name for the terminal from the IRIS R8 Peripherals Handbook.

2. Enable the selected Terminal Translation Module as a $ sign file (i.e., $TERM.name). The file type is 77001.

3. Enable the system driver TERMS as $TERMS.

5.9.2 LINKING A TERMINAL TRANSLATION MODULE

A Terminal Translation Module, including the ability to use its corresponding terminal control mnemonics, is activated when the port is linked to it.

The system will link one or more ports at IPL-time after the TTC byte of Word 3 in the Port Definition Table has been modified.

Four steps are required:

1. Obtain the TTC number from the IRIS R8 Peripherals Handbook.

2. Locate each port's RDE cell in the Port's Device Driver File (see Section 5.8.1.4).

3. Use DSP to store the TTC in the lower (right) byte in the RDE cell of each selected port.

NOTE

> The upper (left) byte is reserved by the system for the Port's Return Delay. This delay remains valid after storing a TTC.

4. Shutdown and Re-IPL the system.

5.9.3 LINKING A TERMINAL TRANSLATION MODULE AFTER AN IPL

After an IPL, a Terminal Translation Module may be linked to a port or the linkage may be changed. Obtain the module's port type from the IRIS R8 Peripherals Handbook.

5.9.3.1 Linking to a Port

A Terminal Translation Module can be linked to a port in two ways:

1. From the system manager's account, the port type may be set for any port with the command

```
PORT p  TYPE n
```

   where
   n - port type
   p - port number

2. From a general account, the port type may be set for the port to which the user is logged on with the command

```
PORT TYPE n
```

5.9.3.2 Changing Linkage

When a Terminal Translation Module is linked to a port, characters of less than 200 octal cannot pass directly to the screen. To remove linkage and allow characters to pass directly, enter the command

```
PORT TYPE 0
```

5.10 TIMESHARING

Timesharing is the method by which numerous users and jobs are serviced seemingly simultaneously by the IRIS Operating System. This is accomplished by the system scheduler.

All jobs receive CPU service based on an inverse relation to total system demand. When overall demand is low, more CPU time is allocated to individual jobs. When demand is high, the ratio of CPU time per job is lower.

When more than one user or job requires CPU service, the scheduler apportions CPU time into time slices and assigns these time slices to each user or job based on relative priority and total system demand. The system manager may modify scheduler parameters by setting program and account priorities to maximize system efficiency.

5.10.1 SCHEDULER FUNCTIONS

Functions of the scheduler include the following:

- Establishing relative priority - The scheduler establishes a relative priority for each job and attempts to assign CPU time based on that priority.
- Foreground and background mode - The system manager may stipulate which programs will run in foreground or background.
- Response time and system throughput - The scheduler adapts itself to interactive and compute-bound states to optimize apparent response time and system throughput.

5.10.1.1 Priority

The scheduler establishes a relative priority for each job based on its effective priority and prior service received. A high priority job receives a higher percentage of CPU time by having time slices assigned to it more frequently. It does not receive longer time slices.

When a job begins, its effective priority is calculated based on the following formula:

Effective priority = 2 * account priority + program priority

When a job chains to another program, a new effective priority is calculated.

A user's account priority is determined by the system manager when that account is first created (see Section 5.7). It may be set in the range of 1 (low) to 7 (high).

Program priority has a range of 1 (low) to 7 (high). When a program is first created, its priority is set to 5 by the system. Program priority may be adjusted by the system manager.

The job with the higher effective priority receives more service than a job with a lower effective priority. However, the scheduler keeps track of service received. It compares the effective priority of each job, weights it with prior service received and determines a relative priority level for each. Thus, a low-priority job's relative priority increases as it waits for service.

5.10.1.2 Foreground and Background Mode

A job run in foreground is in a normal timesharing mode and always takes precedence over a background job. A background job is a low level job. It is enqueued to receive a time slice when CPU service is not assigned to a foreground job.

Account priority is not used in the effective priority calculation for background jobs. Program priority determines whether a program runs in foreground or background mode as follows:

```
    priority 7-3 = foreground
    priority 2-1 = background
```

When a job is initiated, it is assigned an effective priority level ranging from 21 (high) to 1 (low) as follows:

```
    Foreground = 21-5*
    Background =  2-1
```

*Priorities 4-3 are not used.

5.10.1.3 Response Time and System Throughput

Response time and system throughput have an inverse relation; the scheduler attempts to minimize apparent response time and maximize system throughput.

1. Apparent response time is the elapsed time starting with a user's request to the system and ending with the system response output at the user's terminal. It consists of:

   a. Wait time - The time required to wait for CPU service. This is controlled by the scheduler because it determines when and how frequently a job receives CPU service (i.e., a time slice) based on the job's relative priority.

   b. Processing time - The time required to formulate a response depends on the program or processor and is outside the scheduler's direct control.

2. System throughput is the amount of productive work accomplished by the system. It consists of:

   a. Productive work - Execution done by a program.

   b. Overhead - This includes nonproductive tasks such as resource management (e.g., swapping) and system level tasks which are system routines not immediately related to servicing a user's needs.

5.10.1.3.1 INTERACTION

In order to minimize apparent response time and maximize throughput, the scheduler uses the concept of interaction.

An interaction is the interval between the initiation of a user request at a terminal and the output of the system's response at the terminal. It consists of all the wait and processing time that make up apparent response time.

An interaction starts when one of the following conditions occurs:

- Input is signalled by pressing the <RETURN> key.

- <ESC> or <CTRL-C> is pressed.

- Automatic resumption of a job after an output is completed (e.g., when a user enters a LIBR command, multiple outputs occur without user intervention).

An interaction ends when the system outputs a response.

An interaction consists of one or more time slices and a job is made up of one or more interactions.

5.10.1.3.2 INTERACTIVE AND COMPUTE-BOUND STATES

A job is in an interactive or a compute-bound state depending on the progress of the interaction. When an interaction begins, the job is in an interactive state. It remains in this interactive state until it receives one second of CPU service or the interaction completes.

An interaction enters a compute-bound state when it has exceeded the interactive limit (i.e., it has received a second of CPU service and the interaction has not completed).

The system, as a whole, is in an interactive state while one or more jobs are in an interactive state. The initiation of a single interaction causes the system, as a whole, to be in an interactive state.

The system, as a whole, is in a compute-bound state when all jobs are compute-bound. Under IRIS, compute-bound is considered CPU-bound and all I/O-bound modes (except character I/O).

When the system changes from a compute-bound to an interactive state, the regnant job, if it has more than a short time slice remaining, has its remaining time reduced to a short time slice.

When the system is in an interactive state, the scheduler assigns each job a short time slice. When the system is compute-bound, jobs receive long time slices. Short time slices are desirable because they create the potential for minimizing apparent response time but long time slices are more efficient because they reduce swapping overhead.

5.10.1.3.3 BIAS FACTORS

Two bias factors are used to reduce wait time and, at the same time, reduce swapping overhead. The scheduler uses these bias factors to increase a job's relative priority and give enhanced service while the job is in an interactive state. Consequently, that job receives service sooner with minimum or no impact on other jobs and the system can revert to a compute-bound state faster.

Depending on the needs of a particular system, bias parameters may be modified by the system manager.

The two bias factors used by the scheduler to adjust relative priority upwards are as follows:

1. Start Interaction Account Priority Bias (SINT.BIAS) - Used as a multiplier to increase the relative priority of a job that is in an interactive state.

2. Start Interaction Bias (SINT.AOI) - Gives an advantage in relative priority to all interactions when they begin.

Either bias factor or a combination of both may be used to increase the relative priority of a job in an interactive state. However, with too much bias, interactive jobs tend to take over the system. Too little bias causes an undesirable increase in swapping overhead.

5.10.2 TIMESHARING PARAMETERS

Timesharing parameters are preset to a default value. To meet the requirements of a particular installation, the system manager may use DSP to modify scheduler parameters, CHANGE to modify a program file's priority, and ACCOUNTUTILITY to set account priorities. The following subsections discuss the location and range of the parameters that may be modified. All values are given in octal.

5.10.2.1 Scheduler Parameters

Scheduler parameters are located in the CONFIG file's System Information Table (see Table 5-2). The parameters are preset to default values. Two parameters may be modified by using DSP:

1. KTSL (time slice parameters) located at 617 in CONFIG and shown in Figure 5-10.

    a. Contents of the long time slice parameter (bits 15-8) are:

UNIT :	Tenths of a second
RANGE :	1-377
DEFAULT:	50

    b. Contents of the short time slice parameter (bits 7-0) are:

UNIT :	Tenths of a second
RANGE :	1-377
DEFAULT:	3

    The recommended minimum for the short time slice is 2 tenths of a second since the actual time slice will then be between one and two tenths.

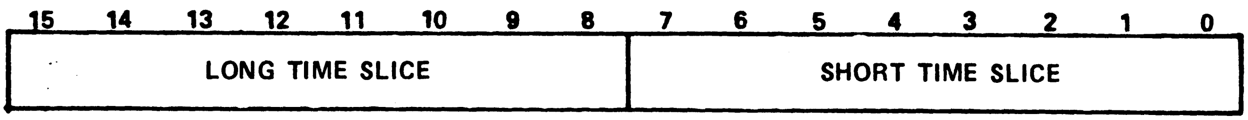


Figure 5-10. Time Slice Parameter Word (KTSL)

2. KSCH (bias factors) located at 621 in CONFIG and shown in Figure 5-11. Two bias factors may be modified. Start Interaction Effective Priority Bias increases the relative priority of a job that is in an interactive state. Start Interaction Bias increases the relative priority of all jobs when they begin an interaction.

   a. Contents of the Start Interaction Effective Priority Bias (SINT.BIAS), bits 10-8, are:

      UNIT   : Octal multiplier
      RANGE   : 1-7
      DEFAULT : 2

   b. Contents of the Start Interaction Bias (SINT.AOI), bits 7-0, are:

      UNIT   : Tenths of a second
      RANGE   : 0-377
      DEFAULT : 144

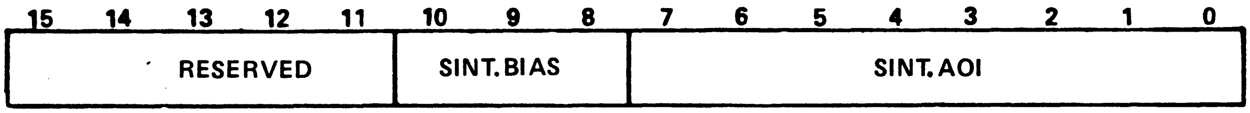


Figure 5-11. Bias Parameter Word (KSCH)

5.10.2.2 Job Priority

Two parameters affect job priority. One is the account priority which is set or modified by the ACCOUNTUTILITY program. The other is program priority. When the program file is created, the system sets a default value of five. It may be modified by the use of the CHANGE command.

1. Account priority consists of:

   UNIT   : Octal number
   RANGE   : 1-7
   DEFAULT : None

2. Program Priority consists of:

   UNIT   : Octal number
   RANGE   : 1-7
   DEFAULT : 5

5.11 PROCESSORS: PASSWORDS AND OPTIONS

Certain processors (system commands) may be restricted by means of passwords or account privilege. Frequently both types of restrictions are used. The first part of this section discusses methods for setting up special passwords to replace the default password which is X.

Certain IRIS processors give access to system files and/or the system configuration. Other processors display information which may be suppressed. Methods for restricting processors to certain ports, times, or accounts and for suppression of the display of information is discussed in Section 5.11.2.

The use of processors that affect configuration are discussed in Section 2 of this manual; the others are discussed in the IRIS Operations Manual and the IRIS User Manual as appropriate.

All changes made to processor files require the use of DSP (see Section 2.3).

5.11.1 PROCESSOR PASSWORDS

Processors like DSP, SHUTDOWN, CLEANUP, etc., allow access to IRIS system files, modification of the system configuration, or shutdown of the system.

The format of a system command is

```
    {filename} <CTRL-E>{key}<CTRL-E>
```

where

```
      key - password assigned by the system manager (the default
            is X)
 <CTRL-E> - disables (or enables) the echo so that the password
            is not visible on the screen
```

A processor password differs from a password given to a user's file in that it is not part of the filename. A processor password may be a string of up to 15 characters and/or numbers. It is contained in the processor file at location 570 (octal).

Use DSP to modify a processor password. At the system command prompt (#), enter

```
    DSP <CTRL-E>{key}<CTRL-E>
    F{filename}
    I570:{password}
    X
```

NOTE

A password must not exceed 15 characters!

Processors that have default passwords are:

Processor	Password Function
CLEANUP	Permits access to the command
DSP	Permits access to the command
KILL	Permits access to extended function: deletion of system files (type 1), driver files (type 36)
PORT	Permits access to extended functions; e.g., eviction of ports and changing port characteristics
SHUTDOWN	Permits access to the command

5.11.2 OPTIONAL PROCESSOR RESTRICTIONS

Several IRIS processors have functions which may be restricted to certain accounts, ports, or times of day by setting flags in their files. Other processors display informational messages that may be modified.

Three options are available for restricting the use of processors or limiting the display of information:

1. Account and Port Restrictions

   INSTALL, PORT, REHASH, REMOVE, and SHUTDOWN are processors which affect the functioning of the system and its configuration. The use of these processors may be restricted to certain accounts and ports.

2. Limited Use of DSP

   DSP makes it possible to change any file; it is strongly recommended that the use of DSP be restricted to the system manager.

3. Modify Display Information

   BYE and CLEANUP display account and/or system information which may be modified.

Instructions for exercising these options are given in the following subsections. The processors appear in alphabetical order for ease of reference.

5.11.2.1 BYE

BYE is the log-on/log-off processor which displays accounting information. At log-on, it usually displays a welcome message which may be customized. At log-off, it sets parity checking on modems.

BYE may be extended to start a BASIC program automatically for selected accounts. Certain accounts may be restricted to designated ports at certain times of day.

5.11.2.1.1 WELCOME MESSAGE

The welcome message prints at log-on time and may be any string of up to 63 characters. It is contained in the BYE processor file at location 540 (octal). Use DSP to create or modify the welcome message. For example, at the system command prompt (#), enter

```
    DSP <CTRL-E>{key}<CTRL-E>
    FBYE
    I450:<CTRL-Z><CTRL-Z> WELCOME TO "IRIS" TIME SHARING!
    X
```

where <CTRL-Z> will result in a carriage return on output.

NOTE

Do not exceed 63 characters!

5.11.2.1.2 ACCOUNT INFORMATION

The account information normally displayed at log-off consists of:

```
    ACCOUNT ID?  PORT #nn  GROUP n  USER nn

    mmm dd, 1982  hh:mm:ss

    CPU TIME AVAILABLE      - nnnnnn
    CONNECT TIME AVAILABLE  - nnnnnn

    nnnnnn BLOCKS IN USE, nnnnn AVAILABLE ON UNIT #n
```

The account information normally displayed at log-off consists of:

```
    #BYE  GROUP n  USER nn        mmm dd, 1982  hh:mm:ss

    NET ACCRUED CHARGES:    $$$.cc

    CPU TIME USED        n:nn:nn
    CONNECT TIME USED    n:nn:nn

    nnnnnn BLOCKS IN USE, nnnnnn AVAILABLE ON UNIT #n
```

To suppress or modify any or all items of the account information, use DSP to set the appropriate inhibit bit in BYE's Message Flag Word (MSGFL) at location 200 (octal). See Figure 5-12.

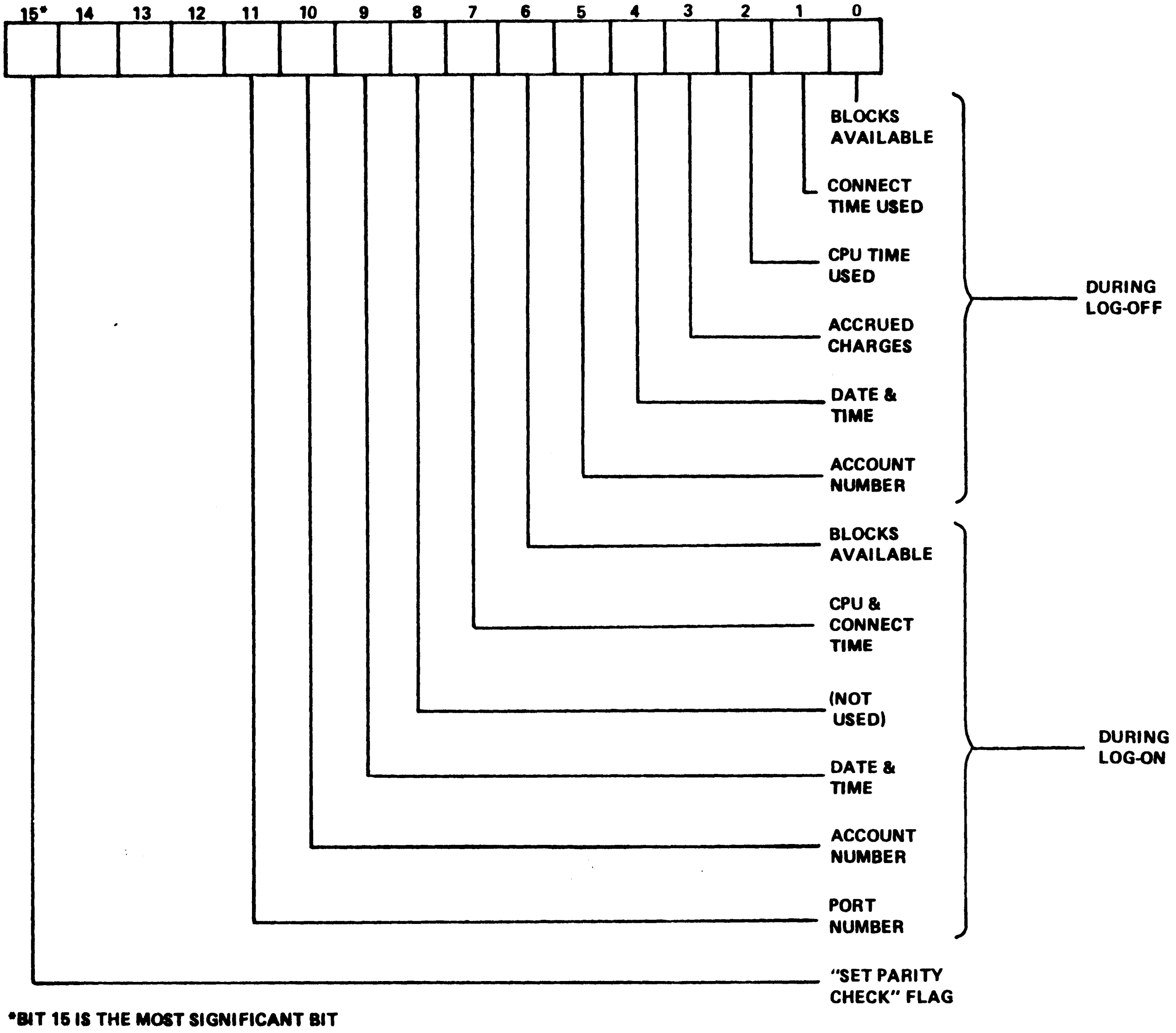


Figure 5-12. Message Flag Word (MSGFL)

5.11.2.1.3 PARITY CHECKING FOR MODEM PORTS

It may be desired that parity checking on a modem port is always in a known state. This avoids problems when different users use the same port. Set bit 15 in BYE's MSGFL word (see Figure 5-12) to 1 at location 200 (octal). This assures that parity checking is set each time a user logs off.

5.11.2.1.4 LOG-ON MESSAGE

BYE may be extended to include a log-on message that will follow the Account Information Display. The message may be any number of lines and each line may contain any number of characters.

The log-on message is contained in the Formatted File "0/LOGONMSG". Each message line is a string item in the file. At log-on, BYE prints each line (string item) found in LOGONMSG until it encounters an end of message (i.e., a null string).

The LOGONMSG file is not supplied with a new IRIS system. Use the FORMAT command to create your own message file. In the following example user input is underlined:

```
    #FORMAT  <33>LOGONMSG
    ITEM #0: S75
    ITEM #1: <RETURN>
```

where S75 creates a normal line length of 75 characters.

A BASIC program is required to place messages into LOGONMSG. The following program may be used to place message lines into the LOGONMSG file:

```
    10 DIM A$(75)
    20 OPEN #1,"LOGONMSG"
    30 INPUT "\215\? "A$
    40 WRITE #1,R;A$
    50 LET R=R+1
    60 IF LEN (A$)>0 GOTO 30
    70 CLOSE #1
```

5.11.2.1.5 LOG-ON RESTRICTIONS

Selected accounts may be restricted to designated ports or certain times of day. These options are controlled by the LOG ON Restriction Table in the CONFIG file (see Section 5.5).

5.11.2.1.6 AUTO PROGRAM START

Selected accounts may have certain BASIC programs started automatically when they log-on and selected initialization programs may be run at IPL-time. Please refer to Section 5.6 for detailed information.

5.11.2.2 CLEANUP

The CLEANUP processor operates in several continuous phases. As each phase begins execution, its number is displayed as a reference point. Some of these phases operate on particular file type groups. As each file is accessed, the name of that file is printed to give an audit trail. In most cases, the audit trail is a desirable feature. If there is a problem, the name of the last file accessed is displayed.

While the phase number display remains, the audit trail may be suppressed. To suppress the audit trail, set location 200 (octal) in the CLEANUP processor file to zero. Any nonzero value in location 200 causes the audit trail to print.

5.11.2.3 DSP

The DSP processor is a powerful tool used to modify system files. POINT 4 recommends that its use be confined to the system manager. A limited use of DSP may be authorized for certain accounts. These accounts may use DSP's F command to access those files which are not protected against them. The G and W commands remain restricted and can be used by the manager account only. All accounts must use the password assigned to DSP.

To give a specific account access to DSP, enter the selected account's number (group,user) in the Authorized Accounts List in DSP. Location 200 contains an address which points to the Authorized Accounts List in the DSP file. The maximum number of entries is 127 and the list must be terminated with an octal zero. One word is used for each account entry. Bits 13-6 contain the group number; bits 5-0 contain the user number.

5.11.2.4 INSTALL

The INSTALL processor gives access to logical units. Its use may be restricted to selected accounts and/or ports.

5.11.2.4.1 ACCOUNT PRIVILEGES

Account privileges are set at location 200 (octal) in the INSTALL processor file; the following options are available:

1. Allow INSTALL from the manager account only (0)

2. Allow INSTALL from all accounts (-1)

3. Allow INSTALL from the manager and one alternate account (p,g,u in standard IRIS Account Word format)

To assign account privileges, at location 200 (octal) in INSTALL set:

```
        0 - Manager account only
       -1 - All accounts
    p,g,u - Manager and one alternate account
```

where

```
    p - Privilege level, set bits 15-14
    g - Group number, set bits 13-6
    u - User number, set bits 5-0
```

5.11.2.4.2 PORT PRIVILEGES

INSTALL allows access from a single designated port or from all ports.

To assign port privileges, at location 201 (octal) in INSTALL set:

```
    p - Allow INSTALL from Port p only
        where p is the logical system port number in octal

   -1 - Allow INSTALL from all ports
```

5.11.2.4.3 INSTALL FAST PRIVILEGES

INSTALL FAST is either enabled or not. If it is enabled, it may be restricted to either the manager account only, or to the manager account and one alternate. Set the account word at location 202 (octal) in INSTALL in one of the following ways:

1. Disallow INSTALL FAST (-1)

2. Allow INSTALL FAST from the manager account only (0)

3. Allow INSTALL FAST from the manager and one alternate account (p,g,u in standard IRIS Account Word format)

To assign account privileges, at location 202 (octal) in INSTALL set:

```
       -1 - not allowed
        0 - Manager account only
    p,g,u - Manager and one alternate account
```

where

```
    p - Privilege level, set bits 15-14
    g - Group number, set bits 13-6
    u - User number, set bits 5-0
```

5.11.2.4.4 QUESTIONABLE FILE HANDLING

While doing its housekeeping, INSTALL may encounter a questionable file. A file is questionable when it contains a damaged header or is in the process of being built (i.e., the build bit is set to 1). INSTALL will handle such a file according to the parameters set at location 203 (octal) in INSTALL:

0 - Retain all questionable files - If the file is in the process of being built, the build bit is reset, the file retained, and INSTALL continues. If the file is damaged, the system halts with the file retained (refer to the IRIS Operations Manual, Section 2).

1 - Retain file being built (build bit is reset), delete a damaged file. INSTALL continues.

2 - Delete all questionable files - Files being built and those which are damaged are deleted. INSTALL continues.

5.11.2.5 PORT

The PORT ALL MONITOR command may be made available to all accounts or it may be restricted to the manager account only.

Set location 200 (octal) in the PORT processor file to one of the following parameters:

Nonzero	- All accounts
0	- Manager only

5.11.2.6 REHASH

The processor REHASH may be restricted either by account or by port.

5.11.2.6.1 ACCOUNT PRIVILEGES

Account privileges are set at location 200 (octal) in the REHASH processor file; the following options are available:

1. Allow REHASH from the manager account only (0)

2. Allow REHASH from all accounts (-1)

3. Allow REHASH from the manager and one alternate account (p,g,u in standard IRIS Account Word format)

To assign account privileges, at location 200 (octal) in REHASH set:

```
        0 - Manager account only
       -1 - All accounts
    p,g,u - Manager and one alternate account
```

where

```
    p - Privilege level, set bits 15-14
    g - Group number, set bits 13-6
    u - User number, set bits 5-0
```

5.11.2.6.2 PORT PRIVILEGES

REHASH allows access from a single designated port or from all ports.

To assign account privileges, at location 201 (octal) in INSTALL set:

```
    p - Allow REHASH from Port p only
        where p is the logical system port number in octal

   -1 - Allow REHASH from all ports
```

5.11.2.7 REMOVE

The processor REMOVE may be restricted either by account or by port.

5.11.2.7.1 ACCOUNT PRIVILEGES

Account privileges are set at location 200 (octal) in the REMOVE processor file; the following options are available:

1. Allow REMOVE from the manager account only (0)
2. Allow REMOVE from all accounts (-1)
3. Allow REMOVE from the manager and one alternate account (p,g,u in standard IRIS Account Word format)

To assign account privileges, at location 200 (octal) in REMOVE set:

```
        0 - Manager account only
       -1 - All accounts
    p,g,u - Manager and one alternate acccount
```

where

```
    p - Privilege level, set bits 15-14
    g - Group number, set bits 13-6
    u - User number, set bits 5-0
```

5.11.2.7.2 PORT PRIVILEGES

REMOVE allows access from a single designated port or from all ports.

To assign account privileges, at location 201 (octal) in INSTALL set:

```
    p - Allow REMOVE from Port p only
        where p is the logical system port number in octal

   -1 - Allow REMOVE from all ports
```

5.11.2.8 SHUTDOWN

The processor SHUTDOWN may be restricted either by account or by port.

5.11.2.8.1 ACCOUNT PRIVILEGES

Account privileges are set at location 200 (octal) in the SHUTDOWN processor file; the following options are available:

1. Allow SHUTDOWN from the manager account only (0)

2. Allow SHUTDOWN from all accounts (-1)

3. Allow SHUTDOWN from the manager and one alternate account (p,g,u in standard IRIS Account Word format)

To assign account privileges, at location 200 (octal) in SHUTDOWN set:

```
        0 - Manager account only
       -1 - All accounts
    p,g,u - Manager and one alternate account
```

For an alternate account, enter the following into location 200:

where

```
    p - Privilege level, set bits 15-14
    g - Group number, set bits 13-6
    u - User number, set bits 5-0
```

5.11.2.8.2 PORT PRIVILEGES

SHUTDOWN allows access from Port Zero and one other designated port or from all ports.

To assign account privileges, at location 201 (octal) in INSTALL set:

```
    p - Allow SHUTDOWN from Port Zero and Port p only
        where p is the logical system port number in octal

   -1 - Allow SHUTDOWN from all ports
```

5.12 BASIC PROGRAM PARTITION REQUIREMENTS

A program partition, also called a user partition, is an area of the CPU main memory which holds a user's BASIC program and its variables, strings, and arrays, while the program is being run. A BASIC program with a large number of statements, large strings, or large arrays requires a large user partition. The user partition area is also used by IRIS processors such as EDIT, COPY, INSTALL, ASSEMBLE, and LIBR's sort option.

Prior to R8, IRIS used multiple fixed-size partitions. That meant, if there were more users on the system than the number of user partitions available, IRIS would save the contents of the program's partition onto disc and replace it with the next user's partition information read from disc. The area on disc used to hold a user's partition information is called the "active file". Each user has an active file located on LU/0.

The process of saving or 'rolling out' one user and 'rolling in' another user from the disc into a program partition is called 'swapping'. Ideally, swapping should be kept to a minimum because it moves the disc's read/write heads away from the data area and involves the transfer of a large number of disc blocks. A reduction in the amount of swapping needed results in the improvement of system response and throughput.

IRIS R8 offers Dynamic Partitioning instead of the multiple fixed-size partitions. Dynamic Partitioning means that there is one combined partition area shared by all users and swapping is reduced to a minimum. The main benefits of Dynamic Partitioning are:

- Large programs may be expanded even to the point of filling the entire combined partition area
- Smaller programs, a number of which fit into the combined partition area, will run without swapping

Under IRIS, two steps are required when configuring a system:

1. Set Partition Size (PSIZ) - the size of the combined partition area.
2. Set the proper active file size for each interactive port.

5.12.1 DETERMINING PSIZ

PSIZ is set at location 400 in the CONFIG file. The minimum value of PSIZ is 10000 (octal), the maximum value is 77400 (octal) depending on the size of memory (32KW or 64KW).

If a BASIC program gets an error 3 indicating program overflow, comments may be deleted from the BASIC program (REMs and ! comments take up program space) or PSIZ may be increased (must be followed by an IPL).

The minimum partition size required for a system is determined by the size of the largest BASIC program (see A below), or by the combined size of the most frequently run programs (see B below).

A. The method for setting PSIZ based on the largest BASIC program is as follows:

   1. Run the largest BASIC program so that all strings and arrays are dimensioned.

   2. After all the DIM statements have been executed, press

      <ESC>

   3. Under BASIC, enter the command

      SIZE

      The total program size (in decimal) is displayed.

   4. Add 30 (decimal) to the size generated in step 3 to adjust for the work space required by some BASIC statements.

   5. See Table 5-10 for the corresponding octal value for PSIZ.

B. If a number of smaller BASIC programs will be run frequently, it is advisable to choose a PSIZ which allows those programs to use the partition area concurrently.

   Use the procedure given in A to compute the octal value corresponding to PSIZ for each of the smaller programs. Then add the sizes together to determine their combined size.

TABLE 5-10. PARTITION SIZE SELECTION TABLE

Partition Size PSIZ (Octal)	Maximum BASIC Program Size (Decimal)
2000	719
2400	975
3000	1231
3400	1487
4000	1743
4400	1999
5000	2255
5400	2511
6000	2767
6400	3023
7000	3279
7400	3535
10000	3791
10400	4047
11000	4303
11400	4559
12000	4815
12400	5071
13000	5327
13400	5583
14000	5839
14400	6095
15000	6351
15400	6607
16000	6863
16400	7119
17000	7375
17400	7631
20000	7887
20400	8143
21000	8399
21400	8655
22000	8911
22400	9167
23000	9423
23400	9679
24000	9935
24400	10191
25000	10447
25400	10703
26000	10959
26400	11215
27000	11471
27400	11727
30000	11983
30400	12239
31000	12495
31400	12751
32000	13007
32400	13263
33000	13519
33400	13775
34000	14031
34400	14287
35000	14543
35400	14799
36000	15055
36400	15311
37000	15567
37400	15823
40000	16079

*Minimum value of PSIZ

5.12.1.1 PSIZ For 64K-Word Memory

On a system with 64K-word memory, the user partition is automatically placed above 32KW. The remaining space, above the user partition area, is used as a buffer pool (see Section 5.13).

Six words of lower memory are required for each buffer in the buffer pool. Decreasing PSIZ in a 64KW system increases the number of buffers in the buffer pool and that reduces the amount of lower memory available.

If PSIZ is substantially reduced, it may result in a Trap 141 on IPL, indicating a lower-memory overflow.

5.12.1.2 PSIZ For 32K-Word Memory

With a POINT 4 MIGHTY MUX ($MMUX), the maximum PSIZ for a 32KW memory is approximately 23000 (octal). If the PSIZ is set too large, IRIS will TRAP on IPL. IRIS then does a minimum configuration IPL automatically, which allows the system manager to use DSP to reduce PSIZ.

A larger PSIZ may be achieved by increasing the port control area (PCA) address in $MMUX in multiples of 1000 (octal) as shown in Figure 5-13. User input is underlined. See also Section 5.8.1.2 (ATRIB Table).

Command	Description
#DSP <CTRL-E>{key}<CTRL-E>	Where key is the password
F$MMUX D32201 <ESC>	Find $MMUX Dump location 32201, then press <ESC>. The contents of location 32201 is a pointer to the first Mux port in the port control block.
Dx1	Dump contents of x1 - where x1 is the first word (PCA) in the $MMUX's port control block.
x1: x2+2000 (70000)	Add 2000 (octal) to value x2 in location x1 - where x2 is the content of location x1 (x2 should be 66000). Increases $MMUX's PCA from 66000 to 70000 (sets PSIZ to 2000 (octal) larger).
Dx1	Dump location x1 to check that the change was made correctly.

Figure 5-13. Changing $MMUX PCA with DSP

5.12.2 ACTIVE FILE SIZE

All interactive ports have an active file on LU/0 which is used for swapping. The size of an active file is the number of sectors used for swapping. If the active file is too large, disc space on LU/0 is wasted. If the active file is too small, there will be a significant performance penalty.

When IRIS finds that an active file is too small, it will automatically allocate extra blocks as needed. However, those blocks will not be contiguous to the rest of the blocks in the active file, resulting in slower system performance because of the increased latency and seek time in swapping.

If there is enough disc space available on LU/0, set the active file sizes large enough to hold the entire PSIZ. In that case:

active file size = PSIZ/400 (octal)

If not enough disc space is available on LU/0 to give each port the maximum active file possible, base the size of each active file on the size of the largest BASIC program to be run on that port as follows:

1. Obtain the size of the program as shown in Section 5.12.1.
2. See Table 5-7 and calculate the minimum PSIZ necessary for that program size.
3. Active file size = X/400 (octal).

The active file size for any interactive port is set in word 6 of that port's Port Definition Table. Please refer to Section 5.8.1.4.

WARNING

> Under earlier versions of IRIS, BASIC programmers were accustomed to using Error 3 as a signal that the program was getting too large. They would then remove unnecessary program statements or break the program into segments that would CHAIN to each other.
>
> Under IRIS R8, Error 3 indicates that an overflow of the combined partition area has occurred.
>
> Programmers should be careful not to let a BASIC program expand to the point where it fills all the available space.
>
> POINT 4 recommends that the system manager check the active file sizes periodically to prevent users from running programs that are larger than intended.

5.13 BUFFER POOL

The purpose of a buffer pool is to reduce the number of disc accesses resulting in disc reads or writes and, thus improve system performance. A minimum number of disc accesses per disc block, contingent on the level of data integrity desired, consists of one disc read and one or more disc writes.

The size of the buffer pool is limited by the size of the user partition. Refer to Section 5.12 for information on user partition requirements.

The following subsections explain the uses of the buffer pool, dirty pages, and the trade-off between performance and data integrity.

5.13.1 EXTRANEOUS DISC READS AND WRITES

Extraneous disc reads and writes occur when a disc block in memory is accessed repeatedly. Reading a disc block into a buffer pool eliminates or significantly reduces extraneous reads and considerably improves system performance given the following circumstances:

- A disc block, or a record contained in a disc block, is reused in a relatively short period of time
- A number of different records contained in a disc block are to be accessed
- Directory, index, header, or other system disc blocks are accessed frequently

Extraneous disc writes are avoided by updating the copy in the buffer pool; the block is written only once after all the checking and updating is completed.

5.13.2 DIRTY PAGES

A dirty page is any block in the buffer pool that has been updated in memory but has not been written to disc. A No Dirty Page Flag (NDPF) is used to control the point at which a dirty page is to be written to disc. The flag is set at location 606 (bit 15) in the CONFIG file (see Section 5.14).

5.13.3 DATA INTEGRITY

When a crash occurs on a system without a buffer pool, only the last update is lost. A system with a buffer pool and NDPF = 0, affords the best system performance but, if a crash occurs, an indeterminate number of updates may be lost.

The only reliable method for file recovery if NDPF = 0 is to load the most recent backup copy of the file and reenter the data.

A system with a buffer pool and NDPF = 1 may lose the most recently entered update. It has the same integrity that a system without a buffer pool has, but offers a substantial increase in system performance.

5.14 SPECIAL CONDITIONS FLAG (SPCF)

The special conditions flags are contained in the SPCF word at location 606 (octal) in the CONFIG file. The flags control certain system functions and options. Currently, only bit 14 (BASIC Error Message Flag) and bit 15 (No Dirty Page Flag) are used. All other bits are reserved for future use.

5.14.1 BASIC ERROR MESSAGE FLAG

Any error detected by the IRIS Business BASIC interpreter during program entry or at run-time generates an error code accompanied by descriptive text. The descriptive text may be suppressed by setting the BASIC Error Message Flag (BEMF) in the SPCF word at 606 (octal) in CONFIG.

When BEMF is set to 0 (0 is the default), the descriptive text is printed with the appropriate error code.

When BEMF is set to 1 (40000 octal), the error codes are printed but the descriptive text is suppressed.

5.14.2 NO DIRTY PAGE FLAG

The No Dirty Page Flag (NDPF) controls the point at which a disc block that was read into the buffer pool (and may have been updated) is written back to disc. NDPF is bit 15 in the SPCF word at locaton 606 (octal) in CONFIG.

If NDPF is set to 0 (i.e., the flag is not on), each disc block is read from and written to disc only once resulting in maximum system performance. The system flushes dirty pages to disc if it is idle. If it is busy, dirty pages are written to disc when a buffer pool block is needed and the current contents of a block have not been accessed recently. The most used blocks in the buffer pool stay 'dirty' for several hours or more. Increased system performance is achieved at the expense of data integrity.

If NDPF is set to 1, all writes are forced to disc while extraneous reads are still eliminated. Generally, a greater number of reads than writes are required, thus system performance will show more than half the gain produced by NDPF = 0 without sacrificing data integrity.

5.15 SYSTEM DRIVERS FOR POINT 4 CPUs

The IRIS Operating System includes two system drivers for CPUs; only one may be enabled at any one time.

- $DEC - For MARK 3 and MARK 5-type CPUs
- $MK8 - For the MARK 8 CPU; allows access to the MARK 8 macro instruction capability

All IRIS Operating Systems shipped on disc, diskette, or cassette tape have $DEC enabled. If the CPU is a MARK 3 or a MARK 5, no further action is required. For a MARK 8 CPU, $DEC must be disabled and MK8 enabled. The procedure is as follows:

1. #CHANGE $DEC

```
IF NO CHANGE, PRESS RETURN
NEW NAME:  DEC

COST = $0.00
NEW COST? <ESC>
#
```

2. #CHANGE MK8

```
IF NO CHANGE, PRESS RETURN
NEW NAME:  $MK8

COST = $0.00
NEW COST? <ESC>
#
```

3. SHUTDOWN the system.

4. Re-IPL.

Systems shipped on paper tape require a sysgen as described in Section 3. Load as $DEC for the MARK 3 or MARK 5 and load MK8 (without the $-sign). For a MARK 8 CPU, load DEC without a $-sign and load MK8 with the $-sign ($MK8).

APPENDICES

Appendix A
LOADING SOFTWARE
From Disc, Diskette, or Streamer Tape

This appendix discusses the recommended methods for loading software from disc, diskette, and streamer tape. Software includes stand-alone utility programs, a standard IRIS Operating System, and optional application packages. Each method is discussed in the following subsections:

A.1 - Disc Pack
A.2 - Diskette
A.3 - Streamer Tape

A.1 SOFTWARE SUPPLIED ON DISC PACK

Software supplied on a disc pack includes stand-alone utilities, LU/5 utility programs, a standard IRIS Operating System, and optional application packages. The system should not be loaded or configured directly from the disc pack supplied by POINT 4. It is recommended that the POINT 4-supplied disc pack be copied onto a scratch disc pack on the user's system because

- A disc pack formatted on another system may have a different drive tolerance, head alignment, temperature tolerance, etc.
- The disc pack supplied by POINT 4 should be kept intact so that the original version of the software is always available and may be recopied for future reconfigurations.

A.1.1 COPYING THE POINT 4-SUPPLIED DISC PACK

The procedure for copying the disc pack supplied by POINT 4 is as follows:

1. Format a scratch disc pack.

2. Adjust the temperature of the POINT 4-supplied disc pack by mounting it in the disc drive and letting it spin for approximately 20 minutes.

3. Bring any available disc-to-disc copy program into memory. This may be a program supplied by the disc drive vendor or another operating system.

   If no other disc-to-disc copy program is available, the POINT 4-supplied disc pack may be IPLed in order to load the appropriate disc copy program.

   If the system has a POINT 4 LOTUS 700/710 or a MARK 3 disc controller, a copy program may be loaded from the disc as follows:

   ```
   SHUTDOWN <CTRL-E>X<CTRL-E> DISCUTILITY
   ```

   where X is the default password.

   If the system has a POINT 4 LOTUS 701 disc controller, a DDCOPY program may be loaded from the disc as follows:

   ```
   SHUTDOWN <CTRL-E>X<CTRL-E> DDCOPY.1
   ```

   The DDCOPY.1 program supplied for the LOTUS 701 disc controller is configured to copy removable-to-fixed surfaces. This configuration may be modified to suit the needs of a particular installation.

   As a convenience, POINT 4 provides an alternative to DDCOPY.1. This is the NEW4234FMT program which is not a POINT 4 product. It is interactive and offers a number of options including formatting. The program may be loaded as follows:

   ```
   SHUTDOWN <CTRL-E>X<CTRL-E> NEW4234FMT
   ```

4. Copy POINT 4's disc pack onto the scratch disc pack.

5. Store the POINT 4-supplied disc pack in a safe place. It should not be used except when it is necessary to make a new copy of the software.

A.1.2 CONFIGURING A SYSTEM SUPPLIED ON DISC

The disc pack supplied by POINT 4 contains a standard IRIS Operating System. The scratch pack containing this standard system should be configured to meet the requirements of a particular installation. Configuration includes:

- Enabling required drivers
- Setting up accounts
- Setting up ports
- Installing the appropriate terminal translation modules
- Making certain processors accessible from selected accounts
- Customizing the system by modifying the preset discsub table, adjusting partition requirements, etc.

Refer to Section 5 for information on configuration procedures. The steps required for configuration are as follows:

1. IPL the system from the scratch disc pack.
2. Configure the system.
3. IPL the configured system.
4. Test the system.
5. Make a copy of the configured system disc pack. Use the copy to run the system and keep the disc pack used for the configuration process as a back-up copy.

A.2 SOFTWARE SUPPLIED ON DISKETTES

Software supplied on diskettes is the same as the software supplied on a disc pack. It includes the following:

1. Diskette containing stand-alone programs including FLBOOT, DBUG, and DISCUTILITY.

2. Diskette containing a standard IRIS Operating System.

3. Diskette containing LU/5 utility programs.

4. Optional diskette(s) containing software packages.

With the exception of the diskette containing the loader (FLBOOT) and DISCUTILITY, no POINT 4-supplied diskette should be used to configure or run the system. POINT 4's diskettes should be copied to scratch diskettes formatted on the user's system. This may be done by loading the diskette containing DISCUTILITY into memory (see Section A.2.1) and using the format and copy options contained in the DISCUTILITY program. The copy of the POINT 4-supplied diskettes should be used to configure and customize the system.

A.2.1 LOADING SOFTWARE FROM DISKETTE

The diskette containing FLBOOT, DISCUTILITY, and DBUG must be loaded into memory first. The procedure is as follows:

1. Turn the power switch to the ON position; this loads the MANIP program from firmware.

2. Insert the diskette into floppy disc drive 0.

3. To read the loader block and pass control to DISCUTILITY, enter

   F

   FLBOOT reads the diskette blocks into memory starting at location 0 and ending with location 67777 (FLBOOT will reside at location 70000).

   NOTE

   > A floppy disc drive is a relatively slow device. Allow enough time for the transfer to take place.

4. Use the F (format) command to format a minimum of four scratch diskettes.

5. Copy the POINT 4-supplied diskettes to the formatted scratch diskettes using the copy option of the DISCUTILITY program.

6. Store POINT 4's diskettes in a safe place.

7. Configure the system using the scratch diskettes (see Section A.1.2).

A.2.2 WRITING FROM MEMORY TO DISKETTE

Some installations may wish to keep backup copies of the configured system or copies of a particular program on scratch diskettes. To make a copy of the configured system, DISCUTILITY (see Section 2.5.1) may be used. To make a copy of a particular stand-alone program, FLBOOT may be used as follows:

1. Insert a formatted scratch diskette into drive 1.

2. Shutdown the system to the program to be written out to diskette with FLBOOT as the second program. The command format is

   ```
   SHUTDOWN <CTRL-E>{key}<CTRL-E>{programname} FLBOOT
   ```

   where key is the password assigned to SHUTDOWN (the default is X).

The loader is written as block 1 onto the diskette followed by blocks containing memory locations 0-67777. When the transfer is completed, control is returned to MANIP.

Multiple copies may be made by loading a scratch diskette and jumping to location 70000.

A.3 SOFTWARE SUPPLIED ON STREAMER TAPE

Software supplied on streamer tape for a POINT 4 MARK 3 Computer System is a disc image copy of the software supplied on a disc pack. It is usually supplied on two streamer tape cartridges. The cartridges supplied by POINT 4 cannot be used to configure or run the system. The cartridges must be copied to disc using the restore option of DISCUTILITY before beginning the configuration process.

The cartridge(s) supplied by POINT 4 contain:

1. Stand-alone utility programs including a loader (STBOOT), DBUG and DISCUTILITY

2. Standard IRIS Operating System; LU/5 utility programs; optional software packages may be included or may be on a separate cartridge

Once the cartridges supplied by POINT 4 are copied to disc, they should be kept in a safe place and used only to make a new copy of the system, if necessary.

A.3.1 COPYING A STREAMER TAPE CONTAINING SOFTWARE

The streamer tape cartridges containing software may be copied to disc by reading the loader (STBOOT) and DISCUTILITY into memory. This procedure is described in Section A.3.1.1.

A.3.1.1 Loading STBOOT and DISCUTILITY

To read the loader (STBOOT) and DISCUTILITY into memory, use the following procedure.

1. Turn the power switch on the Mini-panel to the ON position. This loads MANIP into memory from hardware.

2. To read the loader block into memory, enter

   H

   Streamer blocks are read into memory starting at location 0 and ending at location 67777. STBOOT is loaded at location 70030. When the transfer is completed, the cartridge rewinds and DISCUTILITY is started automatically at location 2.

3. Format a disc by using the F (format) option of the DISCUTILITY program.

4. Copy the cartridge to the formatted disc using the R (restore) streamer tape-to-disc option of DISCUTILITY.

5. Store the streamer tape cartridge in a safe place.

6. Configure the system as described in Section A.3.2.

A.3.1.2 Writing Memory To Streamer Tape

A backup copy of the configured system may be made using the save option of the DISCUTILITY program.

Some installations may wish to keep backup copies of particular stand-alone programs on streamer tape. To make a copy of a particular stand-alone program, STBOOT may be used as follows:

1. Insert a scratch cartridge into the streamer tape unit.

2. Shutdown the system with the command

```
SHUTDOWN <CTRL-E>{key}<CTRL-E>{programname} STBOOT
```

The loader is written as block 1 on the cartridge followed by blocks containing memory locations 0-67777. The cartridge rewinds automatically and control is returned to MANIP.

Multiple copies may be made by loading a scratch cartridge and jumping to location 70030.

A.3.2 CONFIGURING A SYSTEM SUPPLIED ON STREAMER TAPE

The software supplied by POINT 4 on streamer tape contains the standard IRIS Operating System. This standard system must be configured to meet the requirements of a particular installation. A streamer tape may not be used to configure the system. It must be copied to disc. Once this is done, the configuration process is the same as if the software had been received on disc. Refer to Section A.1.2 for the recommended procedure.

Appendix B
SYSGEN LOG

This appendix provides a sample Sysgen Log for use with the paper tape system generation described in Section 3. The Sysgen Log shipped with an IRIS Operating System may be more current than the example given here.

IRIS R 8.1 SYSGEN LOG

Company Name	______________________	Disc controller	________________
Address	______________________	Disc drive	________________
City, State	______________________	Performed by	________________
Telephone	(______)_______-___________	Date	________________

Revised: March 1, 1983

SYSGEN PREPARATIONS

The SYSGEN LOG serves as a checklist and a record of the system generation procedure. Fill out the SYSGEN LOG as the sysgen proceeds and save for later reference. The sysgen procedure steps given here are a summary of Section 3, SYSTEM GENERATION PROCEDURE, of the IRIS R8 Installation and Configuration Manual. References are to individual sections in that manual. Where there are differences, the SYSGEN LOG supersedes the manual.

=======> If using a POINT 4 Mux with the Master Terminal Mode option enabled, set the master terminal baud rate switch on the mux to the desired baud rate.

HARDWARE DIAGNOSTICS

=======> Run hardware diagnostics (See 3.2).

______ CPU Exerciser
______ Power Fail Auto Restart Test
______ Memory Address Test (all memory)
______ Memory Checkerboard Test (all memory)
______ Disc Reliability Test
______ POINT 4 Multiplexer Test (including Q-Test)

INITIAL LOADING OF MEMORY

=======> On a POINT 4 computer use the virtual console's 'Rx' command to load DBUG into memory. (See 3.3.1.1)

```
     Enter:  Rx<RETURN>     to read DBUG

             x = 0 for Teletype reader, device code 10
             x = 1 for high-speed reader, device code 12
```

=======> On a non-POINT 4 CPU, use the BINARY LOADER to load DBUG (See 3.3.1.2).

=======> Enter DBUG and use DBUG's 'K' command to initialize memory. (See 3.3.2)

```
     Enter:  J73000<RETURN>             to enter DBUG from POINT 4
                                        virtual console

     Enter:  K0,72777,77377<RETURN>     to initialize memory
```

=======> Use DBUG's 'Rx' command to load the following tapes into memory. (See 3.3.3)

```
     Enter:  Rx<RETURN>     to read each tape

             x = 0  for Teletype reader, device code 10
             x = 1  for high-speed reader, device code 12
```

Filename	Type	Asm Date	Pun Date	Comments
DBUG	-	________	________	(Loaded above, but enter dates)
BTUP	-	________	________	______________________________
REX	-	________	________	______________________________
SYSGEN	-	________	________	______________________________
SOV	-	________	________	______________________________
BZUD	-	________	________	______________________________

SYSTEM LOADER

=======> Use DBUG's 'J' command to start the SYSTEM LOADER, and load the following tapes with the filenames and types given (See 3.4).

Enter: J50000<RETURN> to start SYSTEM LOADER

DISCSUBS #1	-	________	________	______________________________
SCOPE	33400	________	________	______________________________
BYE	33400	________	________	______________________________
DSP	33400	________	________	______________________________
FAULTPRINT	33400	________	________	______________________________
CONFIG	77001	________	________	______________________________
$MMUX	77001	________	________	Note 1________________________
$DGMX	77001	________	________	Note 1________________________
$RTC	77001	________	________	Note 1________________________
$PTR	36	________	________	Note 1________________________
$PTM	36	________	________	Note 1________________________

Note 1: Load all '$' files but enable only those files (by using the $ prefix) that the system requires (See 3.4.3).

SYSTEM INITIALIZATION ROUTINE

=======> Press <CTRL-C>. This will automatically start the first IPL. Respond to the "PRESS RETURN" message with a 2. No <RETURN> is required. The system will come up in a minimum configuration and only the master terminal (device 10/11) will be active (See 3.4.4)

INITIAL SYSTEM CONFIGURATION

=======> Press <ESC> and log on as MANAGER. Use DSP to set the initial configuration parameters in the CONFIG file and define all multiplexer ports in the appropriate multiplexer driver (See 3.6).

LOADING DISCSUBS

=======> Do a full configuration IPL; respond to the message with a 1 (See 3.7).

DO NOT CONTINUE if the IPL is NOT successful. it must complete without any traps and the CARRY indicator should be flashing at a 1Hz (once per second) rate which indicates that the system is configured correctly and the real-time clock is functioning.

=======> Press <ESC> and log on as MANAGER then enter DSP. Use DSP's 'F' command to select the DISCSUBS file then use the 'R' command to read each of the following components into the DISCSUBS file.

```
          Enter:  FDISCSUBS<RETURN>     to select the DISCSUBS file
                  R<RETURN>             to read each tape

                  DO NOT enter R0 or R1
```

```
DISCSUBS #2  -      _______  ________  ________________________________
DISCSUBS #3  -      _______  ________  ________________________________
DISCSUBS #4  -      _______  ________  ________________________________
DISCSUBS #5  -      _______  ________  ________________________________
DISCSUBS #6  -      _______  ________  ________________________________
```

=======> Exit DSP with the 'X' command and log off.

LOADING DRIVERS AND PROCESSORS

=======> Do a minimum configuration IPL, respond to the message with a 2. Press <ESC> and log on as MANAGER. Use PLOAD to load the following tapes (See 3.8).

Enter: PLOAD<RETURN> to start PLOAD

$PHA	77001			Note 2
$TTY	77001			Note 2
$PTP	36			Note 2
$CTUS	77001			Note 2
$DEC	77001			Required by system
LPTM	36			Note 3
LPTP	36			Note 3
LPTD	36			Note 3
$MTA0	36			Note 2
$MTAS	77001			Note 2
$TERMS	77001			Note 2
$TERM.______	77001			Note 2
$TERM.______	77001			Note 2
$TERM.______	77001			Note 2, Note 4
$FOREIGN	36			
$LUSR.MFP	77001			
MK8	77001			
$LCM	77001			
LCMACTIVATE	33401			
LCMDEACTIVATE	33401			
BCONVERT	33401			
CHANGE	33401			
CLEANUP	77401			
CLEANUPX	77401			
COPY	33401			
FORMAT	33401			
INSTALL	33401			
KILL	33401			
MAIL	33401			
MESSAGES	77001			
PORT	33401			
QUERY	33401			
REHASH	77401			
REMOVE	33401			
SAVE	33401			
PROTECT	33401			
VERIFY	33401			
SHUTDOWN	33403			
LIBR	33401			
ASSEMBLE	33401			Filename ASM optional
EDIT	33401			
RUNMAT	33402			
BASIC	33702			
RUN	33602			

Note 2: Load all '$' files, but enable only those files (by using the $ prefix) that the system requires (See 3.8).

Note 3: Select the appropriate line printer driver for the system and reload as $LPT (See 3.8 and 5.8.7.1).

Note 4: Use the Additional IRIS Components section to record additional $TERM._____ files.

LOADING UTILITIES AND TEXT FILES

=======> Use SHUTDOWN to shutdown the system. IPL into a full configuration, respond to the message with a 1. Press <ESC> and log on as UTILITY. Link BASIC, RUN, and RUNMAT as follows (See 3.9).

```
Enter:  BASIC<RETURN>
        SIZE<RETURN>
        EXIT<RETURN>
```

=======> Load each of the following BASIC programs and SAVE each program using the name and protection given below (See 3.9.1). Use $PTM in place of $PTR for Teletype.

```
Enter:  BASIC<RETURN>
        NEW<RETURN>
        LOAD $PTR<RETURN>
        EXIT<RETURN>
        SAVE <pp>{filename}<RETURN>
```

ACTUTIL.1	pp=77	________	________	________________________________
ACCOUNTUTILITY	pp=77	________	________	________________________________
BASICTEST	pp=77	________	________	________________________________
BUILDXF	pp=33	________	________	________________________________
BUILDPFERR	pp=33	________	________	________________________________
BUILDPF	pp=33	________	________	________________________________
QUERYPF	pp=33	________	________	________________________________
KILLPF	pp=33	________	________	________________________________
ANALYPF	pp=33	________	________	________________________________
EXTRAPORT	pp=77	________	________	________________________________
GUIDE	pp=77	________	________	________________________________
GUIDE.LPT	pp=77	________	________	________________________________
GUIDE.LU	pp=77	________	________	________________________________
GUIDE.BLOCKCPY	pp=77	________	________	________________________________
RETRY	pp=77	________	________	________________________________
SETTIME	pp=33	________	________	________________________________
R7TO8ACTCONV	pp=77	________	________	________________________________
U.CONVERT	pp=33	________	________	________________________________
U.CONVERT1	pp=33	________	________	________________________________
CHECKSUM	pp=77	________	________	________________________________
TRANSMIT	pp=33	________	________	________________________________
RECEIVE	pp=33	________	________	________________________________
LCMC	pp=77	________	________	________________________________
LCMC1	pp=77	________	________	________________________________
LCMC2	pp=77	________	________	________________________________
LCMC3	pp=77	________	________	________________________________
ALOAD	pp=77	________	________	________________________________
MAKEBIN	pp=77	________	________	________________________________
MAKEHEX	pp=77	________	________	________________________________
SWAPTEST	pp=33	________	________	________________________________

=======> Use COPY to load the following stand-alone binary files (See 3.9.2) Use $PTM in place of $PTR for Teletype.

```
Enter:  COPY {file name}*A = $PTR<RETURN>
```

DISCDIAG - ________ ________ (not supplied by POINT 4)________

DISCFORMAT - ________ ________ (not supplied by POINT 4)________

DISCUTILITY - ________ ________ ____________________________

DDCOPY - ________ ________ ____________________________

BLOCKCOPY - ________ ________ ____________________________

MUX310DP - ________ ________ ____________________________

DC700 - ________ ________ ____________________________

LCMDIAG1.3 - ________ ________ ____________________________

M8EXERCISER - ________ ________ ____________________________

=======> Use COPY to load the following text files (See 3.9.3). Use $PTM in place of $PTR for Teletype.

```
Enter:  COPY {file name}*T = $PTR<RETURN>
```

PZ - ________ ________ ____________________________

DEFS - ________ ________ ____________________________

SYMBOLS - ________ ________ ____________________________

L.LCMDEFS - ________ ________ ____________________________

=======> Use FORMAT to create the USERID file as follows (See 3.9.4).

```
Enter:  FORMAT USERID<RETURN>
        D2,S14<RETURN>
        <RETURN>
```

OPTIONAL IRIS SYSTEM COMPONENTS

In addition to the Standard IRIS System Components recorded above, several optional IRIS System Components may be supplied with the system. Installation of such components is covered in a separate document. However, for a complete record of the sysgen, record those components here.

Additional IRIS System Components

__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________
__________ ____ ________ ________ ______________________________

FINAL CONFIGURATION (See 3.10)

At this point complete the system configuration. This includes creating the user accounts; driver setup for line printers, phantom ports, terminal translator; defining processor passwords, options, restrictions; etc.

SYSTEM DISC BACKUP (See 3.11)

Backup the system and create a master copy. Label this master copy and keep it separate from the normal backup rotation.

MANUALS RECEIVED

Quan.	Revision	Date	Description

Appendix C
SOFTWARE DEFINITIONS

This appendix contains a listing of software definitions for IRIS R8.1.

```
ASM,@$LPT,R81DEFSX,R81PZE,R81MTDEFSSB,R81CRTDEFSB
JAN 19, 1983  16:13:02

; SOFTWARE DEFINITIONS FOR "IRIS" R8.1
; 14 JAN 83

;                  All Rights Reserved
;        Copyright (C) 1974, Educational Data Systems
;        Copyright (C) 1980, Educational Data Systems
;        Copyright (C) 1983, POINT 4 Data Corporation
;         This document may not be reproduced without the
;      prior written permission of POINT 4 Data Corporation.

; - - - - - - - - - - - CONTENTS (assumes this is p. 1) - - - - - - - - - -
; p. 1 Miscellaneous              p. 9-11 Discsubs
;  2-3 INFO Table                      12 System subroutines, Accounts
;  4-6 PCB (Port Control Block)        13 File Header block
;    7 User partition, Task Queue      14 DATAPUMP
;    8 TCN (Task Control Node)      15-16 Disc addresses, LUT, LUFIX, LUVAR

; MISCELLANEOUS DEFINITIONS

             400 .DUSR K      =400    ;BYTE SWAP CONSTANT
          100010 .DUSR NOP    =100010 ;NO OPERATION
          136310 .DUSR SAKEY  =136310 ;SAFETY KEY
          177770 .DUSR CHM1   =-10    ;DISP. TO CHANNEL -1 (PROCESSOR) *
          177760 .DUSR CHM2   =CHM1*2 ;DISP. TO CHANNEL -2 (FOR DSP)
          177750 .DUSR CHM3   =CHM1*3 ;DISP. TO CHANNEL -3 (PORT OUTPUT)
          177740 .DUSR CHM4   =CHM1*4 ;DISP. TO CHANNEL -4 (PORT INPUT)

                 ; * MODIFY FOFC1, CHAN2, & CHCHX IN REX IF CHM1 CHANGES

              12       .RDX 10
            3674 .DUSR BASEY =1980    ;BASE YEAR FOR SYSTEM TIME
              10       .RDX 8

; SPECIAL FIXED CORE LOCATIONS

             600 .DUSR INFO   =600     ;SYSTEM INFORMATION TABLE
           32200 .DUSR BPS    =32200   ;BEGINNING OF PROCESSOR STORAGE
           42600 .DUSR MEPS   =42600   ;MINIMUM END OF PROCESSOR STORAGE
           36000 .DUSR LSIR   =36000   ;LOCATION FOR SIR (MIN IS BPS+3600)
           50000 .DUSR LSYSL  =50000   ;LOCATION FOR SYSL
           73000 .DUSR LDBUG  =73000   ;LOCATION FOR DBUG
           76400 .DUSR LBZUD  =76400   ;LOCATION FOR BZUD
           77000 .DUSR LBTUP  =77000   ;LOCATION FOR BTUP

; DEFINITIONS FOR FLAGCHANGE SUBROUTINE

          140000 .DUSR SET    =140000 ;SET FLAG BIT
          100000 .DUSR RESET  =100000 ;RESET FLAG BIT
           40000 .DUSR TOGGL  =040000 ;TOGGLE FLAG BIT
           20000 .DUSR SKIPO  =020000 ;SKIP IF RESULT IS ONE
           10000 .DUSR SKIPZ  =010000 ;SKIP IF RESULT IS ZERO
```

```
- PAGE 2 -

;  INFO TABLE DISPLACEMENTS

                   0 .DUSR SDAT. = 0 ;SYSTEM CREATION DATE (HOURS AFTER BASEYEAR)
                   1 .DUSR SPED. = 1 ;AVG CPU SPEED (INSTR/MSEC)
                   2 .DUSR MILU. = 2 ;MAXIMUM # INSTALLED LOGICAL UNITS
                   3 .DUSR NDCH. = 3 ;NUMBER OF PHYSICAL DATA CHANNELS PER PORT
                   4 .DUSR LPCA. = 4 ;LOCATION OF PORT CONTROL AREA
                   5 .DUSR TNAP. = 5 ;TOTAL NUMBER OF ACTIVE PORTS
                   6 .DUSR SPCF. = 6 ;SPECIAL CONDITION FLAGS

; BITS WITHIN SPCF ARE DEFINED AS FOLLOWS:

              100000 .DUSR NDP  = 100000 ;NO DIRTY PAGES IN BUFFER POOL
               40000 .DUSR SEM  =  40000 ;SUPPRESS ERROR TEXT MESSAGE OUTPUT IN RUN
               20000 .DUSR TDP  =  20000 ;TEMPORARY DIRTY PAGES ALLOWED IN BUFFER POOL

                   7 .DUSR LEPS. = 7 ;LOCATION OF END OF PROCESSOR STORAGE
                  10 .DUSR TOPW. =10 ;TOP WORD OF CORE TO BE USED
                  11 .DUSR ABUF. =11 ;AUXILIARY BUFFER SIZE (NUMBER OF WORDS)
                     ;            12 ;(RESERVED)
                  13 .DUSR NCQN. =13 ;NUMBER OF EXTRA CHARACTER QUEUE NODES
                  14 .DUSR NNOD. =14 ;MINIMUM NUMBER OF FREE NODES
                  15 .DUSR NSIG. =15 ;NUMBER OF SIGNAL BUFFER NODES
                  16 .DUSR NSUB. =16 ;MAXIMUM NUMBER OF DISCSUBS
                  17 .DUSR KTSL. =17 ;TIME SLICE PARAMS, LONG TIME SLICE * 400 + SHORT TIME SLICE
                  20 .DUSR TLOK. =20 ;MINIMUM TIME TO KEEP A PARTITION LOCKED
                  21 .DUSR KSCH. =21 ;SCHED PARAMS, APRI.BIAS*4000 + SINT.BIAS*400 + SINT.AOI

                  32 .DUSR SZICON=32;SIZE OF AREA OF INFO DOWNLOADED FROM CONFIG

;   (CONTINUED NEXT PAGE)
```

```
- PAGE 3 -

; (INFO TABLE CONTINUED)

               32 .DUSR BASY. =32 ;BASE YEAR FOR SYSTEM TIME
               33 .DUSR .TSA. =33 ;TEMPORARY STORAGE "A" POINTER (6 WORDS)
               34 .DUSR .TSB. =34 ;TEMPORARY STORAGE "B" POINTER (6 WORDS)
               35 .DUSR .TSQ. =35 ;TEMPORARY STORAGE "Q" POINTER (6 WORDS)
               36 .DUSR .TSZ. =36 ;TEMPORARY STORAGE "Z" POINTER (6 WORDS)
               37 .DUSR .TSC. =37 ;TEMPORARY STORAGE "C" POINTER (16 WORDS)
               40 .DUSR HRS.  =40 ;CPU TIME - HOURS SINCE JAN 1 OF BASE YEAR
               41 .DUSR TSC.  =41 ;    PART OF HOUR IN TENTH-SECONDS
               42 .DUSR CPLU. =42 ;CURRENT PROCESSOR LOGICAL UNIT
               43 .DUSR CPDA. =43 ;CURRENT PROCESSOR DISC ADDRESS
               44 .DUSR CPTN. =44 ;CURRENT PROCESSOR TYPE NUMBER
               45 .DUSR SDFT. =45 ;SIZE OF EACH PORT'S DATA FILE TABLE
               46 .DUSR DSCO. =46 ;DISC ADDRESS OF "SCOPE"
               47 .DUSR DBYE. =47 ;DISC ADDRESS OF "BYE"
               50 .DUSR DDSP. =50 ;DISC ADDRESS OF "DSP"
               51 .DUSR DSUB. =51 ;DISC ADDRESS OF "DISCSUBS"
               52 .DUSR DMSG. =52 ;DISC ADDRESS OF "MESSAGES"
               53 .DUSR DCON. =53 ;DISC ADDRESS OF "CONFIG"
               54 .DUSR .LSR. =54 ;POINTER TO LOADUSER
               55 .DUSR MAXB. =55 ;MAX # OF BLOCKS IN PARTITION (SET BY SIR)
               56 .DUSR .STN. =56 ;POINTER TO SCHEDULER TASK NODE
               57 .DUSR ..STK=57  ;POINTER TO "CALL" STACK POINTER
               60 .DUSR .RGS. =60 ;POINTER TO REGISTER BUFFER FOR "CALL"
               61 .DUSR FNOD. =61 ;CURRENT # OF FREENODES + 1 ON FREENODE CHAIN
               62 .DUSR .RCV. =62 ;POINTER TO RECOVER ROUTINE
               63 .DUSR .LUT. =63 ;POINTER TO LOGICAL UNIT TABLE (SEE PG 13)
               64 .DUSR SCON. =64 ;CURRENT SYSTEM CONDITION I.E. SYSTEM STATE

; BITS WITHIN SCON ARE DEFINED AS FOLLOWS (NON-EXCLUSIVE):

           100000 .DUSR SR = 100000 ;SYSTEM IS WITHIN SIR
            40000 .DUSR MC = 040000 ;SYSTEM IS OPERATING UNDER MINIMUM CONFIGURATION
            20000 .DUSR SY = 020000 ;SYSTEM IS IN SYSGEN MODE
            10000 .DUSR FL = 010000 ;SYSTEM IS FAULTING
             4000 .DUSR ID = 004000 ;I/O DONE (INTERACTIVE PHASE) -- TIME SHARING ALGORITHM
             2000 .DUSR LC = 002000 ;$LCM IS ACTIVE ON SYSTEM
             1000 .DUSR M8 = 001000 ;$MK8 IS ACTIVE ON SYSTEM
              400 .DUSR M3 = 000400 ;RUNNING ON A POINT 4 MARK 3 CPU
              200 .DUSR RO = 000200 ;READ-ONLY MODE (DISC WRITE COMMANDS ARE IGNORED)
              100 .DUSR CT = 000100 ;$CTR (IE. MONITORING) IS ACTIVE

               65 .DUSR IRUPT=65  ;INTERRUPT FLAG: -1 = NORMAL, >=0 = INTERRUPT BEING SERVICED
               66 .DUSR .BPT. =66 ;POINTER TO BUFFER POOL TABLE
               67 .DUSR .CTT. =67 ;POINTER TO "CALL" TRANSLATE TABLE
               70 .DUSR .SGB. =70 ;POINTER TO SIGNAL BUFFER POINTERS
               71 .DUSR .TTT. =71 ;POINTER TO TERMINAL TYPE TABLE
               72 .DUSR .UPT. =72 ;POINTER TO USER PARTITION (SEE PG 7)
               73 .DUSR THTC. =73 ;TEN HERTZ TASK COUNTER
               74 .DUSR MASK. =74 ;INITIAL INTERRUPT MASK
               75 .DUSR PFRF. =75 ;POWER FAIL FLAG (BIT 15 IS SHUTDOWN FLAG)
                  ;      76-105 ;(RESERVED)
              106 .DUSR BPSP. =106;BEGIN PATCH SPACE (AFTER LAST PATCH)
              107 .DUSR ENDP. =107;END OF PATCH SPACE (CHANGED BY SOV, SIR)
              110 .DUSR INVT. =110;INTERRUPT VECTOR TABLE STARTS HERE
```

```
; PORT CONTROL BLOCK (PCB) DISPLACEMENTS

                 0 .DUSR ICW.  = 0 ;INPUT CONTROL WORD
                 1 .DUSR OCW.  = 1 ;OUTPUT CONTROL WORD
                 2 .DUSR FBA.  = 2 ;FIRST BYTE ADDRESS OF I/O BUFFER -1
                 3 .DUSR LBA.  = 3 ;LAST BYTE ADDRESS OF I/O BUFFER
                 4 .DUSR IBP.  = 4 ;INPUT BYTE POINTER
                 5 .DUSR OBP.  = 5 ;OUTPUT BYTE POINTER
                 6 .DUSR LIB.  = 6 ;LAST INPUT BYTE POINTER
                 7 .DUSR LOB.  = 7 ;LAST OUTPUT BYTE POINTER
                10 .DUSR ACT   =10 ;ACCOUNT NUMBER, PRIVILEGE LEVEL
                11 .DUSR TOB.  =11 ;TEMPORARY OUTPUT CHARACTER BUFFER
                12 .DUSR FLW   =12 ;FLAG WORD (SEE BELOW)
                13 .DUSR ULU.  =13 ;USER'S ASSIGNED LOGICAL UNIT NUMBER
                14 .DUSR URA.  =14 ;USER'S RETURN ADDRESS
                15 .DUSR ORA.  =15 ;OLD RETURN ADDRESS
                16 .DUSR RUA.  =16 ;CPU TENTH-SECONDS USED SINCE LOG-ON (LSB'S)
                   ;     RUAO  =17 ;CPU TENTH-SECONDS USED SINCE LOG-ON (MSB'S)
                20 .DUSR RUI.  =20 ;RESOURCE UNITS USED DURING INTERCTION
                21 .DUSR AOI.  =21 ;AGE OF INTERACTION (TENTH-SECONDS)
                22 .DUSR CTC.  =22 ;CRITICAL TIME COUNTER
                23 .DUSR PF1.  =23 ;APRI*1000 + EFFECTIVE PRIORITY
                   ;               ;RESERVED
                25 .DUSR DFT.  =25 ;POINTER TO DATA FILE TABLE (SEE PAGE 6)
                26 .DUSR PDC.  =26 ;PAUSE DELAY COUNTER (TENTH-SECONDS)
                27 .DUSR AHA.  =27 ;ACTIVE FILE HEADER DISC ADDRESS
                30 .DUSR TON.  =30 ;CPU TIME AT LOG-ON (MINUTES)
                31 .DUSR NLP.  =31 ;NODE LINK POINTER FOR $TERM STORAGE
                32 .DUSR SND.  =32 ;POINTER TO DRIVER'S "SEND" SUBROUTINE
                33 .DUSR OCC.  =33 ;OUTPUT COLUMN COUNTER
                34 .DUSR ODC.  =34 ;OUTPUT DELAY COUNTER
                35 .DUSR RDE.  =35 ;RETURN DELAY, EOM OR TERMINAL TYPE CODE
                36 .DUSR PCW.  =36 ;PORT CONTROL WORD FOR $MMUX (SEE PAGE 5)
                37 .DUSR TTN.  =37 ;TERMINAL TYPE NUMBER & FLAGS (SEE PAGE 5)

; EACH BIT IN FLW IS A FLAG AS FOLLOWS:

;   BIT*    MEANING
;    15_ BINARY INPUT/OUTPUT MODE (PASS BYTE AS IS)
;    14¯ OUTPUT IS PAUSED (CTRL S)
;    13  DSP BREAKPOINT IS SET
;    12_ DSP IS ACTIVE ON THIS PORT
;    11¯ SIGNAL WILL ACTIVATE FROM PAUSE
;    10  A BREAK HAS BEEN DETECTED
;     9_ (RESERVED)
;     8¯ LAST CHARACTER ENTERED WAS CTRL Y
;     7  OUTPUT IS ACTIVE
;     6_ INPUT IS ACTIVE
;     5¯ LOG OFF AFTER PAUSE DELAY
;     4  IGNORE CTRL E & CTRL O (LOG-ON MODE)
;     3_ ACTIVATE ON ANY CONTROL CHARACTER
;     2¯ ENABLE XOFF AND XON
;     1  TRANSPARENT CTRL-E (TOGGLES ECHO BUT IS NOT PUT IN IOB)
;     0_ ECHO INPUT CHARACTERS

; * NOTE:  BIT 15 IS THE MOST SIGNIFICANT BIT
```

```
- PAGE 5 -

; EACH BIT IN PCW IS A FLAG AS FOLLOWS:

;  BIT*    MEANING
;    15_ O
;    14¯ THIS PORT IS ON A POINT 4 MIGHTY-MUX
;    13  O
;    12_ DEVICE CONTROL OUTPUT (1 = HIGH, O = LOW)
;    11¯ NORMAL DEVICE STATUS INPUT (1 = HIGH, O = LOW)
;    10  THIS IS A PHANTOM PORT
;     9_ AUTO LOG-OFF ENABLED
;     8¯ AUTO FREQUENCY SCAN ENABLED
;     7  INHIBIT PARITY CHECK AND GENERATION
;     6_ TWO STOP BITS (NORMAL = ONE)
;     5¯ \  CHARACTER LENGTH: 11 = 8 BITS, 10 = 7 BITS
;     4  /                    01 = 6 BITS, 00 = 5 BITS
;     3_ EVEN PARITY (IF ENABLED)
;     2¯ \  CURRENT BAUD RATE:
;     1   }  7 = 9600, 6 = 4800, 5 = 2400, 4 = 1200,
;     0_ /    3 = 600,  2 = 300,  1 = 150,  0 = 110.


; FORMAT OF TTN WORD, USED IN $TERMS:

;  BIT*    MEANING
;    15_ SPECIAL DELAY CHARACTERS EXIST - SEE $TERM.xxx
;    14¯ CURSOR TRACK MODE FLAG FOR $TERMS
;    13
;    12_
;    11¯
;    10  TYPIST MODE 2: ACTIVATE ON CTRL CHAR AND CONVERT TO !<LTR>
;     9_ TYPIST MODE 1: ACTIVATE ON ANY CHAR, CONVERT CTRL CHARS.
;     8¯ DESTRUCTIVE BACKSPACE, IF NOT MODE 1 OR 2
;     7  ESCAPE SEEN IN INPUT, USE TRANSLATION TABLE #1
;     6_ OUTPUT TRANSLATION IN PROGRESS
;     5¯ INPUT TRANSLATION IN PROGRESS
;     4  EXPECTING CURSOR POSITION ('RD' HAS BEEN SENT)
;     3_ \
;     2¯  } TERMINAL TYPE NUMBER
;     1   }     (O TO 17 OCTAL)
;     0_ /

; * NOTE:  BIT 15 IS THE MOST SIGNIFICANT BIT
```

```
- PAGE 6 -

; DFT HAS EIGHT WORDS PER CHANNEL AS FOLLOWS:

                0 .DUSR FLU = 0 ;FILE'S LOGICAL UNIT NUMBER
                                ;-1 IN FLU ==> DEVICE (ON LU #0)
                1 .DUSR FDA = 1 ;FILE HEADER DISC ADDRESS
                2 .DUSR CBN = 2 ;CURRENT BLOCK NUMBER
                                ;CBN = INIT ENTRY ADDRESS IF DEVICE
                                ;CBN = PARTITION ENTRY POINTER IN CH # -1
                                ;CBN = A (ACT FILE HDR) IN CH # -2
                3 .DUSR STS = 3 ;CHANNEL STATUS (SEE BELOW)
                4 .DUSR FSZ = 4 ;FILE'S SIZE (# DATA BLOCKS IF CONTIGUOUS)
                                ;FSZ = RECORD NUMBER IF NOT CONTIGUOUS FILE
                5 .DUSR WPR = 5 ;NUMBER OF WORDS PER RECORD
                6 .DUSR FRR = 6 ;FIRST REAL RECORD NUMBER
                7 .DUSR CNP = 7 ;CHANNEL NODE POINTER (FOR MODULAR FILES)

                  ; NOTE: THE NEGATIVE NUMBERED CHANNELS ARE USED
                  ;       DIFFERENTLY.  REFER TO MANAGER REFERENCE
                  ;       MANUAL FOR MORE INFORMATION.


; EACH BIT IN CHANNEL STS IS A FLAG AS FOLLOWS:

;  BIT*   MEANING
;   15_ RECORD IS LOCKED (IN CHM1 ==> PROGRAM IS LOCKABLE)
;   14¯ FILE IS WRITE PROTECTED
;   13  FILE IS CONTIGUOUS
;   12_ FILE IS NOT FORMATTED
;   11¯ PERIPHERAL DEVICE
;   10  FILE IS INDEXED
;    9_ (RESERVED FOR BYTE NUMBER OVERFLOW)
;    8¯ \
;    7   \
;    6_   \
;    5¯    }   DISPLACEMENT OF
;    4     }  RECORD INTO BLOCK
;    3_    }  (NUMBER OF BYTES)
;    2¯   /
;    1   /
;    0_ /

; * NOTE:  BIT 15 IS THE MOST SIGNIFICANT BIT
```

```
;; FORM OF OVERALL USER PARTITION AREA
;; (DISPLACEMENTS FROM .UPT. POINTER IN INFO):
;;
              0 .DUSR EPA. = 0       ; POINTER TO END OF PARTITION AREA
              1 .DUSR NPA. = EPA.+1 ; CURRENT NUMBER OF PARTITIONS
              2 .DUSR FPA. = NPA.+1 ; BEGINNING (WORD SZP.) OF FIRST PARTITION ENTRY
;;
;; PARTITION ENTRY FORMAT (DISPLACEMENTS FROM RUS):
;;
         177774 .DUSR SZP. = -4      ; CURRENT SIZE OF THIS PARTITION AREA(INCLUDES GAP).*
         177775 .DUSR JCP. = SZP.+1 ; JCB POINTER (USUALLY A PCB POINTER).
         177776 .DUSR AFP. = JCP.+1 ; ACTIVE FILE HEADER CORE LOCATION.
         177777 .DUSR LKP. = AFP.+1 ; ZERO IF THIS PARTITION IS NOT LOCKED;
                                    ; ELSE TIME AT END OF LAST TIME SLICE.
              0 .DUSR BUP. = LKP.+1 ; BEGINNING OF USER'S STORAGE.
;;
;; * NOTE: ACTUAL CURRENT (OR DESIRED) USER SIZE
;;         IS IN CSIZ OF THE ACTIVE FILE HEADER.


; TASK NUMBERS FOR "QUEUE"

              0 .DUSR SIGNA = 0 ;SEND A SIGNAL
              1 .DUSR TENHZ = 1 ;TEN HERTZ TASK


; task queue priorities

          77777 .DUSR QP.FS= 77777 ; FAULT START task
          77776 .DUSR QP.AW= 77776 ; task to AWAKE tasks on Deferred Queue
          62000 .DUSR QP.DP= 62000 ; DATAPUMP task
          60000 .DUSR QP.TH= 60000 ; TEN HERTZ task
          30000 .DUSR QP.SI= 30000 ; SIGNAL task
              2 .DUSR QP.FW=     2 ; FAULT WRITE task


; DISPLACEMENTS IN INTERRUPT STACK ENTRY (EIGHT WORDS PER LEVEL)

                ; WORDS  0 - 4  SAME AS IN TASK NODE (SEE NEXT P.)
                ; WORD   5      MSB IS CARRY BIT, BITS (14-0) = CURRENT INTERRUPT MASK
                ; WORD   6      SOURCE BYTE BASE ADDRESS (SBA)
                ; WORD   7      DESTINATION BYTE BASE ADDRESS (DBA)
             10 .DUSR ISTKF = 10 ;INTERRUPT STACK FRAME SIZE


; DEFINE USE OF SCHEDULER NODE DISPLACEMENTS BY TIMESHARING ALGORITHM

             20 .DUSR SEP  =20 ;SUM EFFECTIVE PRIORITIES
             21 .DUSR NRJ  =21 ;NEXT REGULAR JOB
             22 .DUSR NBJ  =22 ;NEXT BACKGROUND JOB
             23 .DUSR IPF  =23 ;INTERACTIVE PAHSE FLAG
             24 .DUSR MRC  =24 ;MINIMUM REGULAR CTC
             25 .DUSR MBC  =25 ;MINIMUM BACKGROUND CTC
             26 .DUSR PCT  =26 ;PORT COUNTER
             27 .DUSR CPA  =27 ;CURRENT PORT ADDRESS
             30 .DUSR PRI  =30 ;PRIORITY
             31 .DUSR SVC  =31 ;SERVICE
```

```
- PAGE 8 -

; TASK NODE DISPLACEMENTS

; CPU STATUS IS SAVED IN THE FIRST 8 WORDS IN CASE OF INTERRUPT

                0 .DUSR A2   = 0 ;REGISTER A2
                1 .DUSR A1   = 1 ;REGISTER A1
                2 .DUSR A0   = 2 ;REGISTER A0
                3 .DUSR A3   = 3 ;REGISTER A3
                4 .DUSR PC   = 4 ;PROGRAM COUNTER (RETURN ADDRESS)
                5 .DUSR CPRI = 5 ;CARRY (IN BIT 15) AND PRIORITY
                6 .DUSR SSBA = 6 ;SAVE SOURCE BYTE BASE ADDR (SBA)
                7 .DUSR SDBA = 7 ;SAVED DEST.BYTE BASE ADDRESS (DBA)

; THE NEXT TWO WORDS IDENTIFY THE TASK AND ARE PUT IN BY QUEUE

               10 .DUSR TCBP =10 ;TASK CONTROL BLOCK POINTER
               11 .DUSR TASK =11 ;TASK ENTRY POINTER

; THE FOLLOWING 18 WORDS MAY BE USED AT WILL BY THE REGNANT TASK, WITH
; THESE QUALIFICATIONS:
;     QRTN, QTMP, AND PAUZ ARE USED BY SLEEP

               12 .DUSR PAUZ =12 ;PAUSE (SLEEP) COUNTER
               13 .DUSR QTMP =13 ;TEMPORARY STORAGE FOR SLEEP
               14 .DUSR QRTN =14 ;RETURN ADDRESS FOR QUEUE AND SLEEP
               15 .DUSR N.A0 =15 ;SUGGESTED USE: ACCUMULATOR STORAGE
               16 .DUSR N.A1 =16 ;SUGGESTED USE: ACCUMULATOR STORAGE
               17 .DUSR N.A2 =17 ;SUGGESTED USE: ACCUMULATOR STORAGE
               20 .DUSR N.A3 =20 ;SUGGESTED USE: ACCUMULATOR STORAGE
               21 .DUSR N.CA =21 ;SUGGESTED USE: CARRY STORAGE
               22 .DUSR N.R1 =22 ;SUGGESTED USE: RETURN ADDRESS
               23 .DUSR N.R2 =23 ;SUGGESTED USE: RETURN ADDRESS
               24 .DUSR N.T0 =24 ;SUGGESTED USE: TEMPORARY STORAGE
               25 .DUSR N.T1 =25 ;SUGGESTED USE: TEMPORARY STORAGE
               26 .DUSR N.T2 =26 ;SUGGESTED USE: TEMPORARY STORAGE
               27 .DUSR N.T3 =27 ;SUGGESTED USE: TEMPORARY STORAGE
               30 .DUSR N.T4 =30 ;SUGGESTED USE: TEMPORARY STORAGE
               31 .DUSR N.T5 =31 ;SUGGESTED USE: TEMPORARY STORAGE
               32 .DUSR N.T6 =32 ;SUGGESTED USE: TEMPORARY STORAGE
               33 .DUSR N.T7 =33 ;SUGGESTED USE: TEMPORARY STORAGE

; THE LAST FOUR WORDS ARE RESERVED FOR SYSTEM USE

               34 .DUSR NSTS =34 ;NODE STATUS (WHAT QUEUE IT'S ON: SEE BELOW)
               35 .DUSR AUXL =35 ;LINK TO CALLING TASK CONTROL NODE, IF ANY
               36 .DUSR PLNK =36 ;POINTER TO LINK OF PREVIOUS NODE ON QUEUE
               37 .DUSR LINK =37 ;LINK TO WORD ZERO OF NEXT NODE ON QUEUE


; MEANINGS OF NSTS WORD

                  ; -2 ON FREE NODE CHAIN
                  , -1 LOOSE (NOT ON ANY CHAIN OR QUEUE)
                  ;  0 ON SYSTEM TASK QUEUE
                  ;  1 ON SYSTEM SLEEP QUEUE
                  ; >2 RESERVED FOR FUTURE USE
```

```
- PAGE 9 -

; DEFINE SPECIAL FLAGS FOR DISCSUBS

          40000 .DUSR X =40000 ;EXTENDED SUBROUTINE (TWO BLOCKS)
          20000 .DUSR N =20000 ;INCLUDED WITH ANOTHER IF CORE-RESIDENT
          10000 .DUSR D =10000 ;VERSION IS DISC-RESIDENT ONLY
           4000 .DUSR A =04000 ;ALTERNATE VERSION FOR CORE RESIDENCY
                ;   NOTE:  'A' TYPES ARE NOT IMPLEMENTED FOR "IRIS" R8.0 AND SUBS.


; DISCSUB NUMBERS FOR "CALL"

          10000 .DUSR FAULT = 0+D    ;PRINT TRAP MESSAGE, ABORT TASK          (11000)
              1 .DUSR ALLOC = 1      ;ALLOCATE DISC BLOCKS                     (5000)
              2 .DUSR DALLC = 2      ;DEALLOCATE DISC BLOCKS                   (4400)
              3 .DUSR FFILE = 3      ;FIND FILE IN INDEX                       (4000)
              4 .DUSR EXTEN = 4      ;CHANGE TO EXTENDED FILE                 (24000)
              5 .DUSR ALCON = 5      ;ALLOCATE A CONTIGUOUS FILE              (24400)
              6 .DUSR CDTA  = 6      ;CONVERT DRATSAB TO ASCII
              7 .DUSR CIA   = 7      ;CONVERT INTEGER TO ASCII (ANY RADIX)   (10400)
             10 .DUSR CSTR  =10      ;COMPARE STRINGS                          (3000)
             11 .DUSR PASSC =11      ;PASSWORD COMPARE                          (400)
             12 .DUSR ERROR =12      ;ERROR ROUTINE FOR BASIC                 (14400)
             13 .DUSR MESSA =13      ;CANNED MESSAGE TO I/O BUFFER             (6400)
             14 .DUSR BREAK =14      ;BREAKPOINT SETUP FOR DSP                 (6400)
             15 .DUSR ACNTL =15      ;ACCOUNT LOOKUP                           (6000)
             16 .DUSR DELET =16      ;DELETE A FILE                            (5400)
          20017 .DUSR PDELE =17+N    ;DELETE A PROCESSOR OR DRIVER             (5400)
          40020 .DUSR BUILD =20+X    ;BUILD A NEW FILE                         (1000)
          60021 .DUSR BILDD =21+N+X  ;BUILD A "$" FILE                         (1000)
          40022 .DUSR OPEN  =22+X    ;OPEN A FILE OR A DEVICE                  (2000)
          60023 .DUSR OPENU =23+N+X  ;OPEN A FILE OR A DEVICE FOR UPDATE       (2000)
          60024 .DUSR OPENL =24+N+X  ;OPEN AND LOCK A FILE OR A DEVICE         (2000)
          60025 .DUSR OPENR =25+N+X  ;OPEN A FILE OR A DEVICE FOR REFERENCE    (2000)
             26 .DUSR CLOSE =26      ;CLOSE A CHANNEL                         (12000)
             27 .DUSR CLEAR =27      ;CLEAR A CHANNEL                          (3000)
          40030 .DUSR GETRR =30+X    ;GET RECORD FOR READ                      (3000)
          60031 .DUSR GETRW =31+N+X  ;GET RECORD FOR WRITE                     (3000)
             32 .DUSR FINDI =32      ;FIND AN ITEM (NOT IMPLEMENTED)
             33 .DUSR READI =33      ;READ AN ITEM                              (400)
          20034 .DUSR WRITI =34+N    ;WRITE AN ITEM                             (400)
             35 .DUSR WRITN =35      ;WRITE A NEW ITEM                        (24000)
             36 .DUSR READC =36      ;READ FROM CONTIGUOUS FILE               (25000)
          20037 .DUSR WRITC =37+N    ;WRITE INTO CONTIGUOUS FILE              (25000)
             40 .DUSR CHARG =40      ;CHARGE FOR FILE ACCESS                   (3000)
             41 .DUSR SYSCO =41      ;TRANSMIT A SYSTEM COMMAND (CALL 98)     (21400)
             42 .DUSR CNVDA =42      ;CONVERT DATE TO ASCII                   (10000)
             43 .DUSR CNVAD =43      ;CONVERT ASCII TO DATE                    (7400)
             44 .DUSR CNVDT =44      ;CONVERT DATE AND TIME (CALL 99)         (10000)
             45 .DUSR RDFHI =45      ;READ FILE HEADER INFO (CALL 97)         (20000)
             46 .DUSR SPECI =46      ;SPECIAL FUNCTIONS                       (20400)
          10047 .DUSR RECOV =47+D    ;RECOVER FROM A STALL OR CRASH            (7400)
             50 .DUSR PATNF =50      ;PSEUDO DIVIDE ARC TANGENT FUNCTION      (14400)
             51 .DUSR PLOGF =51      ;PSEUDO DIVIDE NATURAL LOG FUNCTION      (15000)
             52 .DUSR PSQRF =52      ;PSEUDO DIVIDE SQUARE ROOT FUNCTION      (14000)

; (CONTINUED ON NEXT PAGE)
; DISCSUB NUMBERS (CONTINUED)

        40053 .DUSR PEXPF =53+X    ;PSEUDO DIVIDE EXPONENTIAL FUNCTION      (15400)
        40054 .DUSR PSINF =54+X    ;PSEUDO DIVIDE SINE FUNCTION             (17000)
        60055 .DUSR PCOSF =55+N+X  ;PSEUDO DIVIDE COSINE FUNCTION           (17000)
           56 .DUSR PTANF =56      ;PSEUDO DIVIDE TANGENT FUNCTION          (16400)
           57 .DUSR SIGPA =57      ;SIGNAL OR PAUSE                         (10400)
           60 .DUSR DIREC =60      ;SET UP DIRECTORIES FOR INDEXED FILE     (26000)
        40061 .DUSR SEARC =61+X    ;SEARCH INDEXED FILE DIRECTORY           (26400)
           62 .DUSR SHUFF =62      ;SHUFFLE DIRECTORY BLOCKS                (27400)
           63 .DUSR DEKEY =63      ;DELETE KEY FROM DIRECTORY               (30000)
        20064 .DUSR RELEA =64+N    ;RELEASE A DIRECTORY BLOCK               (27400)
        40065 .DUSR FIXDI =65+X    ;FIX DIRECTORIES OF MOVED INDEXED FILE   (33000)
        40066 .DUSR REOPT =66+X    ;RE-OPTIMIZE INDEXED FILE DIRECTORY      (34000)
           67 .DUSR AFSET =67      ;SET UP ACTIVE FILE FOR SWAP-OUT          (7000)
           71 .DUSR MRC3  =71      ;MAG TAPE READ STATUS                    (32400)
           72 .DUSR MTASK =72      ;MAG TAPE SUPPLEMENTARY TASKS            (31000)
           73 .DUSR MRFHD =73      ;MAG TAPE READ FILE HEADER               (33000)
           74 .DUSR MRFIL =74      ;MAG TAPE READ INPUT FILE                (30400)
           75 .DUSR MTFPE =75      ;MAG TAPE READ/WRITE TRANSFERS           (32000)
        20076 .DUSR MNEXT =76+N    ;MAG TAPE GO TO NEXT DRIVE               (31000)
           77 .DUSR MTAPA =77      ;MAG TAPE ALL OTHER FUNCTIONS            (31400)
        40100 .DUSR LINKP =100+X   ;LINK PROGRAMS (BASIC'S "CHAIN")         (12400)
        60101 .DUSR LOADP =101+N+X;LOAD A BASIC PROGRAM                     (12400)
        40102 .DUSR CTNXT =102+X   ;CTU POST-PROCESSING TASK                (35000)
          103 .DUSR CTUSR =103     ;CTU DIR SEARCH ROUTINE                  (32400)
          104 .DUSR CTUWE =104     ;CTU WRITE DIR ENTRY ROUTINE             (36000)
          105 .DUSR ST105 =105     ;STYLUS CALL(89)                         (52000)
        20106 .DUSR ST106 =106+N   ;FIND A SUBSTRING IN A STRING            (51400)
          107 .DUSR ST107 =107     ;REVERSE A STRING                        (51400)
          110 .DUSR FINDF =110     ;FIND A FILE (CALL 96)                   (21000)
          111 .DUSR RDISC =111     ;READ OR WRITE WORD TO DISC (CALL 95)    (22000)
          112 .DUSR CHFLT =112     ;CHANGE FILE TYPE (CALL 94)              (21000)
          113 .DUSR WRWRD =113     ;WRITE WORD TO CORE (CALL 93)            (21000)
          114 .DUSR XCOM1 =114     ;FOR BI-SYNC DRIVER
          115 .DUSR XCOM2 =115     ;FOR BI-SYNC DRIVER
          116 .DUSR BCOMM =116     ;BUILD COMMUNICATIONS FILE
          117 .DUSR BCOMF =117     ;COMMUNICATIONS FILE HANDLER

              ;             (CONTINUED ON NEXT PAGE)
```

```
                 ; DISCSUB NUMBERS (CONTINUED)

      40120 .DUSR MODE0 =120+X  ;POLYMODE MODE 0                          (40000)
      40121 .DUSR MODE1 =121+X  ;POLYMODE MODE 1                          (41000)
      40122 .DUSR PFSEA =122+X  ;POLYMODE SEARCH (MODES 2 - 5)            (42000)
        123 .DUSR MODE4 =123    ;POLYMODE 4 (CALLED FROM PFSEA)           (43000)
        124 .DUSR PRCOM =124    ;POLYFILE COMBINE BLOCKS                  (44400)
        125 .DUSR PFSHF =125    ;POLYMODE 4 KEY SHUFFLE                   (43400)
        126 .DUSR PFSHX =126    ;POLYMODE 4 KEY SHUFFLE(FOR BLK SPLIT)    (44000)
        127 .DUSR MODE5 =127    ;POLYMODE 5 (CALLED FROM PFSEA)           (45000)
        130 .DUSR PFABL =130    ;POLYFILE ALLOCATE DIRECTORY BLK          (45400)
      20131 .DUSR PFALL =131+N  ;POLYFILE ALLOCATE (FROM BIT MAP)         (45400)
        132 .DUSR PFRLS =132    ;POLYFILE RELEASE                         (46000)
        133 .DUSR PFSCN =133    ;POLYFILE SCAN                            (46000)
        134 .DUSR VOLRE =134    ;POLYFILE READ FROM VOLUME                (46000)
        135 .DUSR MAPBU =135    ;POLYFILE MAP BUILD                       (46400)
        136 .DUSR DIRFN =136    ;POLYFILE FIND DIRECTORY                  (46400)
        137 .DUSR SZMAP =137    ;POLYFILE COMPUTE BIT MAP SIZE            (46400)
        140 .DUSR DATCK =140    ;POLYFILE VALIDATE DATA VOL. DEFINIT      (47000)
        141 .DUSR OPENP =141    ;POLYFILE OPEN (CALLED BY other OPEN)     (13400)
      40142 .DUSR READP =142+X  ;POLYFILE READ                            (47400)
      60143 .DUSR WRITP =143+X+N;POLYFILE WRITE                           (47400)
      40144 .DUSR CALLP =144+X  ;POLYFILE BUILD                           (50400)
        145 .DUSR JULIA =145    ;JULIAN DATE ROUTINE
      40146 .DUSR CLPY1 =146+X  ;POLYFILE BUILD EXTENTION                 (51400)

                 ; DISCSUB NUMBERS 147 THROUGH 150 ARE CURRENTLY
                 ; UNASSIGNED AND ARE RESERVED FOR FUTURE EXPANSION

      40151 .DUSR ST151 =151+X  ;STYLUS' EDIT AND JUSTIFY                 (52000)
        152 .DUSR ST152 =152    ;STYLUS' PUSH OR POP A CHARACTER ARRAY    (37400)
        153 .DUSR ST153 =153    ;STYLUS' CALL 56                          (36400)
        154 .DUSR ST154 =154    ;STYLUS' UNDERLINE                        (25400)
        155 .DUSR ST155 =155    ;STYLUS' PORT BUFFER LENGTH ADJ           (25400)
        156 .DUSR ST156 =156    ;STYLUS' JUSTIFICATION CALCULATOR         (37000)
        157 .DUSR ST157 =157    ;STYLUS' STRING LOCATOR                   (51400)
        160 .DUSR TIP01 =160    ;TYPIST'S CALL 81                         (56400)
        161 .DUSR TIP02 =161    ;TYPIST'S CALL 82                         (57000)
        162 .DUSR TIP03 =162    ;TYPIST'S CALL 83                         (57400)
        163 .DUSR TIP04 =163    ;TYPIST'S CALL 84                         (60000)
        164 .DUSR TIP05 =164    ;TYPIST'S CALL 85                         (60400)
        165 .DUSR TIP06 =165    ;TYPIST'S CALL 86                         (61000)
        166 .DUSR ATOE = 166    ;ASCII TO EBCDIC                          (63000)
        167 .DUSR ETOA = 167    ;EBCDIC TO ASCII                          (63400)

                 ; DISCSUB NUMBERS 170 THROUGH 177 ARE TO BE
                 ; USED FOR CUSTOMER WRITTEN SUBROUTINES.
                 ; ANY NUMBER UP TO 777 MAY BE USED IF THE
                 ; NSUB VALUE IN THE INFO TABLE IS INCREASED.

; DEFINE AN ITEM CONTROL BLOCK FOR READITEM - WRITITEM

          0 .DUSR ITRCD =0        ;RECORD NUMBER
          1 .DUSR ITNUM =ITRCD+1  ;ITEM NUMBER
          2 .DUSR ITTYP =ITNUM+1  ;ITEM TYPE
          3 .DUSR ITLEN =ITTYP+1  ;ITEM LENGTH
          4 .DUSR ITDES =ITLEN+1  ;ITEM ADDRESS (DESTINATION OR SOURCE)
```

```
; CORE-RESIDENT SUBROUTINE NUMBERS FOR "CALL"

         100000 .DUSR SCOPE =@ 0 ;EXIT TO "SCOPE" PROCESSOR
                ;     FLUSH =@ 1 ;deleted -- use DATAPUMP call
         100002 .DUSR CHKCH =@ 2 ;CHECK CHANNEL
         100003 .DUSR ALLCL =@ 3 ;CLEAR ALL CHANNELS
         100004 .DUSR FOFC  =@ 4 ;FIND OPEN FILE (CONTINUE)
         100005 .DUSR FOFI  =@ 5 ;FIND OPEN FILE (INITIALIZE)
         100006 .DUSR LOADU =@ 6 ;LOAD USER'S ACTIVE FILE
         100007 .DUSR PILOA =@ 7 ;LOAD A PI USER (FOR IRIS-II)
         100010 .DUSR UNLOC =@10 ;UNLOCK RECORD
         100011 .DUSR WONA  =@11 ;WAIT FOR OUTPUT NOT ACTIVE
         100012 .DUSR CHKRP =@12 ;CHECK READ PROTECTION
         100013 .DUSR CHKWP =@13 ;CHECK WRITE PROTECTION
         100014 .DUSR CHKCP =@14 ;CHECK COPY PROTECTION
         100015 .DUSR MOVEW =@15 ;MOVE WORDS
         100016 .DUSR MOVBY =@16 ;MOVE BYTES
         100017 .DUSR XFIXB =@17 ;RELEASE FIXED BUFFERS
         100020 .DUSR CRLA  =@20 ;CONVERT REAL TO LOGICAL DISC ADDRESS
         100021 .DUSR CLRA  =@21 ;CONVERT LOGICAL TO REAL DISC ADDRESS
         100022 .DUSR CPPPN =@22 ;CONVERT PCB POINTER TO PORT NUMBER
         100023 .DUSR CPNPP =@23 ;CONVERT PORT NUMBER TO PCB POINTER
         100024 .DUSR SLEEP =@24 ;PUT A NODE ON THE SLEEP QUEUE
         100025 .DUSR AWAKE =@25 ;ACTIVATE A NODE FROM THE SLEEP QUEUE
         100026 .DUSR XQUEU =@26 ;RETRIEVE A NODE FROM SPECIFIED QUEUE
         100027 .DUSR SFIXB =@27 ;RELEASE SELECTED FIXED BUFFER
         100030 .DUSR CBSA  =@30 ;CHECK BSA AND WRITE OUT IF DIRTY
                ;           =@31 ;(RESERVED)
                ;           =@32 ;(RESERVED)
                ;           =@33 ;(RESERVED)
                ;           =@34 ;(RESERVED)
                ;           =@35 ;(RESERVED)
         100036 .DUSR VOLFN =@36 ;FIND A POLYFILE VOLUME BY TYPE
                ;           =@37 ;(RESERVED)
                ;           =@40 ;(RESERVED)
                ;           =@41 ;(RESERVED)
                ;           =@42 ;(RESERVED)
                ;           =@43 ;(RESERVED)
                ;           =@44 ;(RESERVED)
         100045 .DUSR VOLLK =@45 ;POLYFILE VOLUME LOOKUP


; DISPLACEMENTS FOR EACH ENTRY IN "ACCOUNTS" FILE

                ;           0-5   ;ACCOUNT ID STRING
              6 .DUSR APR.  = 6   ;ASSIGNED PRIORITY (0 TO 7)
              7 .DUSR ALU.  = 7   ;ASSIGNED LOGICAL UNIT
             10 .DUSR ACN.  =10   ;ACCOUNT NUMBER (PRIV, GROUP, USER)
             11 .DUSR CMR.  =11   ;CONNECT MINUTES REMAINING
             12 .DUSR RUR   =12   ;RESOURCE UNITS REMAINING / 256
             13 .DUSR MDB.  =13   ;MAX. DISC BLOCKS ALLOTTED
             14 .DUSR DBU.  =14   ;DISC BLOCKS NOW IN USE
             15 .DUSR RUR1. =15   ;RESOURCE UNITS REMAINING * 256 (2ND WORD)
             16 .DUSR CHG.  =16   ;FILE USE CHARGES (FLOATING 2-WORD BCD)
                ;            17   ; "           "
; HEADER BLOCK DISPLACEMENTS (SEE MANAGER MANUAL)

          0 .DUSR NAME = 0   ;FILENAME STRING (7 WORDS)
          7 .DUSR ACNT = 7   ;PRIV LEVEL, ACCOUNT (GROUP, USER)
         10 .DUSR TYPE = 10  ;FILE TYPE AND PROTECTION
         11 .DUSR NBLK = 11  ;NUMBER OF BLOCKS IN FILE (INCL. HEADER)
         12 .DUSR STAT = 12  ;FILE STATUS (SEE BELOW)
         13 .DUSR NITM = 13  ;NUMBER OF ITEMS PER RECORD        \  ALSO
         14 .DUSR LRCD = 14  ;LENGTH OF EACH RECORD (# WORDS)    ) USED
         15 .DUSR NRPB = 15  ;NUMBER OF RECORDS PER BLOCK        )  BY
         16 .DUSR NRCD = 16  ;NUMBER OF RECORDS IN FILE         /  DSP.
         17 .DUSR COST = 17  ;DIMES CHARGED FOR ACCESS TO FILE
         20 .DUSR CHGS = 20  ;TOTAL CHARGES FOR FILE USAGE (2 WORDS)
         22 .DUSR LDAT = 22  ;LAST ACCESS DATE (HOURS, TENTH-SECONDS)
         24 .DUSR CDAT = 24  ;FILE CREATION DATE (HOURS, TENTH-SECONDS)
         26 .DUSR NTAC = 26  ;NUMBER OF TIMES ACCESSED
            .DUSR CATR = 27  ;"CATALOG" RECORD NUMBER
         30 .DUSR CLAS = 30  ;CATALOG CLASSIFICATION (2 WORDS)
         32 .DUSR LCMB = 32  ;LCM BLOCK # OF ACTIVE FILE
            ;        33-34     (RESERVED)
         35 .DUSR PFUN = 35  ;PROGRAM'S ASSIGNED POLICY FUNCTION
         36 .DUSR SNUM = 36  ;SCO NUMBER OF LAST SCO APPLIED
         37 .DUSR ADAT = 37  ;DATE LAST SCO APPLIED (HOURS AFTER BASEY)
         40 .DUSR DASA = 40  ;DECIMAL ACCUMULATOR SAVE AREA (10 WORDS)
         50 .DUSR DSPS = 50  ;STORAGE FOR DSP (20 WORDS)
         70 .DUSR FMAP = 70  ;DATA FILE FORMAT MAP (101 WORDS)
        171 .DUSR HTEM =171  ;TEMP CELL USED BY ALLOC, DALLC & ACNTL
        172 .DUSR STAD =172  ;START ADDRESS (DRIVER OR STAND-ALONE)
        173 .DUSR CSIZ =173  ;CURRENT SIZE (NUMBER OF WORDS)
        174 .DUSR DSAF =174  ;DEFAULT SIZE OF ACTIVE FILE (# BLOCKS)
        175 .DUSR CORA =175  ;CORE ADDRESS OF FIRST DATA BLOCK
        176 .DUSR UNIT =176  ;LOGICAL UNIT NUMBER WHERE FILE RESIDES
        177 .DUSR DHDR =177  ;REAL DISC ADDRESS OF HEADER BLOCK

            ; WORDS 200-377 HOLD REAL DISC ADDRESSES OF DATA BLOCKS
            ; FOR A NON-EXTENDED RANDOM FILE, OR REAL DISC ADDRESSES
            ; OF HEADER EXTENDER BLOCKS FOR AN EXTENDED RANDOM FILE.
            ; WORDS 200-377 ARE NOT USED IN A CONTIGUOUS FILE HEADER.

            ; CORE ADDRESSES ARE AT 400 WORD STEPS FROM CORA;
            ; I.E., THE CORE ADDRESS FOR THE NTH DATA BLOCK IS
            ; CORA + 400*N, WHERE N=0 FOR THE FIRST DATA BLOCK.

            ; EACH BIT IN STAT IS A FLAG AS FOLLOWS:
            ;  BIT   MEANING
            ;   15_ FILE IS BEING BUILT, NOT YET CLOSED
            ;   14¯ A FILE IS BEING BUILT TO REPLACE THIS ONE
            ;   13  FILE IS TO BE DELETED
            ;   12_ FILE IS MAPPED (FORMATTED DATA FILE)
            ;   11¯ FILE HAS BEEN OPENED WITH AN OPENLOCK
            ;   10  FILE IS NOT DELETEABLE
            ;  9-1  SPARE
            ;    0  FILE IS EXTENDED
            ; IF BITS 14 OR 13 ARE SET THE FILE WILL BE DELETED WHEN CLOSED

; DEFINE DEFAULT PROGRAM PRIORITY FOR NEW PROGRAM FILE
          5 DUSR DPRI =5
```

```
- PAGE 14 -

; COMMAND WORD DEFINITIONS FOR DATAPUMP OPERATIONS

          0 .DUSR GETBLOCK   =  0 ;GET AND LATCH A BLOCK INTO A POOL BUFFER
          1 .DUSR GETPRIVATE =  1 ;GET A PRIVATE POOL BUFFER
          2 .DUSR QREAD      =  2 ;READ BLOCK INTO PRIVATE BFR. OR NON-POOL SPACE
          3 .DUSR UNLATCH    =  3 ;UNLATCH A POOL BUFFER; WRITE IF DIRTY
          4 .DUSR QWRITE     =  4 ;WRITE BLOCK FROM PRIVATE BFR. OR NON-POOL SPACE
          5 .DUSR PUBLIC     =  5 ;RELEASE A PRIVATE POOL BUFFER
          6 .DUSR PUTBLOCK   =  6 ;WRITE A BLOCK FROM A POOL BUFFER
          7 .DUSR SETDIRTY   =  7 ;SET THE POOL BUFFER DIRTY FLAG.(WRITE)

         10 .DUSR READ       = 10 ;READ BLOCK VIA POOL INTO PRIVATE OR NON-POOL
         11 .DUSR WRITE      = 11 ;WRITE BLOCK VIA POOL FROM PRIVATE OR NON-POOL

         12 .DUSR FILEREAD   = 12 ;READ ENTIRE FILE OF HEADER IN POOL
         13 .DUSR FILEWRITE  = 13 ;WRITE ENTIRE FILE OF HEADER IN POOL

         14 .DUSR BUFFLUSH   = 14 ;FLUSH A SINGLE BUFFER
         15 .DUSR LRUFLUSH   = 15 ;FLUSH THE LEAST RECENTLY USED DIRTY BUFFER
         16 .DUSR LUFLUSH    = 16 ;FLUSH ALL BUFFERS OF GIVEN LU
         17 .DUSR ALLFLUSH   = 17 ;FLUSH ENTIRE POOL

            ;     D.READ     = 20 ;DIRECT READ (NOT VIA POOL)  } FOR DSP
            ;     D.WRITE    = 21 ;DIRECT WRITE (NOT VIA POOL) } ONLY
            ;     DFILREAD   = 22 ;READ FILE BYPASSING POOL  } FOR REX (& $LCM)
            ;     DFILWRITE  = 23 ;WRITE FILE BYPASSING POOL } ONLY
            ;     SAVDISCSUB = 24 ;SAVE (SSA) IN ITS POOL IMAGE, FOR REX USE ONLY

            ; SUFFIX DEFINITIONS FOR THE COMMAND WORDS

     100000 .DUSR UL     = 100000 ;UNLATCH THE POOL BUFFER UPON COMPLETION
      40000 .DUSR CL     =  40000 ;CLEAR THE BUFFER UPON COMPLETION
```

```
; STANDARD DISC BLOCK ALLOCATION ON EACH LOGICAL UNIT

                      ;                   0 ;BZUD, BLOCK ZERO UTILITY DRIVER
                    1 .DUSR KINDEX=       1 ;INDEX HEADER
                    2 .DUSR KUTILITY=     2 ;BTUP, BLOCK TWO UTILITY PACKAGE, ON LU 0 ONLY
                    3 .DUSR KACCOUNTS=    3 ;HEADER OF ACCOUNTS FILE
                    4 .DUSR KREX=         4 ;REX HEADER, ON LU 0 ONLY
                    5 .DUSR KCOPYRIGHT=   5 ;BLOCK CONTAINING COPYRIGHT NOTICE, ON LU 0 ONLY
                      ;     SECTOR 0 ON TRACK 1 ;DMAP HEADER


; STRUCTURE OF A LOGICAL UNIT TABLE (LUT) ENTRY.

                    0 .DUSR .LFX. = 0         ;A(THIS UNIT'S LUFIX TABLE)
                    1 .DUSR .LVR. = .LFX.+1   ;A(THIS UNIT'S LUVAR TABLE)
                    2 .DUSR .LLU. = .LVR.+1   ;THIS UNIT'S LOGICAL UNIT NUMBER


                  575 .DUSR TLUT = INFO-3     ;TEMPORARY (AND MIN.CONFIG) LOCATION OF LUT


; LOGICAL UNIT FIXED INFORMATION TABLE (LUFIX)

               177762 .DUSR RWEP =-16 ;READ/WRITE ENTRY POINT
               177763 .DUSR DSIZ =-15 ;SIZE OF DRIVER
               177764 .DUSR PFRD =-14 ;POWER FAIL RESTART DELAY
               177765 .DUSR EMSK =-13 ;"ANY ERROR" STATUS MASK
                      ;           -12 ;"WRITE PROTECTED" MASK
                      ;           -11 ;"NO SUCH DISC" MASK
                      ;           -10 ;"DATA CHANNEL LATE" MASK
                      ;            -7 ;"ADDRESS CHECK ERROR" MASK
                      ;            -6 ;"ILLEGAL DISC ADDRESS" MASK
               177773 .DUSR IDRV = -5 ;"INITIALIZE DRIVER" SUBROUTINE POINTER
               177774 .DUSR SLUR = -4 ;"SKIP IF LU READY" SUBROUTINE POINTER
               177775 .DUSR SKNB = -3 ;"SKIP IF NOT BUSY" SUBROUTINE POINTER
               177776 .DUSR REDS = -2 ;"READ STATUS" SUBROUTINE POINTER
               177777 .DUSR SEEK = -1 ;"SEEK OR RECALIBRATE" SUBROUTINE POINTER
                      ;             0 ;"READ/WRITE" SUBROUTINE ENTRY
; LOGICAL UNIT VARIABLE INFORMATION TABLE (LUVAR)

              0 .DUSR AVBC = 0 ;AVAILABLE BLOCK COUNT (SET BY SIR)
              1 .DUSR MINB = 1 ;# PHYS. TRACKS/CYL ! MIN AVAIL BLOCKS FOR CREATING A NEW FILE
              2 .DUSR DFLG = 2 ;DISC FLAG WORD (SEE BELOW)
              3 .DUSR DRIV = 3 ;PHYSICAL DRIVE SELECTION CONSTANT (not used in R8.1)
              4 .DUSR PHYU = 4 ;PHYSICAL UNIT SELECTION CONSTANT
              5 .DUSR FCYL = 5 ;FIRST PHYSICAL CYLINDER NUMBER
              6 .DUSR NCYL = 6 ;NUMBER OF CYLINDERS
              7 .DUSR NTRS = 7 ;[# IRIS-TRACKS] * 100 + [# IRIS-SECTORS]
             10 .DUSR FUDA =10 ;FIRST UNUSED REAL DISC ADDRESS (SET BY SIR)
             11 .DUSR ERRC =11 ;DATA CHECK ERROR COUNT
                ;           12 ;ADDRESS CHECK ERROR COUNT
                ;           13 ;DATA CHANNEL LATE COUNT
                ;           14 ;time-out error count


                ; EACH BIT IN DFLG IS A FLAG AS FOLLOWS:

                ;  BIT*   MEANING
                ;   15_ CHANGEABLE CARTRIDGE FLAG
                ;   14‾ FIXED HEAD DISC (USE ALLOC FOR ACTIVE FILE)
                ;   13
                ;   12_
                ;   11‾ SKIP SECTOR BETWEEN TRACKS WITHIN CYLINDER
                ;   10  SAME SECTOR NEXT TRACK \  NEXT BEST BLOCK
                ;    9_ NEXT SECTOR NEXT TRACK  } IF DESIRED IS
                ;    8‾ NEXT SECTOR SAME TRACK /  NOT AVAILABLE.
                ;    7  CANNOT TRANSFER SEQUENTIAL SECTORS
                ;    6_ SECTORS ARE PHYSICALLY SEQUENTIAL
                ;    5‾ SEEK IS IMPLICIT WITH TRANSFER COMMAND (not used in R8.1)
                ;    4  TRUE OVERLAP SEEK IS ALLOWED (not used in R8.1)
                ;    3_ CONCURRENT SEEK IS ALLOWED (not used in R8.1)
                ;    2‾ \
                ;    1   } NUMBER OF PHYSICAL DRIVES ON DRIVER -1 (not used in R8.1)
                ;    0_ /

                ; * NOTE:  BIT 15 IS THE MOST SIGNIFICANT BIT


               .EOT  ;SOFTWARE DEFINITIONS FOR "IRIS" R8.1
```

```
;;  "REX" == REAL-TIME EXECUTIVE FOR "IRIS" R8.1
;;  WRITTEN BY DAN PAYMAR
;;  MODIFIED FOR "IRIS" R8.0 BY G. DAVIE
;;  LAST EDITED 22 AUG 1982
;
;       .RDX 10
;
;MONTH = 8
;DAY   = 22
;YEAR  = 1982
;
;;                  ALL RIGHTS RESERVED
;;      COPYRIGHT (C) 1974, EDUCATIONAL DATA SYSTEMS
;;      COPYRIGHT (C) 1979, EDUCATIONAL DATA SYSTEMS
;;      COPYRIGHT (C) 1982, POINT 4 DATA CORPORATION
;;     THIS DOCUMENT CONTAINS SECRET AND CONFIDENTIAL
;;    INFORMATION OF POINT 4 DATA CORPORATION.  IT MAY
;;   NOT BE REPRODUCED,  USED, OR DISCLOSED WITHOUT THE
;;  PRIOR WRITTEN PERMISSION OF POINT 4 DATA CORPORATION.
;
;;  REX ASSEMBLY DATE (HOURS AFTER JAN 1 OF BASE YEAR)
;RDATE = YEAR-BASEYEAR*12+MONTH-1*31+DAY-1*24
;
;
             1       .TXTM  1
             0       .LOC   0
            10       .RDX   8

      0      1       .BLK   1
      1      1 .BLK  1      ;initial INTERRUPT VECTOR = Ignore interrupts

      2      1 C2:   .BLK   1
      3      1 C3:   .BLK   1
      4      1 .BLK  1      ;SPARE

      5      1 RUP:  .BLK   1
      6      1 RUS:  .BLK   1
      7      1 RTP:  .BLK   1

     10      1 .BSA: .BLK   1
     11      1 .HDA: .BLK   1
     12      1 .HXA: .BLK   1
     13      1 .SSA: .BLK   1

     14      1 .ABA: .BLK   1

     15      1 TASKQ:.BLK   1

     16      1 BPI:  .BLK   1
     17      1       .BLK   1
     20      1       .BLK   1
```

```
- PAGE 10 -

        21        1 C170K:. BLK    1
        22        1 C774C:. BLK    1
        23        1 CM400:. BLK    1

        24        1 C4:    . BLK    1
        25        1 C5:    . BLK    1
        26        1 C6:    . BLK    1
        27        1 C7:    . BLK    1
        30        1 C10:   . BLK    1
        31        1 C11:   . BLK    1
        32        1 C12:   . BLK    1
        33        1 C13:   . BLK    1
        34        1 C14:   . BLK    1
        35        1 C15:   . BLK    1
        36        1 C16:   . BLK    1
        37        1 C17:   . BLK    1
        40        1 C20:   . BLK    1
        41        1 C37:   . BLK    1
        42        1 C40:   . BLK    1
        43        1 C77:   . BLK    1
        44        1 C100:  . BLK    1
        45        1 C177:  . BLK    1

        46        1        . BLK    1
        47        1  . BLK  1              ; AT 46 AND 47 FOR NOVA 3

        50        1 C200:  . BLK    1
        51        1 C205:  . BLK    1
        52        1 C215:  . BLK    1
        53        1 C240:  . BLK    1
        54        1 C244:  . BLK    1
        55        1 C260:  . BLK    1
        56        1 C271:  . BLK    1
        57        1 C300:  . BLK    1
        60        1 C334:  . BLK    1
        61        1 C377:  . BLK    1
        62        1 C400:  . BLK    1
        63        1 C777:  . BLK    1
        64        1 C1000: . BLK    1
        65        1 C1777: . BLK    1
        66        1 C2000: . BLK    1
        67        1 C4000: . BLK    1

        70        1 ESCF:  . BLK    1
        71        1 ETSF:  . BLK    1
        72        1 BSACF: . BLK    1
        73        1 ERRF:  . BLK    1
        74        1 SBA:   . BLK    1
        75        1 DBA:   . BLK    1

        76        1 . BPS:  . BLK   1
        77        1 . INFO: . BLK   1
```

```
                        ; SYSTEM COMMAND CALLS
          6100 CALL     =JSR @.  ;CALL A SYSTEM SUBROUTINE
100          1         .BLK   1

          6101 FLAGC    =JSR @.  ;CHANGE OR CHECK A FLAG
101          1         .BLK   1

          6102 QCHAR    =JSR @.  ;QUEUE A CHARACTER TO BE PROCESSED
102          1         .BLK   1

          6103 QUEUE    =JSR @.  ;PUT A TASK IN THE QUEUE
103          1         .BLK   1

          6104 DQUEUE   =JSR @.  ;REMOVE REGNANT TASK FROM THE QUEUE
104          1         .BLK   1

          6105 CHANNEL =JSR @.   ;PERFORM A CHANNEL OPERATION
105          1         .BLK   1

          6106 FREENODE=JSR @.   ;GET OR RELEASE A FREE 32-WORD NODE
106          1         .BLK   1

          6107 DATAPUMP=JSR @.   ;INITIATE A DMA DATA TRANSFER
107          1         .BLK   1


110          1 .INTR:.BLK     1
111          1 .NRET:.BLK     1
112          1 .SRET:.BLK     1
113          1 .LCM: .BLK     1
```

```
                         ; SYSTEM SUBROUTINE CALLS
           6114 BINDIVIDE   =JSR @.
114           1        .BLK     1
           6115 BINMULTIPLY=JSR @.
115           1        .BLK     1
           6116 BUMPUSER    =JSR @.
116           1        .BLK     1
           6117 DECIMAL     =JSR @.
117           1        .BLK     1
           6120 FIX         =JSR @.
120           1        .BLK     1
           6121 FLOAT       =JSR @.
121           1        .BLK     1
           6122 FINDLUT     =JSR @.
122           1        .BLK     1
           6123 GETBYTE     =JSR @.
123           1        .BLK     1
           6124 INBYTE      =JSR @.
124           1        .BLK     1
           6125 INSTBYTE    =JSR @.
125           1        .BLK     1
           6126 ISA2DIGIT   =JSR @.
126           1        .BLK     1
           6127 ISA2LETTER  =JSR @.
127           1        .BLK     1
           6130 LOADDA      =JSR @.
130           1        .BLK     1
           6131 OUTBYTE     =JSR @.
131           1        .BLK     1
           6132 OUTTEXT     =JSR @.
132           1        .BLK     1
           6133 PUTBYTE     =JSR @.
133           1        .BLK     1
           6134 READBLOCK   =JSR @.
134           1        .BLK     1
           6135 RELJMPRET   =JSR @.
135           1        .BLK     1
           6136 STORDA      =JSR @.
136           1        .BLK     1
           6137 STINPUT     =JSR @.
137           1        .BLK     1
           6140 STOUTPUT    =JSR @.
140           1        .BLK     1
           6141 TRAPFAULT   =JSR @.
141           1        .BLK     1
           6142 WRITBLOCK   =JSR @.
142           1        .BLK     1
           6143 XGETBYTE    =JSR @.
143           1        .BLK     1
           6144 XPUTBYTE    =JSR @.
144           1        .BLK     1
           6145 SPINPUT     =JSR @.
145           1        .BLK     1
```

```
                        ;POINTERS USED ONLY WITHIN "REX" AND "SYSGEN"

 146        1 .BRKP:.BLK    1
 147        1 JFLTO:.BLK    1
 150        1 .FLTO:.BLK    1

 151        7 .BLK  160-.   ;OVERLAP CHECK AND PATCH SPACE

                        ;DECIMAL FLOATING-POINT REGISTERS

 160        1 DA:   .BLK    1
 161        1       .BLK    1
 162        1       .BLK    1
 163        1       .BLK    1
 164        1 DAC:  .BLK    1
 165        1 DAS:  .BLK    1

 166        1 DB:   .BLK    1
 167        1       .BLK    1
 170        1       .BLK    1
 171        1       .BLK    1
 172        1 DBC:  .BLK    1
 173        1 DBS:  .BLK    1

 174        1 .DA:  .BLK    1
 175        1 .DA3: .BLK    1
 176        1 .DB:  .BLK    1
 177        1 .DB3: .BLK    1

          174 C160 =.DA     ;THESE POINTERS USED AS CONSTANTS
          175 C163 =.DA3
          176 C166 =.DB
          177 C171 =.DB3

           77 C600 =.INFO

 200        0 .BLK  INFO-400-.      ;OVERLAP CHECK
              .EOT  ;PAGE ZERO FOR IRIS R8.1
```

```
- PAGE 22 -

; SOFTWARE DEFINITIONS FOR TAPE SOFTWARE FOR "IRIS" R8.1
; JULY 3, 1982

; MAGNETIC TAPE DRIVE TABLE (IN $MTA0, $MTA1, $CTU0, $CTU1, ETC.)
                   0 .DUSR CFN.  = 0     ;CURRENT TAPE FILE NUMBER
                   1 .DUSR CRN.  =CFN.+1 ;CURRENT TAPE RECORD NUMBER
                   2 .DUSR MDE.  =CRN.+1 ;CURRENT OPERATING MODE (SEE BELOW)
                   3 .DUSR SPS.  =MDE.+1 ;SPECIAL OPERATION STATUS
                   0 .DUSR       SPN = 0 ;NORMAL -- NO SPECIAL OPERATION
                   1 .DUSR       SPWS= 1 ;WRITING SEQUENTIAL
                   2 .DUSR       SPRS= 2 ;READING SEQUENTIAL
                   3 .DUSR       SPEF= 3 ;WRITE AN END-OF-FILE AFTER CURRENT OPERATION COMPLETES
                   4 .DUSR       SPRW= 4 ;REWIND AFTER CURRENT OPERATION COMPLETES
                   4 .DUSR STT.  =SPS.+1 ;CURRENT ERROR STATUS OF DRIVE
                   5 .DUSR RST.  =STT.+1 ;CURRENT "RAW" STATUS (NOT INTERPRETTED)
                   6 .DUSR CHA.  =RST.+1 ;CURRENT CHANNEL ADDRESS
                   7 .DUSR PHD.  =CHA.+1 ;POINTER INTO HEADER (0 IF CONTIGUOUS)
                  10 .DUSR PXH.  =PHD.+1 ;POINTER INTO EXTENDER (0 IF NOT EXTENDED)
                  11 .DUSR NWC.  =PXH.+1 ;WORD COUNT (NEGATIVE)
                  12 .DUSR MCN.  =NWC.+1 ;COUNTER
                  13 .DUSR USR.  =MCN.+1 ;CURRENT USER
                  14 .DUSR CMT.  =USR.+1 ;ADDRESS OF COMMAND TABLE IN $MTAS OR $CTUS
                  15 .DUSR ABF.  =CMT.+1 ;A(TAPE BUFFER) FOR THIS UNIT
                  16 .DUSR LAD.  =ABF.+1 ;LAST CONTROLLER MEMORY ADDRESS
                  17 .DUSR UNT.  =LAD.+1 ;DRIVE NUMBER
                  20 .DUSR CTP.  =UNT.+1 ; -1 IF MAG TAPE; PORT # IF CTU
                                         ;    (PCB ADDRESS AFTER .OPN.)
; THE VALUES FOR MDE ARE AS FOLLOWS:
;       0 --> IDLE
;       1 --> SPACING TO A FILE
;       2 --> SPCING TO A RECORD
;       3 --> INPUTTING DATA FROM TAPE
;       4 --> OUTPUTTING DATA TO TAPE
;       5 --> FILE INPUT FROM TAPE
;       6 --> FILE OUTPUT TO TAPE
;       7 --> WRITING OR READING AN END-OF-FILE
;      10 --> REWINDING
;      11 --> DATA INPUT COMPLETE. WAITING FOR CALLER TO RETRIEVE DATA.
;      12 --> INITIATING TAPE

; COMMAND TABLE DISPLACEMENTS IN $MTAS OR $CTUS

                   0 .DUSR NTPU   =0              ;NUMBER OF POSSIBLE DRIVE'S
                   1 .DUSR .OPN.  =NTPU+1  ;OPEN A GIVEN TAPE UNIT
                   2 .DUSR .CLS.  =.OPN.+1 ;CLOSE A TAPE UNIT
                   3 .DUSR .BFSZ  =.CLS.+1 ;A(TAPE BUFFER SIZE) IN BLKS
                   4 .DUSR KDEV   =.BFSZ+1 ;A(ABORT DEVICE SUBROUTINE)
                   5 .DUSR .SGD.  =KDEV+1  ;SIGNAL DONE ROUTINE
                   6 .DUSR .BSY.  =.SGD.+1 ;BUSY SUBROUTINE
                   7 .DUSR RDP.   =.BSY.+1 ;REGNANT DRIVE POINTER
                  10 .DUSR .RDT.  =RDP.+1  ;A(REGNANT DRIVE TABLE)
                  11 .DUSR .MWT.  =.RDT.+1 ;WRITE RECORD
                  12 .DUSR .MRD.  =.MWT.+1 ;READ RECORD
                  13 .DUSR .WEF.  =.MRD.+1 ;WRITE EOF
                  14 .DUSR .RWD.  =.WEF.+1 ;REWIND
                  15 .DUSR .SPC.  =.RWD.+1 ;SPACE FORWARD
                  16 .DUSR .INI.  =.SPC.+1 ;INITIALIZE A VIRGIN TAPE BY TRACK
```

```
; COMMAND TABLE ADDRESSES RELATIVE TO INTH IN $MTAS OR FINTH IN $CTUS

        177775 .DUSR ACMD =  -3       ;A($MTAS COMMAND TABLE) -1 IF NONE

; DISPLACEMENTS USED WITHIN $CTUS DRIVER (NEG TO CMDTBL)

          177777 .DUSR CINTH  = -1        ;A(INTH)
          177776 .DUSR CTISND = CINTH-1   ;A(CTU START INPUT)
          177775 .DUSR CTOSND = CTISND-1 ;A(CTU START OUTPUT)
          177774 .DUSR CTOSTR = CTOSND-1 ;A(CTU STORE OUTPUT BYTE)
          177773 .DUSR CTSEEK = CTOSTR-1 ;A(CTU SEEK ROUTINE)
          177772 .DUSR CTCOPY = CTSEEK-1 ;A(COPY CMD TO IO BUFFER)
          177771 .DUSR CTCDEC = CTCOPY-1 ;A(CONVERT A1 TO ASCII IN CMD BUF)
          177770 .DUSR CTCVCB = CTCDEC-1 ;A(CONVERT CBLK TO TRACK, BLK)

; CONTROL INFORMATION IN NON-INTERACTIVE PCB
              30 .DUSR AINT  =TON. ;A(INTH) FOR $MMUX

; THE FOLLOWING DISPLACEMENTS ARE IN A CONTROL NODE RETRIEVED AT .OPN. TIME
; AND SAVED IN THE PCB'S NLP. CELL.
                0 .DUSR ADRVT =A2    ;A(DRIVE CONTROL TABLE)
                1 .DUSR STATE =A1    ;CURRENT FUNCTION AND STATE OF IO TO PORT.
                                   ;  BITS  7 - 0 = CURRENT STATE(SEE BELOW)
                                   ;  BITS 15 - 8 = CURRENT FUNCTION(SEE BELOW)
                2 .DUSR SFIL  =A0    ;BLOCK # OF START OF FILE
;              NOTE THAT IN CTU DIRECTORY STARTING BLK # IS (SFIL+2)
;              IN ORDER TO IGNORE AN IRIS FILE HEADER.
                3 .DUSR NFBLK =A3    ;# OF BLOCKS IN THIS FILE
                                    ;-1 ==> VIRGIN TAPE
                4 .DUSR CBLK  =PC    ;CURRENT BLOCK # (THIS XFER)
                5 .DUSR BXFR  =CPRI ;# OF BLOCKS IN THIS XFER
                                   ;OR DEST FILE IF SPING
                6 .DUSR NFIL  =SSBA ;# OF FILES ON CTU TRACK #0
                                     ; = 32767 UNTIL TRACK #0 IS FULL
                7 .DUSR CTRK  =SDBA ;CURRENT TRACK NUMBER
               10 .DUSR CTRTY =TCBP ;READ OR WRITE RETRY COUNTER
               11 .DUSR ISTAT =TASK ;INPUT STATE AS FOLLOWS:
;             = -2  ;AUTO INPUT STATE (DATA OR DIR)
;             = -1  ;AWAITING CTU ENTRANCE TO CMD MODE
;             =  0  ;IDLE
;             =  1  ;INPUTTING CMD ECHO
;             =  6  ;INPUTTING STATUS VIA ^E(CR)
               12 .DUSR TBREC =PAUZ ;# TAPE BLKS PER RECORD
               13 .DUSR ACTP =QTMP ;A(PORT'S PCB)
```

```
; FUNCTIONS AND THEIR STATES ARE AS FOLLOWS:
         400 .DUSR CTURW=1*K          ; REWIND
           1         .DUSR RWCLX =1   ;ISSUE ^X TO CLEAR CTU CMD BUFFER
           2         .DUSR RWTRO =2   ;SELECTING TRACK #0
           3         .DUSR RWSEK =3   ;SEEKING TO BLOCK 10

        1000 .DUSR CTUSP=2*K          ;SPACE
           1         .DUSR SPTRO =1   ;TRACK SELECTION, TRACK #0
           2         .DUSR SPTR1 =2   ;TRACK SELECTION, TRACK #1
           3         .DUSR SPVBO =3   ;ISSUING VERIFY FOR DIR BLK #0
           4         .DUSR SPIBO =4   ;INPUTTING ADPI BUFFER (DIR BLK #0)
           5         .DUSR SPVB1 =5   ;ISSUING VERIFY FOR DIR BLK #1
           6         .DUSR SPIB1 =6   ;INPUTTING ADPI BUFFER (DIR BLK #1)
           7         .DUSR SPSEK =7   ;SEEKING

        1400 .DUSR CTUIN=3*K ;INITIALIZE
           1         .DUSR INIRO=1    ;TRACK SELECTION, TRACK #0
           2         .DUSR INCIO=2    ;INITIALIZING, TRACK #0
           3         .DUSR INTR1=3    ;TRACK SELECTION, TRACK #1
           3         .DUSR INCI1=3    ;INITIALIZING, TRACK #1

        2000 .DUSR CTUEF=4*K          ;WRITE END-OF-FILE
           1         .DUSR EFTRO=1    ;TRACK SELECTION, TRACK #0
           2         .DUSR EFTR1=2    ;TRACK SELECTION, TRACK #1
           3         .DUSR EFVBO=3    ;ISSUING VERIFY FOR DIR BLK #0
           4         .DUSR EFIBO=4    ;INPUTTING ADPI BUFFER (DIR BLK #0)
           5         .DUSR EFVB1=5    ;ISSUING VERIFY FOR DIR BLK #1
           6         .DUSR EFIB1=6    ;INPUTTING ADPI BUFFER (DIR BLK #1)
           7         .DUSR EFPBO=7    ;ISSUING PUT FOR DIR BLK #0
          10         .DUSR EFOBO=10   ;ISSUING FORCED WRITE (DIR BLK #0)
          11         .DUSR EFPB1=11   ;ISSUING PUT FOR DIR BLK #1
          12         .DUSR EFOB1=12   ;ISSUING FORCED WRITE (DIR BLK #1)

        2400 .DUSR CTURD=5*K          ;READ
           1         .DUSR RDVFY=1    ;ISSUING VERIFY COMMAND
           2         .DUSR RDINP=2    ;INPUTTING ADPI BUFFER
           3         .DUSR RDTR1=3    ;SELECTING TRACK #1
           4         .DUSR RDSEK=4    ;SEEKING TO BLOCK 10 ON TRACK #1
           5         .DUSR RDVBO=5    ;ISSUING VERIFY FOR DIR BLK #0
           6         .DUSR RDIBO=6    ;INPUTTING ADPI BUFFER (DIR BLK #0)
           7         .DUSR RDVB1=7    ;ISSUING VERIFY FOR DIR BLK #1
          10         .DUSR RDIB1=10   ;INPUTTING ADPI BUFFER (DIR BLK #1)
; NOTE: STATES 1&2 ARE REPEATED UNTIL BXFR IS EXHAUSTED.
;  IN THE CASE OF AN ERROR DURING AN XFER,

        3000 .DUSR CTUWR=6*K          ;WRITE
           1         .DUSR WRPUT=1    ;ISSUING PUT CMD AND OUTPUTTING DATA TO ADPI BUFFER
           2         .DUSR WROUT=2    ;ISSUING FORCED WRITE COMMAND
           3         .DUSR WRTR1=3    ;SELECTING TRACK #1
           4         .DUSR WRSEK=4    ;SEEKING TO BLOCK 10 ON TRACK #1

; NOTE: STATES 1&2 ARE REPEATED UNTIL BXFR IS EXHAUSTED.
;  IN THE CASE OF AN ERROR DURING XFER,

;       F=377, S=377       ;ABORT ALL ACTIVITY ON THIS PORT.
```

```
- PAGE 25 -

; DEFINE SPECIAL FLW BITS USED WITH CTU
           2000 .DUSR ISOV    = 2000    ;INPUT SERVICE OVERLOAD
            400 .DUSR KSOV    = 400     ;INT SRV OVERFLOW MUX STATUS
           4000 .DUSR KDCH    = 4000    ;DATA CHANNEL OVERLOAD
            400 .DUSR IENB    = 400     ;REENABLE SINGLE CHAR INPUT
                                        ;WHEN AUTO INPUT COMPLETE
          40000 .DUSR IBUFO   = 40000   ;ACTIVE AUTO OUT FROM ABF
                                        ;UPON CMD OUTPUT COMPLETE
            200 .DUSR OACT    = 200     ;OUTPUT ACTIVE

; MUX CONTROL WORDS
          20000 .DUSR OAUTO   = 20000   ;AUTO OUTPUT WITHOUT SPCL CHAR INT
          40000 .DUSR IAUTO   = 40000   ;AUTO INPUT WITHOUT SPCL CHAR INT


; SOME DEFINED CONSTANTS FOR USE WITH CTU
             12 .RDX   10
           1700 .DUSR CTETK = 960       ;LOGICAL END OF A CASSETTE TRACK
           3600 .DUSR CTEOT = CTETK*2 ;LOGICAL END OF CASSETTE TAPE


; SYMBOLIZE A CTU TAPE DIRECTORY ENTRY (SEE NEXT PAGE FOR COMPLETE DESCRIPTION)

              0 .DUSR CTFLO = 0         ;BYTES 1&2 OF NAME
;     CTFLO ALSO INDICATES THE TYPE OF CTU FILE AS FOLLOWS:
;             "Tn" ==> 1 TAPE FILE = 1 CTU FILE, n TAPE BLKS PER RECORD
;             "hh" ==> FILE #hh OF MULTIPLE CTU FILES PER TAPE FILE
;             "Xn" ==> LAST CTU FILE IN TAPE FILE, n TAPE BLKS PER REC
              1 .DUSR CTFL1 = CTFLO+1 ;BYTES 3&4 OF NAME (TAPE FILE #)
              2 .DUSR CSBL2 = CTFL1+1 ;BYTE 5 OF NAME (TRACK #)
                                      ;AND BYTE 2 OF STARTING BLK
              3 .DUSR CSBL1 = CSBL2+1 ;BTYE 1 OF STARTING BLK
                                      ;AND ADDITIONAL BLK COUNT
;NOTE:  THE STARTING BLK # IS 2 MORE THAN THE ACTUAL STARTING BLK ON
;       SO THAT MANIP WILL SKIP ANY POSSIBLE IRIS HEADERS.
              4 .DUSR DIRSZ = CSBL1+1 ;SIZE OF AN DIR ENTRY

             77 .DUSR NDIRENTRIES= 256/DIRSZ-1 ;# ENTRIES IN DIR
             10 .RDX   8
```

```
;DESCRIBE A CTU DIRECTORY UNDER $CTUS CONTROL
;
;In the discussion below the term "TAPE FILE" refers to a cohesive set of
;data that would be a regular magnetic tape file on a reel-to-reel tape
;unit. A "TAPE FILE" may be composed of several CTU files since a CTU file
;is restricted to a maximum of 256 256-byte tape records. For convenience
;of this discussion, a 256-byte CTU CTU tape record will be called a "BLK".
;Other blocks such as disc blocks will be spelled in the customary way.
;
;The first CTU blk used on a track is blk 10(decimal). A maximum of 960 blks
;are allowed on a single CTU track.
;
;Since there are no end-of-file gaps on a CTU, CTU file delineation is
;made via the CTU directroy. For $CTUS to access data from the CTU, the
;directory entries must follow $CTUS's file naming conventions.
;
;There are eight bytes per directory entry as follows:
;
;       bytes 1&2 identifies the CTU subfile as 1 of 3 types,
;       1. "Tn" implies that this CTU subfile is an entire simulated tape file.
;          "n" is a decimal digit in ASCII indicating the number of CTU blks
;          per simulated magnetic tape record (see example below).
;
;       2. "hh" implies that this subfile is subfile number hh of a multiple
;          subfile simulated magnetic tape file. "hh" is the hexadecimal
;          subfile sequence number in ASCII (see example below).
;
;       3. "Xn" implies that this is the last CTU subfile in a magnetic tape
;          file. "n" is a decimal digit in ASCII indicating the number of CTU
;          blks per simulated magnetic tape record. (see example below).
;
;       bytes 3&4 indicate the magnetic tape file number as two diecimal digits
;       in ASCII.
;
;       byte 5 indicates the CTU track number that the file resides on. Byte 4
;       is either "0" or "1".
;
;       bytes 7&6 form the binary starting blk number of the CTU subfile. Btye
;       7 is the most signif- wight bits and byte 6 is the least significant
;       icant eight bits. The actual starting blk number is this value minus 2.
;       The starting blk number is bias by two so that MANIP will ignore a IRIS
;       disc file header (512 bytes).
;
;       byte 8 is the binary additional blk count for the CTU subfile.
;         .
;
;
;Below are the directories for tracks 0 and 1 on a CTU containing the
;following magnetic tape files: (1) a 1000 byte file of 512 bytes per
;record, (2) a 245,000 bytes file of 256 bytes per record, and (3) a
;100 byte file of 256 bytes per record.

                        .EOT   ; "IRIS" R8.1 TAPE DEFINITIONS
```

```
; "CRTDEFS" == CRT MNEMONIC DEFINITIONS FOR "IRIS" R8.1

; 11-3-82

;                 All Rights Reserved
;      Copyright (C) 1978, Educational Data Systems
;      Copyright (C) 1982, POINT 4 Data Corporation
;    This document may not be reproduced without the
;  prior written permission of POINT 4 Data Corporation

          3       ET      =3        ;ETX code
          7       RB      =7        ;ring bell
         10       ML      =10       ;move left
         12       LF      =12       ;line feed
         13       VT      =13       ;vertical tab
         14       FF      =14       ;form feed
         15       CR      =15       ;carriage return
         17       MH      =17       ;move home
         20       CS      =20       ;clear screen
         21       S1      =21       ;special code 1
         22       S2      =22       ;special code 2
         23       S3      =23       ;special code 3
         24       S4      =24       ;special code 4
         40       MR      =40       ;move right
         41       RD      =41       ;read cursor position
         43       CU      =43       ;clear unprotected
;                 CL=44;clear to end of line (unprotected)
         45       CE      =45       ;clear to end of screen (unprotected)
         52       MD      =52       ;move down
         53       MU      =53       ;move up
         60       BB      =60       ;begin blink
         61       EB      =61       ;end blink
         62       BR      =62       ;begin reverse video
         63       ER      =63       ;end reverse video
         64       BD      =64       ;begin dimming
         65       ED      =65       ;end dimming
         66       BP      =66       ;begin write protect
         67       EP      =67       ;end write protect
         70       BU      =70       ;begin underline
         71       EU      =71       ;end underline
         72       BX      =72       ;begin expanded print
         73       EX      =73       ;end expanded print
         74       FM      =74       ;enter format mode
         75       FX      =75       ;exit format mode
         76       LK      =76       ;lock keyboard
         77       UK      =77       ;unlock keyboard
        100       BT      =100      ;begin transmission from CRT memory
        101       MP      =101      ;use memory pointer instead of cursor for next positioning.
        102       IL      =102      ;insert line
        103       DL      =103      ;delete line
        104       IC      =104      ;insert character
        105       DC      =105      ;delete character
        177       AT      =177      ;"@" -- start of multi-byte sequence
```

```
; FLAG DEFINITIONS IN TTN WORD

                20        EC      =20    ;EXPECTING.CURSOR
                40        ITIP    =40    ;INPUT.TRANSLATION.IN.PROGRESS
               100        OTIP    =100   ;OUTPUT.TRANSLATION.IN.PROGRESS
               200        ESCS    =200   ;ESCAPE.SEEN


                          .EOT    ;  "CRTDEFS" R8.1
```

```
---

AT        177    BB         60    BD         64    BINDI    6114    BINMU    6115
BP         66    BPI        16    BR         62    BSACF      72    BT        100
BU         70    BUMPU    6116    BX         72    C10        30    C100       44
C1000      64    C11        31    C12        32    C13        33    C14        34
C15        35    C16        36    C160      174    C163      175    C166      176
C17        37    C170K      21    C171      177    C177       45    C1777      65
C2          2    C20        40    C200       50    C2000      66    C205       51
C215       52    C240       53    C244       54    C260       55    C271       56
C3          3    C300       57    C334       60    C37        41    C377       61
C4         24    C40        42    C400       62    C4000      67    C5         25
C6         26    C600       77    C7         27    C77        43    C774C      22
C777       63    CALL     6100    CE         45    CHANN    6105    CM400      23
CR         15    CS         20    CU         43    DA        160    DAC       164
DAS       165    DATAP    6107    DB        166    DBA        75    DBC       172
DBS       173    DC        105    DECIM    6117    DL        103    DQUEU    6104
EB         61    EC         20    ED         65    EP         67    ER         63
ERRF       73    ESCF       70    ESCS      200    ET          3    ETSF       71
EU         71    EX         73    FF         14    FINDL    6122    FIX      6120
FLAGC    6101    FLOAT    6121    FM         74    FREEN    6106    FX         75
GETBY    6123    IC        104    IL        102    INBYT    6124    INSTB    6125
ISA2D    6126    ISA2L    6127    ITIP       40    JFLTO     147    LF         12
LK         76    LOADD    6130    MD         52    MH         17    ML         10
MP        101    MR         40    MU         53    OTIP      100    OUTBY    6131
OUTTE    6132    PUTBY    6133    QCHAR    6102    QUEUE    6103    RB          7
RD         41    READB    6134    RELJM    6135    RTP         7    RUP         5
RUS         6    S1         21    S2         22    S3         23    S4         24
SBA        74    SPINP    6145    STINP    6137    STORD    6136    STOUT    6140
TASKQ      15    TRAPF    6141    UK         77    VT         13    WRITB    6142
XGETB    6143    XPUTB    6144    .ABA       14    .BPS       76    .BRKP     146
.BSA       10    .DA       174    .DA3      175    .DB       176    .DB3      177
.FLTO     150    .HBA       11    .HXA       12    .INFO      77    .INTR     110
.LCM      113    .NRET     111    .SRET     112    .SSA       13
```

Appendix D
LPTD DRIVER FILE LISTING

This appendix contains the listing of the LPTD driver file to aid in the installation and configuration process.

```
ASM ,@8/L.LPTD.2256!,B050,-B051,B052
FEB 28, 1983  10:16:22


;       Batchfile:  R81JCL.LPTD

;       ; A = 2256
;       -R81DEFSPZP
;       R81LPTDSA
;                           << SI = R81LPTDSA; BO = 8/A.LPTD.2256! >>
;  "$LPT" == LINE PRINTER DRIVER FOR "IRIS" R8.1
;  FOR LPT ON DATA GENERAL 4060 MUX PORT
;  1-23-79
;
                 7          SFTYM=7;LPT BUFFER SAFETY MARGIN
                 1          .TXTM   1
             32200          .LOC    BPS
;
;
    32200 177777            -1        ;NO INTH
    32201  32624 .ATRB:ATRIB
    32202  32235            FINIS
    32203  32351            WR1TE
    32204 177777            -1        ;NO READ WR1TE
;
;********** INIT ROUTINE **********
;
    32205  54444 INIT:  STA     3,INPFL
    32206  32773        LDA     2,@.ATRB
    32207 151014        SKZ     2,2          ;FIRST INIT AFTER IPL?
    32210   1401        JMP     1,3          ;NO, JUST RETURN
;
;
    32211  34770 INIT1:LDA      3,.ATRB
    32212  21777        LDA     0,PTOFF,3
    32213   6100        CALL
    32214 100023        CPNPP                ;CHANGE PORT # TO PCB PNTR
    32215   6141        TRAPFAULT            ;ILLEGAL PORT #?
    32216 111000        MOV     0,2          ;AC2 NOW = PRINTER PORT PNTR
    32217  21027        LDA     0,AHA.,2     ;SIZE OF ACTIVE FILE
    32220 101014        SKZ     0,0          ;PORT INTERACTIVE ?
    32221    411        JMP     INERR        ;   YES, SHOULDN'T BE
    32222  52757        STA     2,@.ATRB     ;SAVE PCB = 1ST INIT DONE FLAG
    32223  34756        LDA     3,.ATRB
    32224  54523        STA     3,AATRB      ;SECONDARY PNTR TO ATRIB

;INIT COMPLETE: SET OUTPUT CHAR HANDLER ADDRESS FOR MUX

    32225   4425        JSR     INTH-1       ;GET ADDRESS, SKIP RETURN
    32226  20000 C20K:  20000
    32227  55030        STA     3,TON.,2     ;PUT IT IN PCB
    32230  34421        LDA     3,INPFL
    32231   1401        JMP     1,3          ;NORMAL RETURN

    32232  34402 INERR:LDA      3,ERR43
    32233   2416        JMP     @INPFL

    32234     43 ERR43:43                    ;INCORRECT ATTRIBUTES

;FINISHED (CLOSE) ROUTINE



ASM ,@8/L.LPTD.2256!,B050,-B051,B052
FEB 28, 1983  10:16:28

    32235  54513 FINIS:STA      3,RTNAD
```

```
- PAGE 2 -

;                             << SI = R81LPTDSA; BO = 8/A.LPTD.2256! >>
   32236  20413               LDA    0,INPFL
   32237 101014               SKZ    0,0        ;IS INIT STILL PENDING
   32240    403               JMP    FINI2      ;YES
   32241  20407               LDA    0,FINCL    ;NO
   32242   4447               JSR    QSTRI      ;SEND THEM
   32243   4503 FINI2:JSR    JPROD      ;"KICK" PRINTER JUST IN CASE
   32244 102400               SUB    0,0
   32245  40404               STA    0,INPFL    ;CLEAR INIT PENDING IF ON
   32246  40440               STA    0,WRICC    ;IN CASE OF USER ESC
   32247   2501               JMP    @RTNAD
;
   32250     23 FINCL:ATRIB-FINIZ
   32251      0 INPFL:0
;
;********** OUTPUT CHARACTERS ROUTINE **********
;
;ON ENTRY AC2=PCB
;
   32252   5401               JSR    1,3        ;LET MUX DO PFRST
   32253   1400 INTH:  JMP    0,3        ;NO INPUT ROUTINE
   32254  54426               STA    3,INTHR    ;START OF OUTPUT CHARACTER ROUTINE
   32255  21001               LDA    0,OCW.,2
   32256 101120               MOVZL  0,0
   32257 101220               MOVZR  0,0
   32260  41001               STA    0,OCW.,2 ;CLEAR MUX BUSY FLAG
   32261  25005               LDA    1,OBP.,2 ;OUTPUT BYTE POINTER
   32262  21004               LDA    0,IBP.,2 ;INPUT BYTE POINTER
   32263 106415               SNE    0,1        ;IS BUFFER EMPTY?
   32264    414               JMP    EXIT       ;YES, EXIT MUX RETURN
   32265  35003               LDA    3,LBA.,2 ;LAST BYTE ADDRESS
   32266 136033               ADCZ#  1,3,SNC    ;END OF BUFFER?
   32267  25002               LDA    1,FBA.,2 ;YES, WRAPAROUND
   32270 125400               INC    1,1        ;BUMP OUTPUT BYTE POINTER
   32271  45005               STA    1,OBP.,2
   32272  50411               STA    2,SPCB     ;SAVE PCB PNTR
   32273   6123               GETBYTE           ;GET NEXT CHAR INTO AC2
   32274 141000               MOV    2,0
   32275  30406               LDA    2,SPCB     ;RESTORE PCB PNTR
   32276   7032               JSR    @SND.,2    ;SEND CHAR TO MUX
   32277   2403               JMP    @INTHR     ;RETURN
   32300  10402 EXIT:  ISZ    INTHR
   32301   2401               JMP    @INTHR     ;EXIT RETURN TO MUX
;
   32302      0 INTHR:0
   32303      0 SPCB:  0
   32304      0 USC:   0
   32305     16 MARGN:SFTYM*2
   32306      0 WRICC:0
;
;********** QSTRING **********
```

```
;                         << SI = R81LPTDSA; BO = 8/A.LPTD.2256! >>
;
   32307      0       0
   32310      0       0
   32311  54777 QSTRI:STA     3,.-1
   32312  30435       LDA     2,AATRB
   32313 112400       SUB     0,2
   32314  50773       STA     2,QSTRI-2
   32315  22772 QSTR2:LDA     0,@QSTRI-2
   32316 101112       SSP     0,0       ;NEG. CHAR. TERMINATOR?
   32317   2771       JMP     @QSTRI-1  ;YES, DONE
   32320   4404       JSR     QUP       ;NO, MOVE CHAR
   32321  10766       ISZ     QSTRI-2
   32322    773       JMP     QSTR2
;
;
;********** QUP **********
;
;
; QUE UP CHAR IN ACO BY STORING IT IN CIRCULAR CORE BUFFER.
; IBP POINTS TO LAST CHAR STORED.
;
   32323      0       0
   32324  54777 QUP:  STA     3,.-1
   32325  32422       LDA     2,@AATRB ;POINTER TO PCB
   32326  25004       LDA     1,IBP.,2 ;BYTE PNTR OF LAST BYTE
   32327  35003       LDA     3,LBA.,2 ;LAST BYTE IN BUFFER
   32330 136033       ADCZ#   1,3,SNC  ;WRAPAROUND?
   32331  25002       LDA     1,FBA.,2 ;YES, GET FIRST BYTE PNTR
   32332 125400       INC     1,1
   32333  45004       STA     1,IBP.,2 ;SAVE NEXT BYTE ADDRESS
   32334  34767       LDA     3,QUP-1
   32335  14747       DSZ     USC      ;REDUCE USABLE SPACE
   32336    401       JMP     .+1
   32337   6133       PUTBYTE          ;PUT BYTE INTO BUFFER
   32340   2763       JMP     @QUP-1   ;RETURN
;
;
;
   32341      0 MULCR:0       ;MULTIPLE CR MODE FLAG (O=SET)
   32342     27 ERR27:27      ;RECORD IS LOCKED ERROR
   32343     31 ERR31:31      ;ITEM TYPES DON'T MATCH ERROR
   32344     60 C60:  60      ;ASCII ZERO
   32345    117 C117: 117     ;ASCII O (OH)
   32346    563 JPROD:JMP     PROD
   32347      0 AATRB:0
;
;
;********** WRITE **********
;
;
```

```
;                               << SI = R81LPTDSA; BO = 8/A.LPTD.2256! >>
;AC2 CONTAINS POINTER TO ICB. WRICC = # OF CHARS ALREADY HANHLED.
;IF LPT BUFFER IS OUT OF ROOM, ERROR RETURN BACK TO SYSTEM. THEN, ON
;REENTRY, WRICC > 0 MEANS IGNOR THIS # OF CHARS AS ALREADY HANDLED.
;
;
    32350      0 RTNAD:0
;
    32351  54777 WR1TE:STA    3,RTNAD
    32352  21002       LDA    0,2,2     ;GET TYPE
    32353  24031       LDA    1,C11
    32354 106414       SEQ    0,1       ;ITEM TYPE = STRING?
    32355    543       JMP    WRERR     ;NO, DON'T MATCH ERROR
    32356  25003       LDA    1,3,2     ;YES, GET STRING CHAR COUNT
    32357 124513       NEGL#  1,1,SNC   ;NEG OR ZERO COUNT
    32360    472       JMP    WR1T3     ;YES, EXIT DONE
    32361  20725       LDA    0,WRICC
    32362 101005       MOV    0,0,SNR   ;ANY CHARS PREV SENT?
    32363  44544       STA    1,CHSNT   ;NO, SET FOR DONE RETURN
    32364 106400       SUB    0,1       ;YES, ADJUST FOR THEM
    32365  44535       STA    1,WR1TQ   ;# OF CHARS REQUESTED FOR OUTPUT
    32366  25004       LDA    1,4,2     ;SOURCE BYTE PNTR
    32367 107000       ADD    0,1       ;ADJUST SOURCE BYTE PNTR
    32370  44533       STA    1,WR1TS   ;START OF SOURCE TO OUTPUT
    32371  32756       LDA    2,@AATRB
    32372  35003       LDA    3,LBA.,2  ;LAST BYTE PNTR
    32373  25002       LDA    1,FBA.,2  ;FIRST BYTE PNTR
    32374 136400       SUB    1,3       ;BUFFER SIZE
    32375  21005       LDA    0,OBP.,2  ;CURRENT OUT BYTE PNTR
    32376  25004       LDA    1,IBP.,2  ;CURRENT IN BYTE PNTR
    32377 122023       ADCZ   1,0,SNC   ;USABLE BUFFER SPACE
    32400 163000       ADD    3,0       ;ADJUST IF OBP IS TO LEFT OF IBP
    32401  24704       LDA    1,MARGN
    32402 122423       SUBZ   1,0,SNC   ;ENOUGH SPACE FOR ZERO & MARGIN?
    32403    511       JMP    WR1TX     ;NO, WAIT A BIT
    32404  40700       STA    0,USC     ;SAVE USABLE BUFFER SPACE
    32405  20644       LDA    0,INPFL
    32406 101015       SNZ    0,0       ;INIT STILL PENDING?
    32407    406       JMP    WR1T1     ;NO, CONTINUE
    32410  20515       LDA    0,INICL   ;YES, GET INIT CHARS OFFSET
    32411   4700       JSR    QSTRI     ;PUT CHARS INTO LPT BUFFER
    32412 102400       SUB    0,0
    32413  40636       STA    0,INPFL   ;CLEAR INIT PENDING FLAG
    32414  40725       STA    0,MULCR   ;RESET MULTIPLE CR MODE
;
;
;WR1T1 TRANSFERS CHARS FROM THE USERS BUFFER TO LPT'S BUFFER
;
;
    32415  20667 WR1T1:LDA    0,USC
    32416 101112       SSP    0,0       ;ANY USABLE SPACE LEFT?
```

```
;                                 << SI = R81LPTDSA; BO = 8/A.LPTD.2256! >>
   32417    475             JMP     WR1TX    ;NO, WAIT A BIT
   32420  24503             LDA     1,WR1TS  ;YES, FETCH SOURCE POINTER
   32421   6143             XGETBYTE          ;GET CHAR INTO AC2
   32422  20045             LDA     0,C177
   32423 143405             AND     2,0,SNR  ;IS CHAR A NULL?
   32424    426             JMP     WR1T3    ;YES, END OF STRING
   32425  24521             LDA     1,WR1TL  ;CHAR LOW RANGE
   32426  34521             LDA     3,WR1TH  ;CHAR HIGH RANGE
   32427 122432             SUBZ#   1,0,SZC  ;SKIP IF CHAR < LOW RANGE
   32430 162032             ADCZ#   3,0,SZC  ;SKIP IF CHAR <= HIGH RANGE
   32431    430             JMP     WR1T6    ;NO, GO CHECK IT
   32432  40707             STA     0,MULCR  ;YES, RESET MULTIPLE CR MODE
   32433  34714             LDA     3,AATRB
   32434  25776             LDA     1,EXOFF,3
   32435 125213             SKO     1,1      ;EXCHANGE O (OH) & ZERO?
   32436    407             JMP     WR1T4    ;NO
   32437  24706             LDA     1,C117   ;YES
   32440  30704             LDA     2,C60
   32441 106415             SNE     0,1      ;IS CHAR O (OH)?
   32442 145001             MOV     2,1,SKP  ;YES, SEND 0 (ZERO)
   32443 112415             SNE     0,2      ;IS CHAR ZERO?
   32444 121000             MOV     1,0      ;YES, SEND O (OH) INSTEAD
   32445   4657 WR1T4: JSR     QUP      ;SEND CHAR TO LPT BUFFER
   32446  10640 WR1T2: ISZ     WRICC
   32447  10454             ISZ     WR1TS
   32450  14452             DSZ     WR1TQ    ;DONE WITH REQUESTED STRING?
   32451    744             JMP     WR1T1    ;NO, LOOP BACK
   32452   4457 WR1T3: JSR     PROD     ;"KICK" THE PRINTER
   32453 102400             SUB     0,0
   32454  40632             STA     0,WRICC  ;CLEAR CHARS SENT COUNT
   32455  20452             LDA     0,CHSNT  ;GET CHARS SENT COUNT
   32456  34672             LDA     3,RTNAD
   32457   1401             JMP     1,3      ;GOOD RETURN TO SYSTEM
   32460    631 JQSTR: JMP     QSTRI
;
   32461  24035 WR1T6: LDA     1,C15    ;ENTRY NOT MIDRANGE
   32462 106414             SEQ     0,1      ;IS CHAR A CR?
   32463    415             JMP     WR1T5    ;NO, TEST FOR SPECIAL CHAR
   32464  30655             LDA     2,MULCR
   32465 151004             MOV     2,2,SZR  ;MULTIPLE CR MODE SET?
   32466    405             JMP     WR1T7    ;NO, GO SET IT
   32467  20032             LDA     0,C12    ;GET LF CHAR
   32470   4634             JSR     QUP      ;QUEUE IT UP
   32471 102400             SUB     0,0      ;AC0=NULL
   32472    753             JMP     WR1T4    ;QUEUE NULL & GO
;
   32473 152400 WR1T7: SUB     2,2
   32474  50645             STA     2,MULCR  ;SET MULTIPLE CR MODE
   32475  20427             LDA     0,WRCRL
   32476   4762             JSR     JQSTR    ;SEND CR CHARS INSTEAD OF CR
```

```
;                          << SI = R81LPTDSA; BO = 8/A.LPTD.2256! >>
  32477    747          JMP     WR1T2    ;YES, NOT FOUND - IGNORE IT
;
  32500   4450 WR1T5:JSR     WR1TP    ;AC3=PNTR TO SPECIAL CHAR LIST
  32501  25400        LDA     1,0,3
  32502 125112        SSP     1,1      ;END OF LIST (-1)?
  32503    743        JMP     WR1T2    ;YES, NOT FOUND - IGNORE IT
  32504 122415        SNE     1,0      ;CHAR IN LIST?
  32505    403        JMP     WR1T8    ;YES, PASS SPECIAL CHAR TO LPT
  32506 175400        INC     3,3      ;NO, KEEP CHECKING
  32507    772        JMP     WR1T5+1
;
  32510   4614 WR1T8:JSR     QUP      ;SEND CHAR TO LPT BUFFER
  32511  20415        LDA     0,NULST
  32512   4746        JSR     JQSTR    ;SEND NULLS (DELAY)
  32513    733        JMP     WR1T2    ;CONTINUE
;
  32514   4415 WR1TX:JSR     PROD     ;"KICK" THE PRINTER
  32515  24504        LDA     1,WR1TD  ;LOCKED RETURN DELAY
  32516  34624        LDA     3,ERR27  ;"RECORD IS LOCKED" ERROR
  32517   2631        JMP     @RTNAD   ;ERROR RETURN
;
  32520  34623 WRERR:LDA     3,ERR31  ;"ITEM TYPES DON'T MATCH" ERROR
  32521   2627        JMP     @RTNAD
;
  32522      0 WR1TQ:0                ;# OF CHARS REQUESTED IN WR1TE
  32523      0 WR1TS:0                ;BYTE ADDRESS OF SOURCE STRING
  32524     43 WRCRL:ATRIB-WRICR      ;OFFSET TO CR CHAR LIST
  32525     33 INICL:ATRIB-INITZ      ;OFFSET TO INIT CHAR LIST
  32526     13 NULST:ATRIB-NULLS      ;OFFSET TO NULLS LIST
  32527      0 CHSNT:0                ;CHARACTERS SENT CELL
;
;
;********** PROD ROUTINE **********
;
;PROD STIMULATES THE FIRST INTERUPT FROM THE LPT IN ORDER TO GET
;OUTPUT GOING. ALSO TO "KICK" LPTIN THE EVENT IT FAILS TO
;COMPLETE PRINTING, YET REMAINS READY.
;
;
  32530      0        0
  32531  54777 PROD:  STA     3,.-1
  32532  32615        LDA     2,@AATRB ;PNTR TO PCB FOR LPT
  32533  35005        LDA     3,OBP.,2
  32534  25004        LDA     1,IBP.,2
  32535 136415        SUB#    1,3,SNR  ;BUFFER EMPTY?
  32536   2772        JMP     @PROD-1  ;YES, EXIT
  32537  21001        LDA     0,OCW.,2
  32540 101102        MOVL    0,0,SZC  ;IS MUX BUSY?
  32541   2767        JMP     @PROD-1  ;YES, WAIT FOR INTERRUPT
  32542 102400        SUB     0,0      ;SEND A NULL
```

```
;                       << SI = R81LPTDSA; BO = 8/A.LPTD.2256! >>
   32543  32604          LDA     2,@AATRB ;SET AC2=PCB FOR MUX
   32544   7032          JSR     @SND.,2  ;SEND CHARACTER TO MUX
   32545   2763          JMP     @PROD-1
;
;
;********** TABLES AND WORKING STORAGE **********
;
;
   32546      40 WR1TL:40        ;LOWEST NON-SPECIAL ASCII CHAR
   32547     174 WR1TH:174       ;HIGHEST NON-SPECIAL ASCII CHAR
;
   32550    5400 WR1TP:JSR     0,3       ;GENERATE PNTR TO FOLLOWING LIST
   32551      14       14
   32552       0       0
   32553      12       12
   32554       0       0
   32555 177777       -1
   32556 177777       -1
   32557 177777       -1
   32560 177777       -1
;
;SEND THIS STRING IN PLACE OF CR
;
   32561      15 WRICR:15        ;CR CHAR LIST
   32562       0       0
   32563      12       12
   32564       0       0
   32565 177777       -1
   32566 177777       -1
   32567 177777       -1
   32570 177777       -1
;
                0       .DMR WRICZ=JMP WRICR+SFTYM+1-.;CR CHAR OVERFLOW TEST
;
;THE INIT CHARS ARE OUTPUT WHEN LPT IS OPENED
;
   32571      15 INITZ:15
   32572       0       0
   32573      14       14
   32574       0       0
   32575 177777       -1
   32576 177777       -1
   32577 177777       -1
   32600 177777       -1
;
                0       .DMR INIZZ= JMP INITZ+SFTYM+1-.;INITZ OVERFLOW TEST
;
;THE FINIZ CHARS ARE OUTPUT WHEN LPT IS CLOSED
;
   32601      15 FINIZ:15        ;CR FLUSHES THE LPT BUFFER
```

```
- PAGE 8 -

;                         << SI = R81LPTDSA; BO = 8/A.LPTD.2256! >>
   32602      0           0
   32603 177777           -1
   32604 177777           -1
   32605 177777           -1
   32606 177777           -1
   32607 177777           -1
   32610 177777           -1
;
              0           .DMR FINZZ= JMP FINIZ+SFTYM+1-. ;FINIZ OVFLO TEST
;
;
;TIMING STRING SENT AFTER ALL SPECIAL CHARS
;
   32611      0 NULLS:0
   32612      0       0
   32613      0       0
   32614 177777       -1
   32615 177777       -1
   32616 177777       -1
   32617 177777       -1
   32620 177777       -1
;
              0           .DMR NULLZ= JMP NULLS+SFTYM+1-. ;OVFLO TEST
;
   32621      3 WR1TD:3        ;LOCKED RETURN DELAY, ADJUST FOR MAX LPT SPEED
;
;
   32622      0 EXCHF:0
;
;
   32623      2 PORTN:2        ;LPT ASSIGNED TO PORT 1
   32624      0 ATRIB:0        ;PCB FURNISHED
   32625      0       0
   32626      0       0
   32627 177777       -1       ;LINKAGE POINTER TO TERMINATOR
   32630 177777       -1       ;NO PORT DEFINATION TABLE
;
         177776           EXOFF= EXCHF-ATRIB ;OFFSET TO EXCHANGE FLAG
         177777           PTOFF= PORTN-ATRIB ;OFFSET TO PORT NUMBER
;
;
                          .END
```

```
---

AATRB  32347   ATRIB  32624   BINDI   6114   BINMU   6115   BPI       16
BSACF     72   BUMPU   6116   C10       30   C100      44   C1000     64
C11       31   C117   32345   C12       32   C13       33   C14       34
C15       35   C16       36   C160     174   C163     175   C166     176
C17       37   C170K     21   C171     177   C177      45   C1777     65
C2         2   C20       40   C200      50   C2000     66   C205      51
C20K   32226   C215      52   C240      53   C244      54   C260      55
C271      56   C3         3   C300      57   C334      60   C37       41
C377      61   C4        24   C40       42   C400      62   C4000     67
C5        25   C6        26   C60    32344   C600      77   C7        27
C77       43   C774C     22   C777      63   CALL    6100   CHANN   6105
CHSNT  32527   CM400     23   DA       160   DAC      164   DAS      165
DATAP   6107   DB       166   DBA       75   DBC      172   DBS      173
DECIM   6117   DQUEU   6104   ERR27  32342   ERR31  32343   ERR43  32234
ERRF      73   ESCF      70   ETSF      71   EXCHF  32622   EXIT   32300
EXOFF 177776   FINCL  32250   FINDL   6122   FINI2  32243   FINIS  32235
FINIZ  32601   FIX     6120   FLAGC   6101   FLOAT   6121   FREEN   6106
GETBY   6123   INBYT   6124   INERR  32232   INICL  32525   INIT   32205
INIT1  32211   INITZ  32571   INPFL  32251   INSTB   6125   INTH   32253
INTHR  32302   ISA2D   6126   ISA2L   6127   JFLTO    147   JPROD  32346
JQSTR  32460   LOADD   6130   MARGN  32305   MULCR  32341   NULLS  32611
NULST  32526   OUTBY   6131   OUTTE   6132   PORTN  32623   PROD   32531
PTOFF 177777   PUTBY   6133   QCHAR   6102   QSTR2  32315   QSTRI  32311
QUEUE   6103   QUP    32324   READB   6134   RELJM   6135   RTNAD  32350
RTP        7   RUP        5   RUS        6   SBA       74   SFTYM      7
SPCB   32303   SPINP   6145   STINP   6137   STORD   6136   STOUT   6140
TASKQ     15   TRAPF   6141   USC    32304   WR1T1  32415   WR1T2  32446
WR1T3  32452   WR1T4  32445   WR1T5  32500   WR1T6  32461   WR1T7  32473
WR1T8  32510   WR1TD  32621   WR1TE  32351   WR1TH  32547   WR1TL  32546
WR1TP  32550   WR1TQ  32522   WR1TS  32523   WR1TX  32514   WRCRL  32524
WRERR  32520   WRICC  32306   WRICR  32561   WRITB   6142   XGETB   6143
XPUTB   6144   .ABA      14   .ATRB  32201   .BPS      76   .BRKP    146
.BSA      10   .DA      174   .DA3     175   .DB      176   .DB3     177
.FLTO    150   .HBA      11   .HXA      12   .INFO     77   .INTR    110
.LCM     113   .NRET    111   .SRET    112   .SSA      13
```

Appendix E
IRIS SYSTEM HALTS

Data Lights	Meaning of Halt
63077	Good halt from DDCOPY, etc.
63277	INSTALL has encountered a bad file (see the IRIS Operations Manual).
63377	Interrupt stack overflow. An interrupt mask hardware problem or some device's mask bit is incorrectly specified in its driver.
67077	Power fail restart. A Power Fail Auto Restart was attempted, and one or more disc drives require operator intervention. Ready all disc drives, and press CONTinue to resume system operation.
67277	FATAL memory management halt # 1, 2 or 3. Record the location of the halt and the values in each of the four accumulators and the carry state (1=on, 0=off).
67377	Interrupt not acknowledged. Some hardware device has interrupted but did not present its device code to an INTA instruction. This indicates a hardware problem, most likely improper routing of the interrupt priority line on the computer backplane (pins A95 and A96).
73077	Power fail halt. A power failure occurred, and there was a failure in the Power Fail Auto Restart hardware. Ready all disc drives, and press CONTinue to resume system operation.
73277	Reserved

Data Lights	Meaning of Halt
73377	Insufficient free nodes. Not enough free nodes available for task queuing. Do a minimum configuration IPL and increase the number of free nodes. Do a full configuration IPL.
77077	Double Trap. A trap occurred while attempting to print the trap message. After writing down all information, press CONTinue, then again examine and write down the contents of all registers and the carry state. The second set of information is from the initial trap.
77277	Attempted to do a TRAPFAULT when FAULT could not be executed. For example: ● In CLEANUP after DISCSUBS were relocated ● At IPL-time if there are more physical mux ports on the system than have been identified to IRIS in $MMUX's Port Definition Table. The accumulators contain: A0 = 25 (POINT 4 Mux device code) A2 = PCB address of the first port physically present but not configured in $MMUX.
77377	Unknown halt. Unused areas of memory are usually filled with 77377 halt instructions. Such a halt indicates an abnormal jump in the software.
xxxxxx	Any value other than above in the data lights. Most likely a power failure occurred and either the computer's power switch was not set to LOCK, or the computer does not have the Power Fail Auto Restart option. A power failure is indicated by 62677 octal at location zero. A START at zero should resume system operation (first ready all disc drives).

Appendix F
IRIS SYSTEM TRAPS

This appendix illustrates a typical trap and provides tables of trap numbers showing the associated messages and the contents of the registers.

A trap number may appear in a trap message in the form

```
    TRAP #n AT (location) IN (processor)
```

where:

```
        n - Trap number (see Tables F-1 thru F-4)
 location - address where fault was detected
processor - Filename of processor that was in memory at time
            fault was detected
```

For example,

```
    TRAP #3 AT 4410 IN COPY
```

indicates a disc error (data check error, seek error, data channel late, or address check error) which was not recoverable in 16 retries which occurred at location 4410 while running the COPY processor. In the case of a TRAP #3, the actual location in the calling processor (COPY in this example) is given in register A0.

In Tables F-1 thru F-4, the following notation is used:

```
    a( ) - memory address of
    d( ) - disc address of
    ==>  - implies
    X    - register contains no useful information
```

Trap numbers 100 and greater occur only during an IPL, an INSTALL, or a Sysgen; the computer halts after any such trap unless it determines that a retry is possible.

TABLE F-1. TRAPS FROM REX

Trap No.	Reason	Contents of Registers		
		A0	A1	A2
0	Any JUMP or JSR to loc. 0, @0, or any soft fault.	(indeterminate, refer to listing)		
1	Disc is write protected.	Return address	RDA	Memory address
2	No such disc drive.	Return address	RDA	Memory address
3	Irrecoverable disc error.	Return address	RDA	Memory address
4	Disc timed out.	Return address	RDA	Memory address
5	Illegal disc address (RDA).	Return address	RDA	Memory address
6	Disc busy before transfer started.	Return address	RDA	Memory address
7	Inactive or illegal LU.	Return address	RDA	Memory address
10	Illegal memory address in a disc transfer.	Return address	RDA	Memory address
11	Writing HBA; but given LU or RDA do not match values in HBA.	Return address	RDA	Memory address
12	Error detected by DATAPUMP in a file header.	Return address	RDA	Memory address
13	Not enough blocks on LU #0 to expand an active file as required.	No. of blocks needed	X	X
14	Discsub calls nested too deep.	DISCSUB keyword	stack limit	a(keyword)
15	No such discsub number.	DISCSUB Subroutine no.	177777	a(keyword)
16	Bad directory in an indexed file.	Directory flags		
17	DATAPUMP latching or unlatching error or attempt to modify an unlatched buffer.	Return address	X	Memory address
20	BASIC not on the system disc (LU #0).	X	X	X

TABLE F-1. TRAPS FROM REX (Cont)

Trap No.	Reason	Contents of Registers		
		A0	A1	A2
21	BASIC has been given the wrong file type. File type for BASIC must be 33702.	X	Actual file type	a(BASIC header)
22	RUN not on system disc (LU #0).	X	X	X
23	RUN has been given wrong file type. File type for RUN must be be 33602.	X	Actual file type	a(RUN header)
24	RUNMAT has been given wrong file type. File type for RUNMAT must be 33402.	X	Actual file type	a(RUNMAT header)
25	User area is too small for the necessary stacks. Should not occur.	Required end of user area	End of user area	Start of user area
26	Illegal DATAPUMP command word.	Return address	X	X
27	Reserved			
30	Channel # specified in a CLEAR or ALLCLEAR call is illegal.	Channel no.	X	X
31	Reserved			
32	OUTBYTE called while output was active.	Return address	Return address from OUTTEXT	a(offending port)
33	Reserved			
34	Processor has run 25.6 seconds overtime.	X	X	X
35	RUN has detected a bad token. Impossible statement code from clobbered partition.	X	X	X
36	Illegal priority given in a call to QUEUE.	Return address	X	a(node used by QUEUE)
37	Attempt to dequeue the scheduler.	Return address	X	X
40	An attempt was made to wake up or extract a node which was not on any queue.	Return address	X	a(node)

TABLE F-1. TRAPS FROM REX (Cont)

Trap No.	Reason	Contents of Registers		
		A0	A1	A2
41	An attempt to put the scheduler to sleep.	Pause value	a(scheduler task node) .	a(scheduler task node)
42	An attempt to return a node to the FREE chain which was still in a queue.	NSTS node	X	a(node)
43	Processor is requesting a partition size greater than can be supplied by the current system.	Requested partition size	maximum partition size	X
44	LOADUSER failed to select an area for the regnant user and end of area was reached.	0	X	0
45	$LUSR.LCM unable to swap Active File due to invalid LCM block # retrieved from Active File header.	Active File Size	Invalid LCM block # of Active File	RUS
46	Reserved			
47	RUP contents do not match user's PCB in partition area's table.	0	X	a(RUP.DFT)
50	BSACF set but BSA has been cleared.	X	X	X
51	Reserved			
52	LOADUSER has given away all of partition area.	0	X	a(partition entry)
53	Reserved			
54	Reserved			
55	Reserved			
56	I/O error on LCM	Block count	Block #	Error code: 4= I/O error 5= Power Fail 6= D.C. late 7= Illegal block 10= Time out
57	Entire file not on LCM	X	X	X

TABLE F-1. TRAPS FROM REX (Cont)

Trap No.	Reason	Contents of Registers		
		A0	A1	A2
60	Not enough blocks remaining on LCM.	LCM block #	FUDA	a(new range table entry)
61	$LCM not active.	X	X	X
62	Reserved			
63	A "BASIC" program is not in R8 format.	X	X	X
64	Reserved			
65	Reserved			
66	Block flagged as being in fixed buffers cannot be located in any fixed buffer.	Return address	X	X
67	Reserved			
70	Reserved			
71	Buffer pool has run out of buffers.	X	X	X
72	Reserved			
73	Reserved			
74	Reserved			
75	Reserved			
76	Reserved			
77	Reserved			

TABLE F-2. TRAPS FROM SIR

Trap No.	Reason	Contents of Registers		
		A0	A1	A2
100	More than 16 sectors per track.	NTRK	20	a(LUVAR)
101	System disc or account out of space.	X	X	X
102	Block is already marked in DMAP.	x	RDA of block being marked	X
103	INDEX has <2 or >128 blocks.	X	X	X
104	Config driver table >1 block in size.	X	X	X
105	Improper or illegal subroutine selected in residency list. (It is included when another is made resident.)	Subroutine no.	X	X
106	SCOPE not on disc or not a processor.	X	X	X
107	ACCOUNTS not on disc or not at block 3.	X	X	X
110	BYE not on disc or not a processor.	X	X	X
111	Insufficient memory as configured.	Memory address at overflow	X	Return address
112	PCA overlaps 'DBUG' or 32K.	a(end of port control area)	X	a(port control area)=PCA
113	No DISCSUBS, or DISCSUBS has a damaged header.	X	X	X
114	Two discsubs with same number.	X	X	Subroutine no.
115	Illegal discsub #.	X	X	Subroutine no.
117	Not enough memory for SYSGEN.	Memory request overflow	X	Return address
121	Insufficient memory for minimum configuration IPL.	X	X	X
122	Topword not there.	X	X	TOPW

TABLE F-2. TRAPS FROM SIR (Cont)

Trap No.	Reason	Contents of Registers		
		A0	A1	A2
123	Illegal value for EPS (End of Processor Storage).	X	X	X
124	PCA overlaps SIR.	a(end of SIR)	PCA=a(port control area)	Size of PCA
125	Negative patch space size.	Amount of overflow	ENDP	BPSP
127	Partition area (PSIZ) too small or too large (<4006 octal or >77406 words octal).	Partition size	Maximum size	Minimum size
130	LU (not 0) is too large (>65536 blocks); occurs during INSTALL.	FUDA	FUDA overflow	X
141	Can't allocate sufficient buffers.	0	X	X
142	Insufficient memory to allocate adequate number of pool buffers.	X	X	X
143	Active file size too large (>201 blocks, including header).	No. of blocks in active file	200	X
160	CONFIG driver uses assigned device code.	X	a(in CONFIG of driver)	X
161	FAULTPRINT not on disc or not a processor.	0	RDA of FAULTPRINT	33401
162	FAULTHISTORY not on disc or not contiguous.	X	X	X
163	Reserved			

TABLE F-3. TRAPS FROM SYSGEN

Trap No.	Reason	Contents of Registers		
		A0	A1	A2
100	More than 16 sectors per track.	NTRK	20	a(LUVAR)
101	Not enough room to allocate the disc blocks.	X	X	X
102	Block is already marked in MAP.	X	RDA of block being marked	X
122	Topword not there.	X	X	TOPW
124	PCA overlaps SIR.	a(end of SIR)	PCA=a(port control area)	size of PCA
125	Negative patch space size.	Amount of overflow	ENDP	BPSP
131	LU #0 is too large (>32768).	FUDA for LU/0	100000	X
176	Reserved			
177	Patches overlap SOV.	BPSP	ENDP	X

TABLE F-4. TRAPS FROM PLOAD AND SYSGEN

Trap No.	Reason	Contents of Registers		
		A0	A1	A2
116	Gap in DISCSUBS tape.	177777	0==> Gap in tape	X
126	Disc full reading discsubs.	177777	X	X

COMMENT SHEET

MANUAL TITLE IRIS Installation and Configuration Manual

PUBLICATION NO. SM-030-0009 REVISION 06

FROM: NAME/COMPANY: ____________________

BUSINESS ADDRESS: ____________________

CITY/STATE/ZIP: ____________________

COMMENTS: Your evaluation of this manual will be appreciated by POINT 4 Data Corporation. Notation of any errors, suggested additions or deletions, or general comments may be made below. Please include page number references where appropriate.

NO POSTAGE NECESSARY IF MAILED IN U.S.A.
Fold on Dotted Lines and Tape

NO POSTAGE
NECESSARY
IF MAILED IN
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 5755 SANTA ANA, CA.

POSTAGE WILL BE PAID BY ADDRESSEE:

POINT 4 Data Corporation
PUBLICATIONS DEPARTMENT
2569 McCabe Way
Irvine, CA 92714

CUT ON THIS LINE

S5268 1

POINT 4 DATA CORPORATION

2569 McCabe Way / Irvine, California 92714 / (714) 754-4114

LOTUS 700

DISC CONTROLLER USER MANUAL

POINT 4 DATA CORPORATION

POINT 4 DATA CORPORATION
2569 McCabe Way / Irvine, California 92714

LOTUS 700

DISC CONTROLLER
USER MANUAL

NOTICE

Every attempt has been made to make this reference manual complete, accurate and up-to-date. However, all information herein is subject to change due to updates. All inquiries concerning this manual should be directed to POINT 4 Data Corporation.

POINT 4 Data Corporation
2569 McCabe Way
Irvine, CA 92714
(714) 863-1111

REVISION RECORD

PUBLICATION NUMBER: HM-121-0012

Revision	Description	Date
A	Initial Release	6/15/80

LIST OF EFFECTIVE PAGES

Changes, additions, and deletions to information in this manual are indicated by vertical bars in the margins or by a dot near the page number if the entire page is affected. A vertical bar by the page number indicates pagination rather than content has changed.

Page	Rev	Page	Rev	Page	Rev
Cover	-				
Title	-				
ii thru ix	A				
1-1 thru 1-6	A				
2-1 thru 2-25	A				
3-1 thru 3-43	A				
A-1 thru A-2	A				
B-1 thru B-7	A				
Comment Sheet	A				
Mailer	-				
Back Cover	-				

PREFACE

This manual is for use by the system installation technician and the system programmer involved in the installation and software integration of the LOTUS 700 Disc Controller into a minicomputer system.

It includes features, hardware characteristics, installation procedures, configuration PROM coding, I/O instruction programming, and disc controller commands. Disc controller register functions, accumulator bit usage, disc controller commands, and controller and drive status bits are given in tabular form for ease of reference.

The appendices include a summary of accumulator formats and configuration chart examples.

Related manuals include:

Title	Document Number
POINT 4 User Reference Manual	HM-080-0003
LOTUS 700 Disc Controller Utilities and Diagnostics Manual	HM-121-0016

CONTENTS

APPENDICES

Section Title Page

FIGURES

TABLES

Section 1
INTRODUCTION

1.1 GENERAL DESCRIPTION

POINT 4 Data Corporation's LOTUS 700 Disc Controller offers economical, high-performance moving-head disc storage interface for the POINT 4 Computer and NOVA*-type minicomputer systems. The single-board design uses low-power Shottky and MSI logic, providing the highest level of performance and reliability. The controller occupies one slot in the processor chassis and all connections to the drives are made directly from the disc controller board.

The LOTUS 700 Disc Controller interfaces up to four storage module-type drives, at transfer rates up to 1.2 megabytes per second. Drives supported include: the CDC 9448, 9730, and 9760 series; Ampex; Century Data; Okidata; Kennedy and Fujitsu. The four possible drives interfaced may be a mixture of any of the drives supported. Automatic program load is always from the lowest-numbered, ready drive unit.

Data to or from computer memory is transferred in two-byte words using the Direct Memory Access (DMA) Data Channel of the computer. The 9.67 MHz drive data transfer rate translates into 1.209 megabytes per second or 1.65 microseconds per DMA cycle at the computer. The computer must be able to support this DMA transfer rate. An 18-word FIFO buffer is used to prevent Data Late conditions.

The LOTUS 700 Controller is software-compatible with the IRIS (Interactive Real-time Information System) Operating System used on the POINT 4 and many NOVA-type computers. Four data-out instructions supply the controller with information required to perform any operation. Sixteen operations (including read/write/verify, seek and disc formatting) may be specified using controller commands. Seven data-in instructions obtain status information from the controller. Disk utility and diagnostic programs are provided.

Figure 1-1 is a photograph of the LOTUS 700 Disc Controller.

*NOVA is a trademark of Data General Corporation

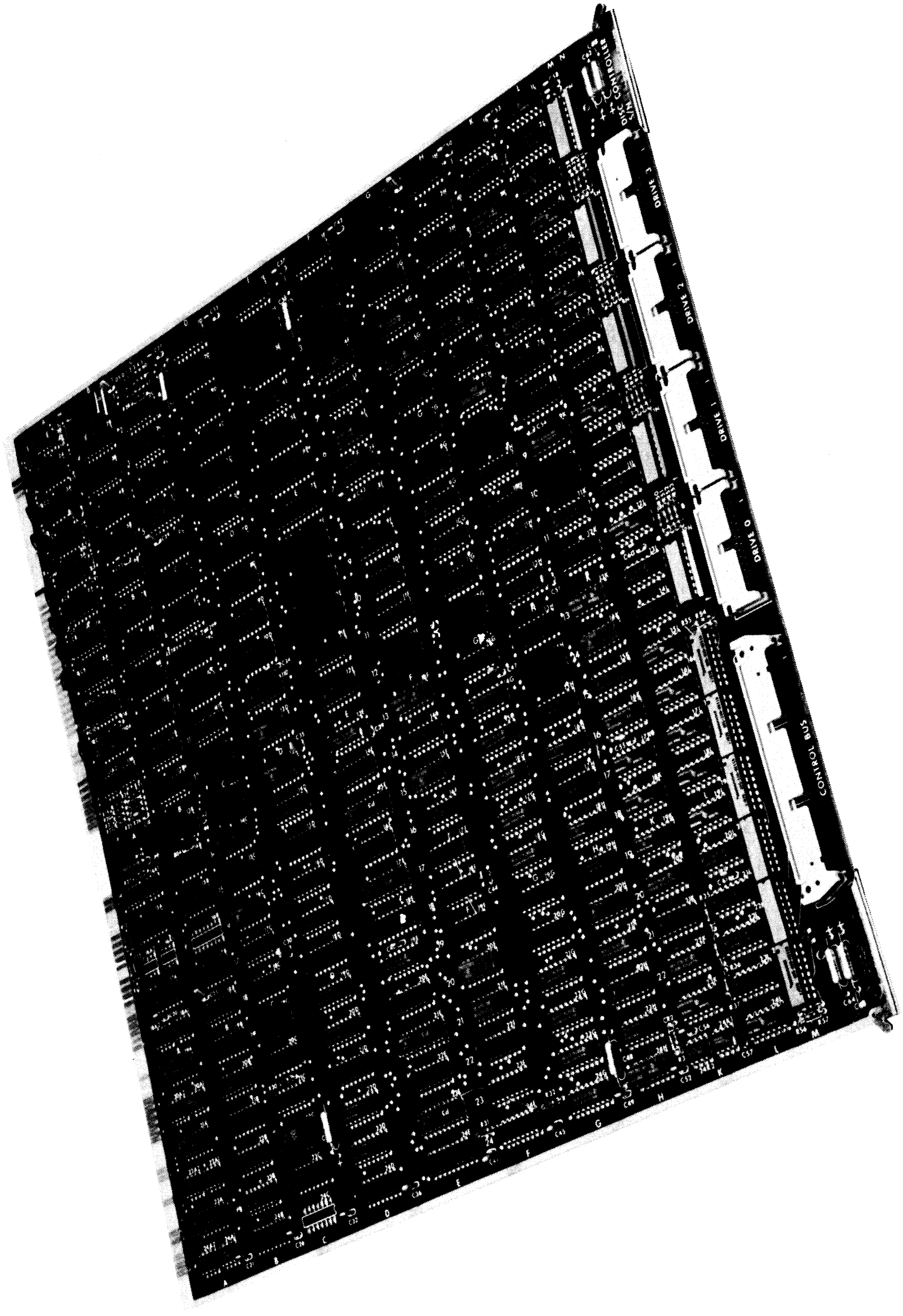


Figure 1-1. LOTUS 700 Disc Controller Board

1.1.1 Features

Many features make the LOTUS 700 Disc Controller a valuable addition to a POINT 4 Computer system. Compatibility, reliability, flexibility and advanced design are apparent in the following features:

- POINT 4 Computer I/O bus-compatible
- IRIS Operating System-compatible
- Interfaces up to four storage module-type drives in any combination
- Data transfer rate of 1.209 megabytes per second
- Whole track transfer in a single operation
- Format routine included in controller logic
- Overlapped seek
- 32-bit ECC error detection and provision for software correction
- Bad and alternate sectoring flags
- Supports dual port drives interfacing to two computers
- Built-in reliability and maintenance features
- Disc utility and diagnostic programs

1.1.2 Disc Controller Operation

The computer uses I/O instructions to transfer information between the processor's general purpose 16-bit accumulators and registers in the LOTUS 700 Disc Controller. Information transferred may be commands, computer memory addresses, disc addresses, controller or drive status, or ECCR codes. From this information the controller initiates seeks, reads and writes to and from the drive.

Addressing capacity per drive is 1024 tracks, 32 surfaces, and 32 sectors. Each sector contains 512 bytes of data. Up to 32 consecutive sectors may be transferred per operation with the cylinder boundary being crossed if necessary.

Hardware alternate sectoring and automatic retry are switch-selectable. Bad sector and alternate sector flags are provided to point out sectors known to be faulty. These flags are set during disc formatting. If the alternate sector flag is set, the alternate sector address is specified in the sector header. Alternate sectoring is done automatically requiring no program intervention.

Overlapped seek allows simultaneous seek and data transfers on multi-drive systems. Once a seek command has been issued to the drive, the LOTUS 700 Controller is ready to accept another command from the CPU. When the data transfer command is completed, a program interrupt request is issued to alert the CPU of the completed transfer.

Additional features include a 32-bit ECC error detection code for read or write operations. Error correction is programmable. In addition, system flexibility is increased by use of a programmable device code.

1.2 INPUT/OUTPUT INTERFACE

The LOTUS 700 Controller is designed to operate on POINT 4 and NOVA-type computers which are compatible with the specifications listed under Computer Interface. Disc drives controlled by the LOTUS 700 Controller operate under the specifications listed under Disc Drive Interface.

COMPUTER INTERFACE	
I/O Bus	POINT 4 and NOVA-type Computer I/O bus-compatible
Backplane wiring	None required
I/O bus loading	Single 7400-type input load; Single 75453-type output driver
Device Code	Programmable
Priority mask bit	7; 8 is optional
DMA transfer rate	1.209 Megabytes/sec (1.65 microsecond/DMA cycle)**
DISC DRIVE INTERFACE	
Type of interface	Storage module drive-compatible
Drives per controller	4 maximum
Drive type and size	Any mixture
Number of surfaces per drive	32 maximum*
Number of tracks per surface	1024 maximum*
Number of sectors per track	32 maximum*
Data transfer rate	9.67 MHz maximum**
Access time	1/2 revolution average, 1 revolution maximum*
Sector size: Header	6 bytes + 2 bytes of CRC
Data	512 bytes + 4 bytes of ECC
Number of consecutive sectors transferable in one operation	32 maximum
Bad sector flag	1st bit of header
Alternate sector flag	2nd bit of header
Alternate sector location	Any; specified in header
Program load	From lowest numbered ready drive
FIFO buffer size	18 words
Overlap seek execution	Yes
Alternate sectoring	Yes
ECC error detection	Yes
ECC error correction	Provision for software correction

* Function of Disc Drive
**From SMD Specification

1.3 SPECIFICATIONS

POINT 4 Data Corporation's LOTUS 700 Disc Controller package includes: the 700 Controller board, the cable set to the first drive, a disc utility program for formatting and surface analysis, and a diagnostic program. Physical, electrical and environmental specifications for the controller follow:

PHYSICAL	
700 Dimensions	Single board, 15"x15"
Cabling to drives	A Cable: 30-pair control bus; daisy-chained; terminated at last drive B Cables: 26-wire flat cable; radial; one per drive
Cable Connector	Located at front of controller board
POWER REQUIREMENTS (maximum)	
Current	+5 ±5%, 3.8A maximum -5 ±5%, 0.8A maximum
Power	23W
OPERATING ENVIRONMENT	
Operating Temperature	0-55C
Relative Humidity	0-90% noncondensing

Section 2
HARDWARE INTERFACE

2.1 INTRODUCTION

This section covers those hardware characteristics for which an understanding is essential to the user for installation and programming of the LOTUS 700 Controller. Covered in this section are:

- Controller architecture
- Controller registers used in I/O programming
- Installation and cabling
- Input/Output cable signal charts
- Hardware-selectable options
- Disc I/O system configuration

2.2 LOTUS 700 CONTROLLER ARCHITECTURE

The LOTUS 700 Controller serves as an interface between the processor and up to four disc drives. Figure 2-1 is a block diagram of the 700 Controller logic.

Interface to the processor is via the processor I/O DMA bus. Information passed between the processor and the controller includes:

- Data
- DMA transfer control signals
- Status information
- I/O instructions
- Device select signals

Internal processing which must take place within the controller includes:

- Command decoding
- Memory addressing
- Data buffering
- ECC code generation
- Head and sector PROM decoding
- Cylinder/alternate cylinder addressing
- Controller and drive status
- Controller storage registers
- Word and sector counting
- Header, data and sector comparison
- Sync detection
- Seek status detectors

Interface to the disc drives is via the Drive Cables (B) and the Control Bus Cables (A). Drivers and receivers control transfer of signals from each drive cable and the Control Bus to internal controller logic.

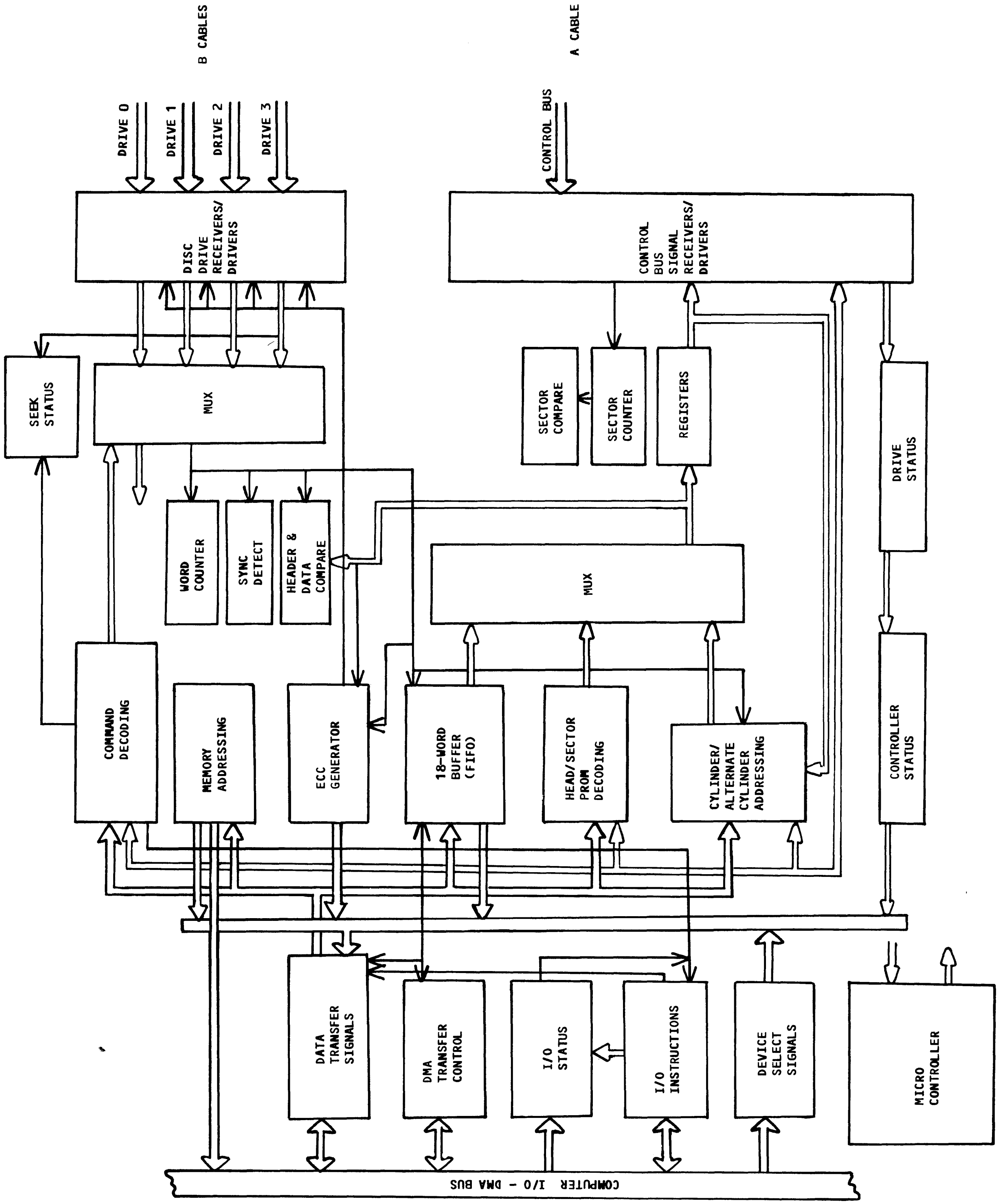


Figure 2-1. LOTUS 700 Disc Controller Block Diagram

2.2.1 Controller Registers

The LOTUS 700 Controller has a series of registers which are used for storage of information pertinent to the data transfer operation. This information includes addresses, status, commands, and the error correction code remainder. These registers perform important functions during the transfer of information in and out of the processor via I/O instructions. The registers and their functions are listed below.

Register	No. of Bits	Function
COMMAND	4	Holds the disc controller command, specified in bits 5-8 of the specified processor accumulator. The command is transferred to the controller by the DOA (Specify Drive and Command) instruction. There are 16 commands which control disk drive operation, ranging from 0000 to 1111 in bit arrangement. See Section 3.2.1 for specific descriptions of controller commands.
DRIVE ADDRESS	2	Holds the disc drive address, specified in bits 9 and 10 of the specified processor accumulator. The drive address is transferred to the controller by the DOA (Specify Drive and Command) instruction. The drive address is a two-bit programmable number. Only four drives may be connected to one LOTUS 700 Disc Controller.
MEMORY ADDRESS	16	Holds the processor memory address to which or from which the data transfer will take place. The 16-bit address is transferred to the controller's Address Register from the specified processor accumulator by the DOB (Specify Memory Address) instructions. The DIA instruction in Alternate Mode 1 reads the memory address from the Memory Address Register into the specified processor accumulator.

Register	No. of Bits	Function
CYLINDER ADDRESS	10	Holds the disc sector address of the sector involved in the data transfer operation. This address is transferred into the controller's Cylinder Address Register from bits 6-15 of the specified processor accumulator by the DOC (Specify Cylinder) instruction when a seek command has been specified by the previous instruction.
VOLUME	1	Holds the volume select bit which defines the non-removable surface(s) in CMD drives. This bit is transferred into the controller's Volume Register from bit 11 of the specified processor accumulator by the DOA (Specify Command and Drive) instruction, if the drive addressed is a CMD drive.
SURFACE ADDRESS	5	Holds the surface address of the sector involved in the data transfer. This address is transferred into the controller's Surface Address Register from bits 1-5 of the specified processor accumulator by the DOC instruction when the previous DOA instruction specified other than a seek command. This address is read from the Surface Address register into bits 1-5 of the specified processor accumulator by the DIC instruction.

Register	No. of Bits	Function
SECTOR ADDRESS	5	Holds the sector address of the sector involved in the data transfer. The sector address is transferred into the controller's Sector Address Register from bits 5-10 of the specified processor accumulator by the DOC instruction if the previous DOA instruction specified other than a seek command. This address is read from the Surface Address Register into bits 5-10 of the specified processor accumulator by the DIC instruction.
SECTOR COUNT	5	Holds the two's complement of the number of sectors to be transferred. The sector count is transferred into the controller's Sector Count Register from bits 11-15 of the specified processor accumulator by the DOC instruction if the previous DOA instruction specified other than a seek command. The sector count is read from the Sector Count Register into bits 11-15 of the specified processor accumulator by the DIC instruction.
ECCR	32	Holds the 32-bit error correction code remainder (ECCR) which is used to detect and correct certain data errors. The high-order 16 bits of the controller's ECCR Register are read into the specified processor accumulator by a DIA instruction in Alternate Mode 2. The low-order 16 bits of the controller's ECCR Register are loaded into the specified processor accumulator by a DIB instruction in Alternate Mode 2.

Register	No. of Bits	Function
CONTROLLER STATUS	15	Holds the controller's status flags. There are 15 flags which, when set to one (1), identify error conditions related to data transfer operations. See Subsection 3.2.5.1 for descriptions of error conditions. The controller status information is read from the Controller Status Register into the specified processor accumulator by the DIA (Read Controller Status) instruction.
DRIVE STATUS	10	Holds the drive status information of the selected disc drive. There are four status flags and six error condition flags which are set to one (1) to report on drive status. See Subsections 3.2.7 and 3.2.7.1 for descriptions of status and error conditions. The drive status information is read from the controller's Drive Status Register to the specified processor accumulator by the DIB (Read Drive Status) instruction.
EXTENDED MEMORY ADDRESS (optional)	4	Holds the optional extended memory address used when memory mapping is required to access all of system memory. The DOB (Specify Memory Address) instruction causes the transfer of 4 bits of XMA information from the controller's Auxiliary Register into the Extended Memory Address Register. The DIB instruction in Alternate Mode 1 reads the XMA information from the controller's Extended Memory Address Register into bits 12-15 of the specified processor accumulator.

Register	No. of Bits	Function
AUXILIARY	4	The controller's Auxiliary Register is used for temporary storage of the XMA information transferred to the controller from bits 12-15 of the specified AC by the DOA (Specify Command and Drive) instruction until the memory address for the operation specified in the DOA instruction is transferred into the Memory Address Register. When the memory address is transferred to the controller by a DOB instruction the 4 XMA bits are transferred from the Auxiliary Register to the Extended Memory Address Register.

2.3 CONTROLLER INSTALLATION

The LOTUS 700 Controller is a single-board design occupying only one slot in the processor chassis. The disc controller normally occupies the slot nearest the CPU board in the processor chassis.

The controller board should be added to the chassis with caution, making sure the card edge connector slides smoothly into the backplane sockets. No backplane wiring is required.

Cabling between the controller board and the disc drives consists of a control bus daisy-chained between drives and a separate drive cable to each drive. Cable descriptions are as follows:

Cable	Function	Description
A	Control Bus	30-pair (60-pin) control bus, daisy-chained between drives, terminated at the last drive
B	Drive Cables	26-wire flat cable, radial one per drive

Connections to the controller are made at the front of the controller board.

One end of the Control Bus Cable mounts at the 60-pin connector on the front of the LOTUS 700 Controller board. The other end mounts to the control bus input connector on the first disc drive. If there is more than one drive on the system, control bus cables must be connected between the output connector on each drive and the input connector on the next drive. The output connector on the last drive must be fitted with a bus terminator.

Drive cables are connected on the LOTUS 700 board to the 26-pin connector adjacent to the Control Bus connector and the Drive 3 Cable at the 26-pin connector furthest from the Control Bus connector. The opposite end of the drive cables connect to the 26-pin drive connectors on each respective drive.

NOTE

A ground strap must be strung between the drives at the ground connector and attached to the computer chassis.

Figure 2-2 is an illustration of controller-to-drive cabling.

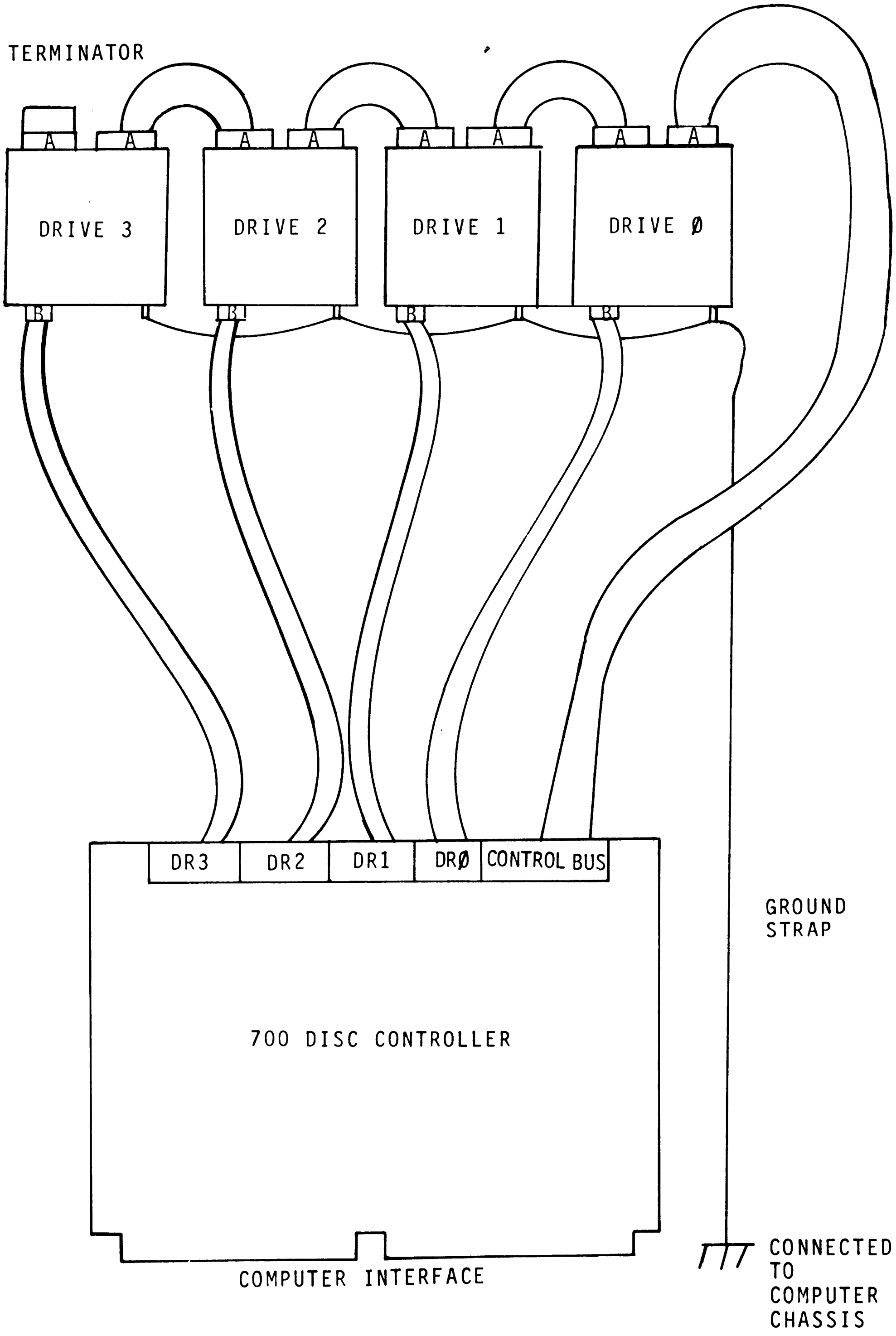


Figure 2-2. Controller-to-Drive Cabling

2.3.1 Cable Interface Signals

Successful system operation requires both that cabling is properly installed and that cable signals are properly configured. Tables 2-1 and 2-2 list signal configuration of the cables provided with the LOTUS 700 Controller. Check documentation provided with disc drives to be connected to the controller for verification that drive I/O interface signals correspond to control bus and drive cable connector signals.

Connector signal configurations are provided on the following tables:

Table 2-1. Control Bus Signals, A Cable

Table 2-2. Drive Cable Signals, B Cable

TABLE 2-1. CONTROL BUS SIGNALS
A CABLE

Controller/Drive 60-Pin Flat Cable Pin #	Signal Name	Optional 75-Pin Trailing Cable (available on some drives) Pin #
1 31	- + Tag 1	46 49
2 32	- + Tag 2	48 51
3 33	- + Tag 3	52 55
4 34	- + Bit 0	23 26
5 35	- + Bit 1	24 27
6 36	- + Bit 2	28 31
7 37	- + Bit 3	29 32
8 38	- + Bit 4	30 33
9 39	- + Bit 5	34 37
10 40	- + Bit 6	35 38
11 41	- + Bit 7	36 39
12 42	- + Bit 8	40 43
13 43	- + Bit 9	41 44
14 44	- + Open Cable Detect	16 20

Controller/Drive 60-Pin Flat Cable Pin #	Signal Name	Optional 75-Pin Trailing Cable (available on some drives) Pin #
15 45	- Fault +	11 14
16 46	- Seek Error +	75 78
17 47	- On Cylinder +	15 18
18 48	- Index +	10 13
19 49	- Unit Ready +	17 21
20 50	- (Address Mark Found) +	42 45
21 51	- Unit Reserved (Busy) +	47 50
22 52	- Unit Select Tag +	22 25
23 53	- Unit Select 2^0 +	1 4
24 54	- Unit Select 2^1 +	2 5
25 55	- (Sector) +	74 77
26 56	- Unit Select 2^2 +	3 7
27 57	- Unit Select 2^3 +	8 12
28 58	- Write Protected +	53 56
29 59	- PICK - HOLD	73 76
30 60	- Spare +	54 57

TABLE 2-2. DRIVE CABLE SIGNALS B CABLE

Controller/Drive 26-Pin Flat Cable Pin #	Signal Name	Optional 32-pin Trailing Cable Pin #
1	GND	K
14 2	+ - Servo Clock	N M
15	GND	T
3 16	- + Read Data	U V
4	GND	Y
17 5	+ - Read Clock	X W
18	GND	E
6 19	- + Write Clock	H J
7	GND	D
20 8	+ - Write Data	B A
21	GND	C
9 22	+ - Unit Selected	BB DD
10 23	- + Seek End	AA CC
11	GND	LL
24 12	+ - (Index Mark)	HH EE
25	GND	NN
13 26	- + (Sector Mark)	FF JJ

2.4 HARDWARE-SELECTABLE OPTIONS

Three options are selectable via etch cuts/jumpering on the LOTUS 700 Controller board. These options are:

- Automatic Retry
- Mask Bit
- Device Address

Selection of these options is explained in the following subsections.

2.4.1 Automatic Retry Option

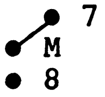
Automatic Retry after an unsuccessful seek operation is enabled by an etch located on the LOTUS 700 Controller board at coordinate C/9. It is identified by "RETRY" on the PC board. The two feed-through-holes to the right of RETRY are connected by an etch. This etch enables the Automatic Retry option. The option appears as follows on the PC board:

RETRY ●—●

To disable this option, cut the etch.

2.4.2 Mask Bit Option

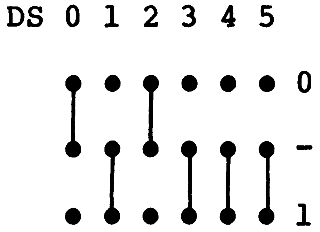
The LOTUS 700 Controller offers the option of using either bit 7 or bit 8 as the mask bit. Bit 7 is standard and bit 8 is optional. The mask bit option is located between column C and B and between rows 13 and 14. It is identified by an "M" on the PC board. The feed-through-hole directly to the left of the "M" enables the mask bit. It is etched to the feed-through-hole by the "7" as the mask bit. The option appears as follows on the PC board:



To select bit 8 as the mask bit, cut the etch between "7" and "M" and make a connection between "M" and "8".

2.4.3 Device Address Option

The device address for the 700 Controller is established by a 6-bit etch located at coordinate C/1 on the PC board. The existing etch on the delivered PC board is set to device code 27 (octal). The existing etch appears as follows:



To change the controller device address cut the existing etches and jumper the center feed-through-holes to 0 (zero) or 1 (one) as required for the new octal address.

2.5 DISC I/O SYSTEM CONFIGURATION

The LOTUS 700 Controller features flexibility in system configuration by allowing any mixture of compatible drives to be attached to a controller. This flexibility is made possible through the use of two configuration PROMs which must be programmed with control information for the attached drives.

The Head (Surface) PROM (22C on the 700 board) and the Sector PROM (19C on the 700 board) provide information within the controller about the type and size of the attached drives. These PROMs flag illegal surface and/or illegal sector numbers supplied by the CPU for data transfer operations. They also provide information on sector and/or surface overflow in multisector transfers.

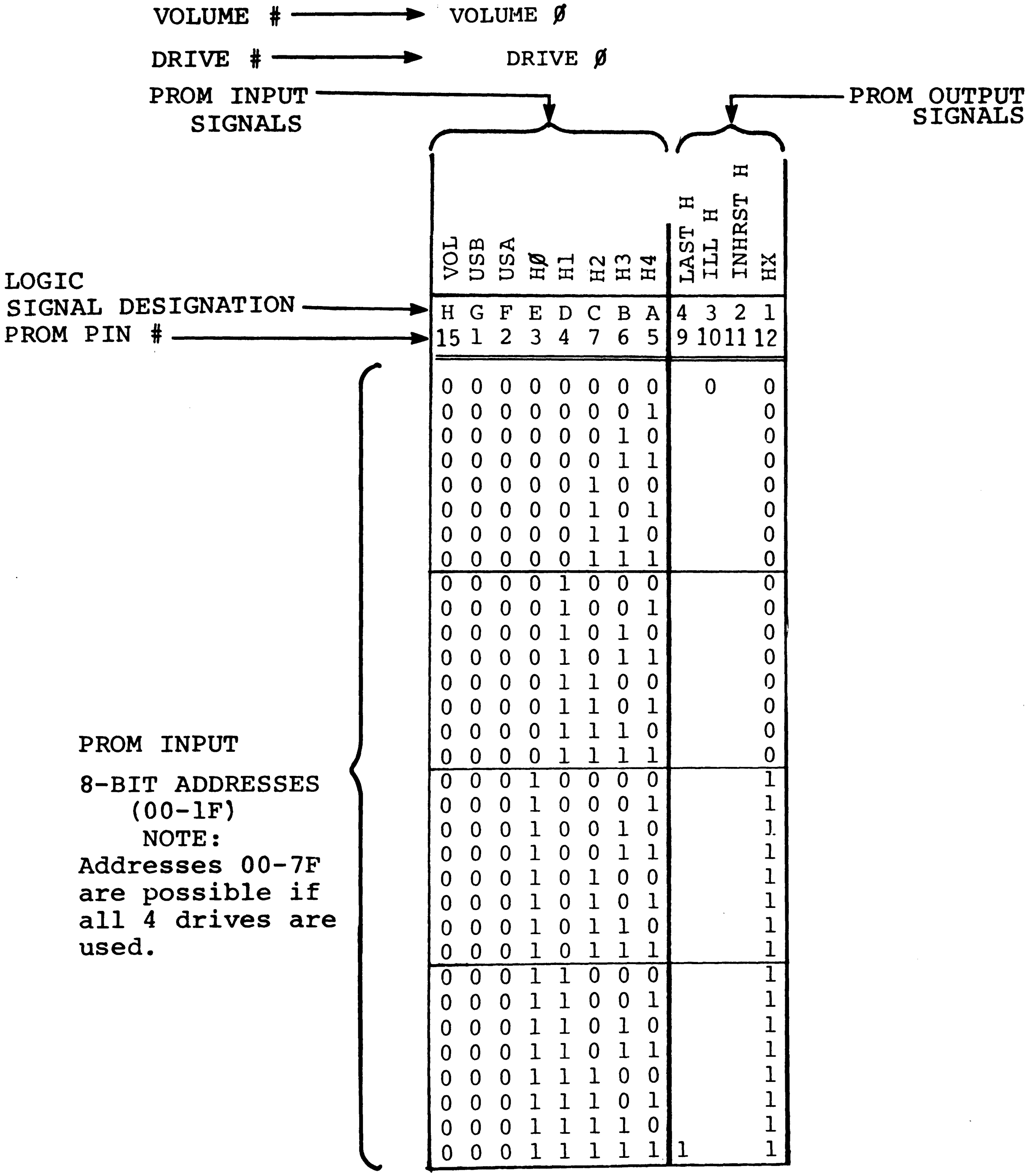
All LOTUS 700 Disc Controllers ordered from POINT 4 Data will require submission of Head PROM and Sector PROM Configuration Charts for the system configuration in which the controller is to be used. POINT 4 Data will program the Head and Sector PROMs according to these configuration charts.

The following two sections are provided for LOTUS 700 users who find it necessary to reconfigure a system after receipt of the 700 Controller board. Instructions for coding the information necessary to program the PROMs are provided. Appendix B contains examples of charts coded for specific disc drives.

2.5.1 Head PROM Configuration Chart Coding

The Head PROM Configuration Chart must be prepared to specify head information on each disc drive in the system.

The Head PROM Configuration Chart can be interpreted as follows:



NOTE: Addresses 00-7F are possible if all 4 drives are used.

VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX
H	G	F	E	D	C	B	A	4	3	2	1
15	1	2	3	4	7	6	5	9	10	11	12
0	0	0	0	0	0	0	0		0		0
0	0	0	0	0	0	0	1				0
0	0	0	0	0	0	1	0				0
0	0	0	0	0	0	1	1				0
0	0	0	0	0	1	0	0				0
0	0	0	0	0	1	0	1				0
0	0	0	0	0	1	1	0				0
0	0	0	0	0	1	1	1				0
0	0	0	0	1	0	0	0				0
0	0	0	0	1	0	0	1				0
0	0	0	0	1	0	1	0				0
0	0	0	0	1	0	1	1				0
0	0	0	0	1	1	0	0				0
0	0	0	0	1	1	0	1				0
0	0	0	0	1	1	1	0				0
0	0	0	0	1	1	1	1				0
0	0	0	1	0	0	0	0				1
0	0	0	1	0	0	0	1				1
0	0	0	1	0	0	1	0				1
0	0	0	1	0	0	1	1				1
0	0	0	1	0	1	0	0				1
0	0	0	1	0	1	0	1				1
0	0	0	1	0	1	1	0				1
0	0	0	1	0	1	1	1				1
0	0	0	1	1	0	0	0				1
0	0	0	1	1	0	0	1				1
0	0	0	1	1	0	1	0				1
0	0	0	1	1	0	1	1				1
0	0	0	1	1	1	0	0				1
0	0	0	1	1	1	0	1				1
0	0	0	1	1	1	1	0				1
0	0	0	1	1	1	1	1	1			1

A sample of a Head PROM Configuration Chart format can be found in Figure 2-3.

VOLUME Ø DRIVE Ø												DRIVE 1												DRIVE 2												74S287 DRIVE 3											22C
VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX	VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX	VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX	VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX
H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1
15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12
0	0	0	0	0	0	0	0		0		0	0	0	1	0	0	0	0	0		0		0	0	1	0	0	0	0	0	0		0		0	0	1	1	0	0	0	0	0		0		0
0	0	0	0	0	0	0	1				0	0	0	1	0	0	0	0	1				0	0	1	0	0	0	0	0	1				0	0	1	1	0	0	0	0	1				0
0	0	0	0	0	0	1	0				0	0	0	1	0	0	0	1	0				0	0	1	0	0	0	0	1	0				0	0	1	1	0	0	0	1	0				0
0	0	0	0	0	0	1	1				0	0	0	1	0	0	0	1	1				0	0	1	0	0	0	0	1	1				0	0	1	1	0	0	0	1	1				0
0	0	0	0	0	1	0	0				0	0	0	1	0	0	1	0	0				0	0	1	0	0	0	1	0	0				0	0	1	1	0	0	1	0	0				0
0	0	0	0	0	1	0	1				0	0	0	1	0	0	1	0	1				0	0	1	0	0	0	1	0	1				0	0	1	1	0	0	1	0	1				0
0	0	0	0	0	1	1	0				0	0	0	1	0	0	1	1	0				0	0	1	0	0	0	1	1	0				0	0	1	1	0	0	1	1	0				0
0	0	0	0	0	1	1	1				0	0	0	1	0	0	1	1	1				0	0	1	0	0	0	1	1	1				0	0	1	1	0	0	1	1	1				0
0	0	0	0	1	0	0	0				0	0	0	1	0	1	0	0	0				0	0	1	0	0	1	0	0	0				0	0	1	1	0	1	0	0	0				0
0	0	0	0	1	0	0	1				0	0	0	1	0	1	0	0	1				0	0	1	0	0	1	0	0	1				0	0	1	1	0	1	0	0	1				0
0	0	0	0	1	0	1	0				0	0	0	1	0	1	0	1	0				0	0	1	0	0	1	0	1	0				0	0	1	1	0	1	0	1	0				0
0	0	0	0	1	0	1	1				0	0	0	1	0	1	0	1	1				0	0	1	0	0	1	0	1	1				0	0	1	1	0	1	0	1	1				0
0	0	0	0	1	1	0	0				0	0	0	1	0	1	1	0	0				0	0	1	0	0	1	1	0	0				0	0	1	1	0	1	1	0	0				0
0	0	0	0	1	1	0	1				0	0	0	1	0	1	1	0	1				0	0	1	0	0	1	1	0	1				0	0	1	1	0	1	1	0	1				0
0	0	0	0	1	1	1	0				0	0	0	1	0	1	1	1	0				0	0	1	0	0	1	1	1	0				0	0	1	1	0	1	1	1	0				0
0	0	0	0	1	1	1	1				0	0	0	1	0	1	1	1	1				0	0	1	0	0	1	1	1	1				0	0	1	1	0	1	1	1	1				0
0	0	0	1	0	0	0	0				1	0	0	1	1	0	0	0	0				1	0	1	0	1	0	0	0	0				1	0	1	1	1	0	0	0	0				1
0	0	0	1	0	0	0	1				1	0	0	1	1	0	0	0	1				1	0	1	0	1	0	0	0	1				1	0	1	1	1	0	0	0	1				1
0	0	0	1	0	0	1	0				1	0	0	1	1	0	0	1	0				1	0	1	0	1	0	0	1	0				1	0	1	1	1	0	0	1	0				1
0	0	0	1	0	0	1	1				1	0	0	1	1	0	0	1	1				1	0	1	0	1	0	0	1	1				1	0	1	1	1	0	0	1	1				1
0	0	0	1	0	1	0	0				1	0	0	1	1	0	1	0	0				1	0	1	0	1	0	1	0	0				1	0	1	1	1	0	1	0	0				1
0	0	0	1	0	1	0	1				1	0	0	1	1	0	1	0	1				1	0	1	0	1	0	1	0	1				1	0	1	1	1	0	1	0	1				1
0	0	0	1	0	1	1	0				1	0	0	1	1	0	1	1	0				1	0	1	0	1	0	1	1	0				1	0	1	1	1	0	1	1	0				1
0	0	0	1	0	1	1	1				1	0	0	1	1	0	1	1	1				1	0	1	0	1	0	1	1	1				1	0	1	1	1	0	1	1	1				1
0	0	0	1	1	0	0	0				1	0	0	1	1	1	0	0	0				1	0	1	0	1	1	0	0	0				1	0	1	1	1	1	0	0	0				1
0	0	0	1	1	0	0	1				1	0	0	1	1	1	0	0	1				1	0	1	0	1	1	0	0	1				1	0	1	1	1	1	0	0	1				1
0	0	0	1	1	0	1	0				1	0	0	1	1	1	0	1	0				1	0	1	0	1	1	0	1	0				1	0	1	1	1	1	0	1	0				1
0	0	0	1	1	0	1	1				1	0	0	1	1	1	0	1	1				1	0	1	0	1	1	0	1	1				1	0	1	1	1	1	0	1	1				1
0	0	0	1	1	1	0	0				1	0	0	1	1	1	1	0	0				1	0	1	0	1	1	1	0	0				1	0	1	1	1	1	1	0	0				1
0	0	0	1	1	1	0	1				1	0	0	1	1	1	1	0	1				1	0	1	0	1	1	1	0	1				1	0	1	1	1	1	1	0	1				1
0	0	0	1	1	1	1	0				1	0	0	1	1	1	1	1	0				1	0	1	0	1	1	1	1	0				1	0	1	1	1	1	1	1	0				1
0	0	0	1	1	1	1	1	1			1	0	0	1	1	1	1	1	1	1			1	0	1	0	1	1	1	1	1	1			1	0	1	1	1	1	1	1	1	1			1

SH. 1.

Figure 2-3. Head PROM Configuration Chart (Sheet 1)

VOLUME 1 DRIVE Ø												DRIVE 1												DRIVE 2												74S287 DRIVE 3								22C			
VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL II	INHRST H	HX	VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX	VOL	USB	USA	HØ	H1	H2	H3	H4	LAST II	ILL H	INHRST H	HX	VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX
H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1
15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12
1	0	0	0	0	0	0	0				1	1	0	1	0	0	0	0	0				1	1	1	0	0	0	0	0	0				1	1	1	1	0	0	0	0	0				1
1	0	0	0	0	0	0	1				1	1	0	1	0	0	0	0	1				1	1	1	0	0	0	0	0	1				1	1	1	1	0	0	0	0	1				1
1	0	0	0	0	0	1	0				1	1	0	1	0	0	0	1	0				1	1	1	0	0	0	0	1	0				1	1	1	1	0	0	0	1	0				1
1	0	0	0	0	0	1	1				1	1	0	1	0	0	0	1	1				1	1	1	0	0	0	0	1	1				1	1	1	1	0	0	0	1	1				1
1	0	0	0	0	1	0	0				1	1	0	1	0	0	1	0	0				1	1	1	0	0	0	1	0	0				1	1	1	1	0	0	1	0	0				1
1	0	0	0	0	1	0	1				1	1	0	1	0	0	1	0	1				1	1	1	0	0	0	1	0	1				1	1	1	1	0	0	1	0	1				1
1	0	0	0	0	1	1	0				1	1	0	1	0	0	1	1	0				1	1	1	0	0	0	1	1	0				1	1	1	1	0	0	1	1	0				1
1	0	0	0	0	1	1	1				1	1	0	1	0	0	1	1	1				1	1	1	0	0	0	1	1	1				1	1	1	1	0	0	1	1	1				1
1	0	0	0	1	0	0	0				1	1	0	1	0	1	0	0	0				1	1	1	0	0	1	0	0	0				1	1	1	1	0	1	0	0	0				1
1	0	0	0	1	0	0	1				1	1	0	1	0	1	0	0	1				1	1	1	0	0	1	0	0	1				1	1	1	1	0	1	0	0	1				1
1	0	0	0	1	0	1	0				1	1	0	1	0	1	0	1	0				1	1	1	0	0	1	0	1	0				1	1	1	1	0	1	0	1	0				1
1	0	0	0	1	0	1	1				1	1	0	1	0	1	0	1	1				1	1	1	0	0	1	0	1	1				1	1	1	1	0	1	0	1	1				1
1	0	0	0	1	1	0	0				1	1	0	1	0	1	1	0	0				1	1	1	0	0	1	1	0	0				1	1	1	1	0	1	1	0	0				1
1	0	0	0	1	1	0	1				1	1	0	1	0	1	1	0	1				1	1	1	0	0	1	1	0	1				1	1	1	1	0	1	1	0	1				1
1	0	0	0	1	1	1	0				1	1	0	1	0	1	1	1	0				1	1	1	0	0	1	1	1	0				1	1	1	1	0	1	1	1	0				1
1	0	0	0	1	1	1	1				1	1	0	1	0	1	1	1	1				1	1	1	0	0	1	1	1	1				1	1	1	1	0	1	1	1	1				1
1	0	0	1	0	0	0	0	1	1	0	1	1	0	1	1	0	0	0	0	1	1	0	1	1	1	0	1	0	0	0	0	1	1	0	1	1	1	1	1	0	0	0	0	1	1	0	1
1	0	0	1	0	0	0	1	1	1	0	1	1	0	1	1	0	0	0	1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1
1	0	0	1	0	0	1	0	1	1	0	1	1	0	1	1	0	0	1	0	1	1	0	1	1	1	0	1	0	0	1	0	1	1	0	1	1	1	1	1	0	0	1	0	1	1	0	1
1	0	0	1	0	0	1	1	1	1	0	1	1	0	1	1	0	0	1	1	1	1	0	1	1	1	0	1	0	0	1	1	1	1	0	1	1	1	1	1	0	0	1	1	1	1	0	1
1	0	0	1	0	1	0	0	1	1	0	1	1	0	1	1	0	1	0	0	1	1	0	1	1	1	0	1	0	1	0	0	1	1	0	1	1	1	1	1	0	1	0	0	1	1	0	1
1	0	0	1	0	1	0	1	1	1	0	1	1	0	1	1	0	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	1	1	0	1	0	1	1	1	0	1
1	0	0	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	0	1	1	0	1
1	0	0	1	0	1	1	1	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1
1	0	0	1	1	0	0	0	1	1	0	1	1	0	1	1	1	0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	1	1	0	1	1	1	1	1	1	0	0	0	1	1	0	1
1	0	0	1	1	0	0	1	1	1	0	1	1	0	1	1	1	0	0	1	1	1	0	1	1	1	0	1	1	0	0	1	1	1	0	1	1	1	1	1	1	0	0	1	1	1	0	1
1	0	0	1	1	0	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1	1	1	0	1	0	1	1	0	1
1	0	0	1	1	0	1	1	1	1	0	1	1	0	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	0	1
1	0	0	1	1	1	0	0	1	1	0	1	1	0	1	1	1	1	0	0	1	1	0	1	1	1	0	1	1	1	0	0	1	1	0	1	1	1	1	1	1	1	0	0	1	1	0	1
1	0	0	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1
1	0	0	1	1	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	0	1	1	1	0	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1	0	1
1	0	0	1	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1

SH. 2.

Figure 2-3. Head PROM Configuration Chart (Sheet 2)

Input signals for Head PROM addressing are:

Volume #	VOL
Drive #	USB, USA
Head #	H0, H1, H2, H3, H4

Output signals generated by the Head PROM are:

LAST H — 1 indicates the last head (surface) in this volume

ILL H — 1 indicates an illegal head (surface) for this volume

INHRST H — 0 indicates normal organization
1 indicates horizontal organization

Hx — Hx = 0 for Volume 0
Hx = 1 for Volume 1

<u>To program the Head PROM use the following procedure:</u>

1. Determine the type of drives to be used as Drive 0, 1, 2, and 3.

2. Determine the number of logical heads in each drive.

3. Prepare a Head PROM Configuration Chart as shown in Figure 2-3. To fill in the PROM output signal information, use the following guidelines:

   Hx = H0 + VOL

   ILL H = 0 — For valid head number
   = 1 — For illegal head number

   LAST H = 1 — For last head or illegal head numbers

   INHRST H = 0 — For normal operation
   = 1 — For horizontal organization in multi-surface volumes

   (NOTE: If INHRST H = 1, make LAST H = 1 also)

   The above information must be filled in for each volume (sheets 1 and 2 of the Head PROM Configuration Chart.

4. Having properly coded the Head PROM Configuration sheets. The user can program Head PROMs using the chart as program input to the PROM. The PROM I.C. required is a 74S287 and the P.C. board location for installation of the PROM is 22C. installation of the PROM is 22C.

2.5.2 Sector PROM Configuration Chart Coding

The Sector PROM Configuration Chart must be prepared to specify sector information on each disc drive in the system.

The Sector PROM Configuration Chart can be interpreted as follows:

DRIVE # → DRIVE Ø

PROM INPUT SIGNALS

PROM OUTPUT SIGNALS

LOGIC SIGNAL DESIGNATION →

PROM PIN # →

PROM INPUT 8-BIT ADDRESSES (00-1F) NOTE: Addresses 00-7F are possible if all 4 drives used.

ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S	ILL S		
H	G	F	E	D	C	B	A	4	3	2	1
15	1	2	3	4	7	6	5	9	10	11	12
0	0	0	0	0	0	0	0		1		
0	0	0	0	0	0	0	1				
0	0	0	0	0	0	1	0				
0	0	0	0	0	0	1	1				
0	0	0	0	0	1	0	0				
0	0	0	0	0	1	0	1				
0	0	0	0	0	1	1	0				
0	0	0	0	0	1	1	1				
0	0	0	0	1	0	0	0				
0	0	0	0	1	0	0	1				
0	0	0	0	1	0	1	0				
0	0	0	0	1	0	1	1				
0	0	0	0	1	1	0	0				
0	0	0	0	1	1	0	1				
0	0	0	0	1	1	1	0				
0	0	0	0	1	1	1	1				
0	0	0	1	0	0	0	0				
0	0	0	1	0	0	0	1				
0	0	0	1	0	0	1	0				
0	0	0	1	0	0	1	1				
0	0	0	1	0	1	0	0				
0	0	0	1	0	1	0	1				
0	0	0	1	0	1	1	0				
0	0	0	1	0	1	1	1				
0	0	0	1	1	0	0	0				
0	0	0	1	1	0	0	1				
0	0	0	1	1	0	1	0				
0	0	0	1	1	0	1	1				
0	0	0	1	1	1	0	0				
0	0	0	1	1	1	0	1				
0	0	0	1	1	1	1	0				
0	0	0	1	1	1	1	1	1			

A sample of a Sector PROM Configuration Chart format can be found in Figure 2-4.

																																				74S287											19C
DRIVE Ø												DRIVE 1												DRIVE 2												DRIVE 3											
ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S / ILL S				ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S / ILL S				ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S / ILL S				ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S / ILL S			
H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1
15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12
0	0	0	0	0	0	0	0		1			0	0	1	0	0	0	0	0		1			0	1	0	0	0	0	0	0		1			0	1	1	0	0	0	0	0		1		
0	0	0	0	0	0	0	1					0	0	1	0	0	0	0	1					0	1	0	0	0	0	0	1					0	1	1	0	0	0	0	1				
0	0	0	0	0	0	1	0					0	0	1	0	0	0	1	0					0	1	0	0	0	0	1	0					0	1	1	0	0	0	1	0				
0	0	0	0	0	0	1	1					0	0	1	0	0	0	1	1					0	1	0	0	0	0	1	1					0	1	1	0	0	0	1	1				
0	0	0	0	0	1	0	0					0	0	1	0	0	1	0	0					0	1	0	0	0	1	0	0					0	1	1	0	0	1	0	0				
0	0	0	0	0	1	0	1					0	0	1	0	0	1	0	1					0	1	0	0	0	1	0	1					0	1	1	0	0	1	0	1				
0	0	0	0	0	1	1	0					0	0	1	0	0	1	1	0					0	1	0	0	0	1	1	0					0	1	1	0	0	1	1	0				
0	0	0	0	0	1	1	1					0	0	1	0	0	1	1	1					0	1	0	0	0	1	1	1					0	1	1	0	0	1	1	1				
0	0	0	0	1	0	0	0					0	0	1	0	1	0	0	0					0	1	0	0	1	0	0	0					0	1	1	0	1	0	0	0				
0	0	0	0	1	0	0	1					0	0	1	0	1	0	0	1					0	1	0	0	1	0	0	1					0	1	1	0	1	0	0	1				
0	0	0	0	1	0	1	0					0	0	1	0	1	0	1	0					0	1	0	0	1	0	1	0					0	1	1	0	1	0	1	0				
0	0	0	0	1	0	1	1					0	0	1	0	1	0	1	1					0	1	0	0	1	0	1	1					0	1	1	0	1	0	1	1				
0	0	0	0	1	1	0	0					0	0	1	0	1	1	0	0					0	1	0	0	1	1	0	0					0	1	1	0	1	1	0	0				
0	0	0	0	1	1	0	1					0	0	1	0	1	1	0	1					0	1	0	0	1	1	0	1					0	1	1	0	1	1	0	1				
0	0	0	0	1	1	1	0					0	0	1	0	1	1	1	0					0	1	0	0	1	1	1	0					0	1	1	0	1	1	1	0				
0	0	0	0	1	1	1	1					0	0	1	0	1	1	1	1					0	1	0	0	1	1	1	1					0	1	1	0	1	1	1	1				
0	0	0	1	0	0	0	0					0	0	1	1	0	0	0	0					0	1	0	1	0	0	0	0					0	1	1	1	0	0	0	0				
0	0	0	1	0	0	0	1					0	0	1	1	0	0	0	1					0	1	0	1	0	0	0	1					0	1	1	1	0	0	0	1				
0	0	0	1	0	0	1	0					0	0	1	1	0	0	1	0					0	1	0	1	0	0	1	0					0	1	1	1	0	0	1	0				
0	0	0	1	0	0	1	1					0	0	1	1	0	0	1	1					0	1	0	1	0	0	1	1					0	1	1	1	0	0	1	1				
0	0	0	1	0	1	0	0					0	0	1	1	0	1	0	0					0	1	0	1	0	1	0	0					0	1	1	1	0	1	0	0				
0	0	0	1	0	1	0	1					0	0	1	1	0	1	0	1					0	1	0	1	0	1	0	1					0	1	1	1	0	1	0	1				
0	0	0	1	0	1	1	0					0	0	1	1	0	1	1	0					0	1	0	1	0	1	1	0					0	1	1	1	0	1	1	0				
0	0	0	1	0	1	1	1					0	0	1	1	0	1	1	1					0	1	0	1	0	1	1	1					0	1	1	1	0	1	1	1				
0	0	0	1	1	0	0	0					0	0	1	1	1	0	0	0					0	1	0	1	1	0	0	0					0	1	1	1	1	0	0	0				
0	0	0	1	1	0	0	1					0	0	1	1	1	0	0	1					0	1	0	1	1	0	0	1					0	1	1	1	1	0	0	1				
0	0	0	1	1	0	1	0					0	0	1	1	1	0	1	0					0	1	0	1	1	0	1	0					0	1	1	1	1	0	1	0				
0	0	0	1	1	0	1	1					0	0	1	1	1	0	1	1					0	1	0	1	1	0	1	1					0	1	1	1	1	0	1	1				
0	0	0	1	1	1	0	0					0	0	1	1	1	1	0	0					0	1	0	1	1	1	0	0					0	1	1	1	1	1	0	0				
0	0	0	1	1	1	0	1					0	0	1	1	1	1	0	1					0	1	0	1	1	1	0	1					0	1	1	1	1	1	0	1				
0	0	0	1	1	1	1	0					0	0	1	1	1	1	1	0					0	1	0	1	1	1	1	0					0	1	1	1	1	1	1	0				
0	0	0	1	1	1	1	1	1				0	0	1	1	1	1	1	1	1				0	1	0	1	1	1	1	1	1				0	1	1	1	1	1	1	1	1			

SH. 1.

Figure 2-4. Sector PROM Configuration Chart (Sheet 1)

DRIVE Ø												DRIVE 1												DRIVE 2												74S287 DRIVE 3											19C
ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S	ILL S			ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S	ILL S			ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S	ILL S			ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S	ILL S		
H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1
15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12
1	0	0	0	0	0	0	0	1	0			1	0	1	0	0	0	0	0	1	0			1	1	0	0	0	0	0	0	1	0			1	1	1	0	0	0	0	0	1	0		
1	0	0	0	0	0	0	1	1	0			1	0	1	0	0	0	0	1	1	0			1	1	0	0	0	0	0	1	1	0			1	1	1	0	0	0	0	1	1	0		
1	0	0	0	0	0	1	0	1	0			1	0	1	0	0	0	1	0	1	0			1	1	0	0	0	0	1	0	1	0			1	1	1	0	0	0	1	0	1	0		
1	0	0	0	0	0	1	1	1	0			1	0	1	0	0	0	1	1	1	0			1	1	0	0	0	0	1	1	1	0			1	1	1	0	0	0	1	1	1	0		
1	0	0	0	0	1	0	0	1	0			1	0	1	0	0	1	0	0	1	0			1	1	0	0	0	1	0	0	1	0			1	1	1	0	0	1	0	0	1	0		
1	0	0	0	0	1	0	1	1	0			1	0	1	0	0	1	0	1	1	0			1	1	0	0	0	1	0	1	1	0			1	1	1	0	0	1	0	1	1	0		
1	0	0	0	0	1	1	0	1	0			1	0	1	0	0	1	1	0	1	0			1	1	0	0	0	1	1	0	1	0			1	1	1	0	0	1	1	0	1	0		
1	0	0	0	0	1	1	1	1	0			1	0	1	0	0	1	1	1	1	0			1	1	0	0	0	1	1	1	1	0			1	1	1	0	0	1	1	1	1	0		
1	0	0	0	1	0	0	0	1	0			1	0	1	0	1	0	0	0	1	0			1	1	0	0	1	0	0	0	1	0			1	1	1	0	1	0	0	0	1	0		
1	0	0	0	1	0	0	1	1	0			1	0	1	0	1	0	0	1	1	0			1	1	0	0	1	0	0	1	1	0			1	1	1	0	1	0	0	1	1	0		
1	0	0	0	1	0	1	0	1	0			1	0	1	0	1	0	1	0	1	0			1	1	0	0	1	0	1	0	1	0			1	1	1	0	1	0	1	0	1	0		
1	0	0	0	1	0	1	1	1	0			1	0	1	0	1	0	1	1	1	0			1	1	0	0	1	0	1	1	1	0			1	1	1	0	1	0	1	1	1	0		
1	0	0	0	1	1	0	0	1	0			1	0	1	0	1	1	0	0	1	0			1	1	0	0	1	1	0	0	1	0			1	1	1	0	1	1	0	0	1	0		
1	0	0	0	1	1	0	1	1	0			1	0	1	0	1	1	0	1	1	0			1	1	0	0	1	1	0	1	1	0			1	1	1	0	1	1	0	1	1	0		
1	0	0	0	1	1	1	0	1	0			1	0	1	0	1	1	1	0	1	0			1	1	0	0	1	1	1	0	1	0			1	1	1	0	1	1	1	0	1	0		
1	0	0	0	1	1	1	1	1	0			1	0	1	0	1	1	1	1	1	0			1	1	0	0	1	1	1	1	1	0			1	1	1	0	1	1	1	1	1	0		
1	0	0	1	0	0	0	0	1	0			1	0	1	1	0	0	0	0	1	0			1	1	0	1	0	0	0	0	1	0			1	1	1	1	0	0	0	0	1	0		
1	0	0	1	0	0	0	1	1	0			1	0	1	1	0	0	0	1	1	0			1	1	0	1	0	0	0	1	1	0			1	1	1	1	0	0	0	1	1	0		
1	0	0	1	0	0	1	0	1	0			1	0	1	1	0	0	1	0	1	0			1	1	0	1	0	0	1	0	1	0			1	1	1	1	0	0	1	0	1	0		
1	0	0	1	0	0	1	1	1	0			1	0	1	1	0	0	1	1	1	0			1	1	0	1	0	0	1	1	1	0			1	1	1	1	0	0	1	1	1	0		
1	0	0	1	0	1	0	0	1	0			1	0	1	1	0	1	0	0	1	0			1	1	0	1	0	1	0	0	1	0			1	1	1	1	0	1	0	0	1	0		
1	0	0	1	0	1	0	1	1	0			1	0	1	1	0	1	0	1	1	0			1	1	0	1	0	1	0	1	1	0			1	1	1	1	0	1	0	1	1	0		
1	0	0	1	0	1	1	0	1	0			1	0	1	1	0	1	1	0	1	0			1	1	0	1	0	1	1	0	1	0			1	1	1	1	0	1	1	0	1	0		
1	0	0	1	0	1	1	1	1	0			1	0	1	1	0	1	1	1	1	0			1	1	0	1	0	1	1	1	1	0			1	1	1	1	0	1	1	1	1	0		
1	0	0	1	1	0	0	0	1	0			1	0	1	1	1	0	0	0	1	0			1	1	0	1	1	0	0	0	1	0			1	1	1	1	1	0	0	0	1	0		
1	0	0	1	1	0	0	1	1	0			1	0	1	1	1	0	0	1	1	0			1	1	0	1	1	0	0	1	1	0			1	1	1	1	1	0	0	1	1	0		
1	0	0	1	1	0	1	0	1	0			1	0	1	1	1	0	1	0	1	0			1	1	0	1	1	0	1	0	1	0			1	1	1	1	1	0	1	0	1	0		
1	0	0	1	1	0	1	1	1	0			1	0	1	1	1	0	1	1	1	0			1	1	0	1	1	0	1	1	1	0			1	1	1	1	1	0	1	1	1	0		
1	0	0	1	1	1	0	0	1	0			1	0	1	1	1	1	0	0	1	0			1	1	0	1	1	1	0	0	1	0			1	1	1	1	1	1	0	0	1	0		
1	0	0	1	1	1	0	1	1	0			1	0	1	1	1	1	0	1	1	0			1	1	0	1	1	1	0	1	1	0			1	1	1	1	1	1	0	1	1	0		
1	0	0	1	1	1	1	0	1	0			1	0	1	1	1	1	1	0	1	0			1	1	0	1	1	1	1	0	1	0			1	1	1	1	1	1	1	0	1	0		
1	0	0	1	1	1	1	1	1	0			1	0	1	1	1	1	1	1	1	0			1	1	0	1	1	1	1	1	1	0			1	1	1	1	1	1	1	1	1	0		

SH. 2.

Figure 2-4. Sector PROM Configuration Chart (Sheet 2)

Input signals for Sector PROM addressing are:

Illegal Head	ILL H
Drive #	USB, USA
Sector #	S0, S1, S2, S3, S4

Output signals generated by the Sector PROM are:

LAST S 1 indicates the last sector on any track

ILL S 1 indicates that the sector # exceeds the capacity of the drive, or indicates that ILL H is set to 1

To program the Sector PROM use the following procedure:

1. Determine the type of drives to be used as Drive 0, 1, 2, and 3.

2. Determine the number of sectors per track for each drive.

3. Prepare a Sector PROM Configuration Chart as shown in Figure 2-4. To fill in the PROM output signal information, use the following guidelines:

   LAST S = 1 For last sector or ILL H = 1

   ILL S = 0 For illegal sector number or for ILL H = 1

4. Verify that sector count in each drive matches that coded in the Sector PROM Configuration Chart.

5. Having properly coded the Sector PROM Configuration sheets, the user can program Sector PROMs using the charts as program input to the PROM. The PROM I.C. required is a 74S287 and the P.C. board location for installation of the PROM is 19C.

Section 3
PROGRAMMING

3.1 INTRODUCTION

This section outlines the programming protocols for driving up to four storage module-type drives via POINT 4 Data's Lotus 700 Disc Controller. Covered are: instructions for programming data channel transfers, disc controller commands, error conditions and flags, disc formatting and error correction procedures.

3.2 INSTRUCTIONS

Input/Output Instructions enable the processor to communicate with peripheral devices in the computer system. I/O instructions transfer data between accumulators and devices or device controllers, start or reset device operation, and check the status of each device. These instructions also transfer information on the starting block in computer memory, the block length, the disc cylinder/surface/starting sector address and the function to be performed. Each I/O instruction contains a 6-bit device code field that specifies the particular device for this data transfer.

All I/O instruction words have the following format:

0 1 2	3 4	5 6 7	8 9	10 11 12 13 14 15
0 1 1	AC	OP CODE	CTRL	DEVICE CODE

An I/O instruction is designated by 011 in bits 0-2. The OPCODE and Control (CTRL) fields define the I/O operation to be performed. When a register data transfer is involved, the AC field (bits 3-4) specifies the accumulator involved in the data transfer. Bits 10-15 select the device that is to respond to the instruction.

In order to program register data transfers, the programmer must properly code the processor instruction and also load the specified processor accumulator (if required) with the information to be transferred to the controller. Data-out (to the controller) transfers require both the instruction coding and loading of an accumulator with information for the disc controller. Data-in (to the processor) transfers require only instruction coding since information will be read from the disc controller into the specified accumulator.

Four data-out instructions supply the controller with all information necessary to perform any operation required. To obtain status information from the controller, seven data-in instructions are provided.

In the description that follows, coding conventions are used so that the assembler can recognize and translate the instruction into machine language. Instructions are coded according to the following format.

MNEMONIC [optional mnemonics] **OPERAND STRING**

The mnemonic shown in BOLD must be coded exactly as shown in the instruction description. For example, the mnemonic for the "data-out to buffer A" instruction is:

DOA

Operands printed in BOLD are also required to be coded exactly as shown.

Some instructions have optional mnemonics that are appended to the main mnemonic if the option is desired. These mnemonics are enclosed in [ ]. Optional mnemonics may require substitution of a specific control character in order to be properly decoded. Instructions for controlling data transfer between the processor and the disc subsystem use two optional fields:

ac = Accumulator
f = Control Function Flag

The accumulator field (ac) specifies the accumulator number from which information is to be gained or into which information will be loaded during the data transfer.

The control function field controls the disc controller's BUSY and DONE flags as follows:

f = S (start)	Sets BUSY flag to one (1); clears DONE flag to zero (0); clears all controller error flags; starts R/W timeout. Disables drive attention interrupts; starts the following operations as specified in the command: read, read offset, format, write header, verify, read buffer, write, read header.

f = C
(Clear)

Clears BUSY flag to zero (0); clears DONE flag to zero (0); terminates any data transfer operation; clears all controller error flags; clears all drive attention flags. It does not terminate seek or recalibrate commands.

f = P
(Pulse)

Sets the control-full flag to one (1) and starts the following operations as specified in the command: recalibrate, seek release, trespass.

A special instruction is used to initialize and clear all flags in all I/O devices. This instruction, IORST, performs the following functions in the disc controller:

> Performs all operations listed above at f = C; initiates a recalibrate operation on the lowest numbered ready/non-reserved drive. Sets the surface/sector address register to zero, sets the command register to zero, sets the memory address register to zero.

3.2.1 Specify Command and Drive Instruction

Instruction Mnemonic: **DOA[f] ac,DSKP**

Instruction Function: Clears the DONE/DRIVE ATTENTION flags, as selected by bits 0-4 of the specified AC, to zero (0). Loads bits 5-8 of the AC into the controller's command register. Loads bits 9 and 10 of the AC into the controller's drive select register. For CMD drives, loads bit 11 into the controller's volume select register. Loads bits 12-15 into the auxiliary register.

NOTE

This instruction is ignored if the control-full bit is set to one (1).

Accumulator Format:

SPECIFY COMMAND AND DRIVE

DOA(f) ac, DSKP

CLR R/W DN	CLR DRIVE ATTN.				COMMAND				DRIVE#		VOL	XMA			
	0	1	2	3								MSB			
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Chart of Accumulator Bit Functions:

Bits	Functions
0	Clears R/W DONE flag.
1-4	Clears the respective Seek DONE/ATTENTION flags for drives 0-3
5-8	Command: 0000 Read
	0001 Recalibrate
	0010 Seek
	0011 Write header
	0100 Read offset +
	0101 Read offset -
	0110 Format
	0111 Release
	1000 Trespass
	1001 Alternate Mode 1
	1010 Alternate Mode 2
	1011 No operation
	1100 Verify
	1101 Read FIFO buffer
	1110 Write
	1111 Read format
9-10	Selects drive. Also selects drive status register.
11	Volume select bit. Defines non-removeable surface(s) in CMD drives.
12-15	Extended memory address (optional).

3.2.1.1 Detailed Description of Commands

Accumulator bits 5-8 transferred by the DOA instruction call for a command to be given to the controller. The 16 possible commands cause the controller to interact with the specified drive and perform the function indicated in the command. The chart below gives detailed descriptions of the controller and disc drive operations which result from each of the 16 commands.

Command Name	Bit Config.	Resulting Operations
READ	0000	Reads data up to 32 consecutive sectors from the selected drive starting at the specified surface and sector on the cylinder selected by the previous seek command. Data is read from up to 32 consecutive sectors into computer memory starting at the specified memory address. Surface or cylinder boundaries are crossed if necessary. The controller finds and verifies the header of the desired sector(s); reads the data and transmits it via the DMA channel to computer memory. During sector read the controller accumulates an ECC checkword. The ECC checkword is compared with the checkword recorded in the ECC words recorded when the data was written to disc. If an ECC error is detected, the same sector is reread and retransmitted. If an ECC error occurs again, the operation is terminated.
RECALIBRATE	0001	Moves the heads of the selected disc drive to cylinder zero (0) and attempts to clear drive faults (if any).
SEEK	0010	Moves the heads of the selected disc drive to the specified cylinder. In cartridge module drives, switches to the desired volume.

Command Name	Bit Config.	Resulting Operations
WRITE HEADER	0011	Writes the three header words, transferred from the specified computer memory address, into up to 32 consecutive sectors in the selected drive. Data is written from memory starting at the specified memory address onto the disc starting at the specified surface and sector on the cylinder selected by the previous seek command. Surface or cylinder boundaries are crossed if necessary. The controller finds the specified sector and writes the preamble, sink bit, header, header CRC, and gap into the sector using the three words from memory. Any data previously written in this sector remains intact.
READ OFFSET +	0100	Same as a Read command except with the disc drive heads offset towards the spindle. This command is used for data recovery. Offset is only for the duration of this command.
READ OFFSET -	0101	Same as a Read command except with the disc drive heads offset away from the spindle. This command is used for data recovery. Offset is only for the duration of this command.
FORMAT	0110	Formats up to 32 consecutive sectors in the selected disc drive starting at the specified surface and sector on the cylinder selected by the previous seek command. Surface or cylinder boundaries are crossed if necessary. The controller finds the specified sector and writes the preamble, sink bit, header, header CRC, gap, data splice/preamble, sink bit, all zero data, ECC and postamble into the sector using address information from the controller. Any data previously written in this sector is lost.

Command Name	Bit Config.	Resulting Operations
RELEASE DRIVE	0111	Clears the reserved status of the drive selected in the DOA instruction. Clearing reserved status applies only to this status in relation to the processor issuing a DOA instruction (Dual Processor System).
TRESPASS	1000	Clears the reserved status of the drive selected in the DOA instruction, as reserved by the other processor in the system. It then reserves the selected drive for the processor issuing the DOA instruction (Dual Processor System).
ALTERNATE MODE 1	1001	In Alternate Mode 1, the DIA instruction reads the controller's memory address register and the DIB instruction reads the controller's extended memory address register.
ALTERNATE MODE 2	1010	In Alternate Mode 2, the DIA and DIB instructions read the ECC remainder words.
NO OPERATION	1011	None.
DATA VERIFY	1100	Reads up to 32 consecutive sectors from the selected drive, starting at the specified surface and sector on the cylinder selected by the previous seek command. Data read from the disc drive is compared word-by-word to data read from computer memory starting at the specified memory address. Surface or cylinder boundaries are crossed if necessary. The controller finds and verifies the header of the specified sector(s); reads the data and compares it to data read from computer memory. During sector read the controller accumulates an ECC checkword. This ECC checkword is compared with the recorded checkword. If an ECC or data compare error is detected, the operation terminates without retry.

Command Name	Bit Config.	Resulting Operations
READ FIFO	1101	This is a diagnostic test command. Eighteen words of data will be read into memory starting at the specified memory address. In order to read from the controller's FIFO into computer memory, the FIFO buffer must first be filled by issuing a write command to a write disabled drive.
WRITE	1110	Writes up to 32 consecutive sectors from memory starting at the specified memory address onto the selected drive. Writing starts at the specified surface and sector on the cylinder specified in the previous seek command. Surface or cylinder boundaries are crossed if necessary. The controller finds and verifies the header of the specified sector and writes 256 words of data from memory onto the specified sector. During write the controller accumulates a 32-bit ECC and appends it to the data field.
READ FORMAT	1111	The controller finds the desired sector and reads the three-word sector header, header CRC and two-word ECC from up to 32 consecutive sectors on the selected drive, starting at the specified surface and sector on the cylinder selected by the previous seek command. The data read is transferred into computer memory starting at the specified memory address. Surface or cylinder boundaries are crossed if necessary.

3.2.2 Specify Memory Address Instruction

Instruction Mnemonic: **DOB[f] ac,DSKP**

Instruction Function: Loads bits 0-15 of the specified AC into the controller's memory address register. Transfers the four XMA bits from the auxiliary register of the controller into the optional extended memory address register.

NOTE
This instruction is ignored if the BUSY flag is set to one (1).

Accumulator Format:

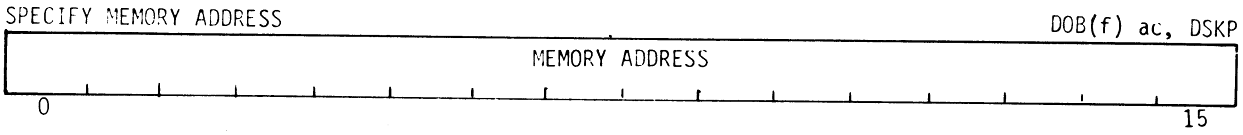


Chart of Accumulator Bit Functions:

Bits	Functions
0-15	Memory address of next data channel transfer.

3.2.3 Specify Cylinder Instruction

Instruction Mnemonic: **DOC**[f] ac,**DSKP**

(If the previous DOA specified a seek command)

Instruction Function: Loads bits 6-15 into the cylinder bits of the controller's cylinder register.

NOTE
This instruction is ignored if the control-full bit is set to one (1).

Accumulator Format:

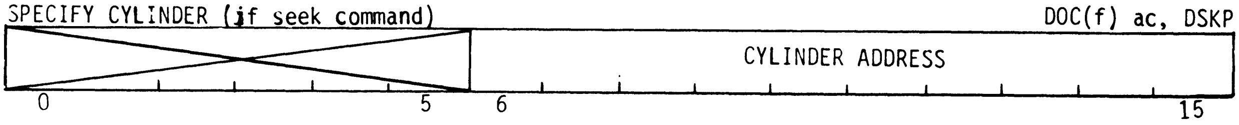


Chart of Accumulator Bit Functions:

Bits	Functions
0-5	Unused
6-15	Cylinder address for seek operation

3.2.4 Specify Surface, Sector and Sector Count Instruction

Instruction Mnemonic: **DOC**[f] ac,**DSKP**
(If the previous DOA specified other than a seek command)

Instruction Function: Loads bits 1-5 of the specified AC into the controller's surface address register. Loads bits 6-10 of the specified AC into the controller's sector address register. Loads bits 11-15 of the specified AC into the controller's sector count register.

NOTE
This instruction is ignored if the BUSY flag is set to one (1).

Accumulator Format:

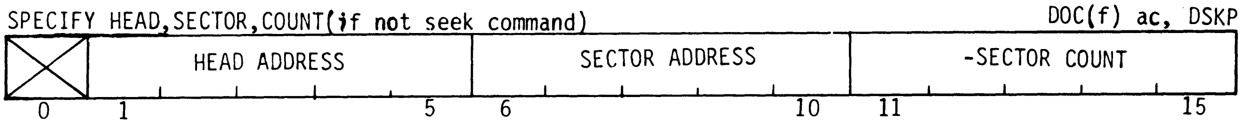


Chart of Accumulator Bit Functions:

Bits	Function
0	Unused.
1-5	Starting surface address for data transfer operation.
6-10	Starting sector address for data transfer operation.
11-15	Two's complement of the number of sectors to be transferred in one operation.

3.2.5 Read Controller Status Instruction

Instruction Mnemonic: **DIA[f] ac,DSKP**

Instruction Function: Loads the controller's status flags into bits 0-15 of the specified AC.

Accumulator Format:

READ CONTROLLER STATUS DIA(f) ac, DSKP

CONTR FULL	R/W DONE	DRIVE SEEK DONE				☒	ILL SECT	ECC ERR	BAD SECT	CYL A ERR	HD A ERR	VFY ERR	R/W TIME	DATA LATE	R/W ERR
		0	1	2	3										
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Chart of Accumulator Bit Functions:

Bits	Function
0	Control full. Drive command initiated by previous IOPLS has not yet been issued to the selected drive.
1	DONE flag of controller.
2-5	Seek DONE flags of drives. Respective drives have executed a recalibrate or seek command, became ready, or rejected an illegal seek command.
6	Unused.
7	Illegal sector/surface address.
8	ECC error during read or verify.
9	Bad sector flag was detected in desired sector header.
10	Cylinder address error was detected during sector header check.
11	Surface address error was detected during sector header check.
12	Verify error. Data read from disc did not match data read from memory.
13	R/W timeout. The data transfer operation was not completed in 1 second.
14	Data-late error. The FIFO buffer overflowed or underflowed during data transfer.
15	Controller error flag. One or more of bits 7 thru 14 is set to one (1), or a drive fault occurred on the drive selected for the current data transfer operation.

3.2.5.1 Controller Error Conditions

Accumulator bits 7-15 which are transferred to the computer by the DIA instruction carry controller status information to the processor. The accumulator bits and the error conditions flagged by setting of these bits to one (1) are listed below.

Accumulator Bit	Error Flag	Error Condition Description
7	ILLEGAL SECTOR/ SURFACE ADDRESS	The sector or surface address received by the controller for a data transfer operation exceeds the drive capacity set in the controller. Illegal sector/surface address is detected before any data transfer takes place. Operation terminates with the error flag and device-done flag set.
8	ECC ERROR	An ECC error has been detected during read or verify operations. An ECC error will cause one retry (read only). If the ECC error persists, operation terminates at the end of the sector with error flag and device-done flag set. The surface/sector address/sector count points to the next sector to be read.
9	BAD SECTOR	The bad sector flag is set in the header of the faulty sector. This sector cannot store data reliability. Data transfer is terminated promptly. The surface/sector address/sector count points to the sector in which the error occurred.
10	CYLINDER ADDRESS ERROR	The heads are not positioned on the cylinder specified by the previous seek operation. Data transfer operations are terminated promptly. The surface/sector address/sector count points to the sector in which the error occurred. A recalibrate command should be issued to the drive.

Accumulator Bit	Error Flag	Error Condition Description
11	SURFACE ADDRESS ERROR	Surface address in the sector header is different from the surface address currently specified. Data transfer operation is terminated promptly. The surface/sector address/sector count points to the sector in which the error occurred.
12	VERIFY ERROR	The data read from the disc did not match data read from computer memory. The data transfer operation is terminated at the end of the sector with error flags and device-done flags set to one (1). The surface/sector address/count points to the next sector to be verified.
13	READ/WRITE TIMEOUT	The data transfer operation initiated by the last "f = S (start)" was not completed in 1 second. The data transfer operation is terminated promptly. The surface/sector address/sector count points to the sector in which the error occurred.
14	DATA LATE ERROR	The FIFO buffer overflowed or underflowed. The data channel did not support the transfer rate required by the controller. Data transfer operation is terminated promptly. The surface/sector address/sector count points to the sector in which the error occurred.
15	CONTROLLER ERROR FLAG	One or more error flags listed above or a drive error flag (see subsection 3.2.6.1) on the drive selected has been set to one (1).

3.2.6 Read Drive Status Instruction

Instruction Mnemonic: **DIB[f] ac,DSKP**

Instruction Function: Loads the drive status flags of the drive selected by the previous DOA instruction into bits 0-15 of the specified AC.

Accumulator Format:

READ DRIVE STATUS DIB(f) ac, DSKP

☒	DR RES	☒	DR RDY	DR BUSY	☒	WRITE DISA	☒	ILL ADDR	ILL COM	DR FLT	DR FLT	DR FLT	☒		DR ERR
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Chart of Accumulator Bit Functions:

Bits	Function
0	Unused.
1	Drive reserved by other processor.
2	Unused.
3	Drive ready.
4	Drive busy. Drive is executing a position command.
5	Unused.
6	Write disabled.
7	Unused.
8	Illegal address. The cylinder address specified exceeds capacity of the drive.
9	Illegal command. A seek command was issued to a busy drive or a write command was issued to a write protected drive.
10-12	Drive Fault.
13-14	Unused.
14	Drive error flag. One or more of bits 8 thru 12 is set to one (1).

3.2.6.1 Drive Error Conditions

Accumulator bits 8-12 and 15, which are transferred to the computer by the DIB instruction, carry disc drive status information to the processor. The accumulator bits and the error conditions flagged by setting of these bits to a one (1) are listed in the chart below.

Accumulator Bit	Error Flag	Error Condition Description
8	ILLEGAL ADDRESS	The cylinder address received by the drive exceeds the capacity of the drive or a seek command was not completed in 500 milliseconds by the drive. An automatic recalibrate is issued by the controller.
9	ILLEGAL COMMAND	A seek command was issued to a busy drive or a write command was issued to a write protected drive.
10-12	DRIVE FAULT	Any drive fault. A drive fault on the selected drive during data transfer terminates operation, sets to 1 (one) the device-done flag and the controller-error flag. If the controller detects a drive fault, it issues a fault-clear command to that drive. If the controller detects a seek error in the drive it issues a recalibrate command to that drive. It also sets to 1 (one) the drive-DONE flag.
15	DRIVE ERROR FLAG	One or more of the error conditions listed above exist.

3.2.7 Read Memory Address Instruction

Instruction Mnemonic: **DIA[f]** ac,**DSKP** (in Alternate Mode 1)

Instruction Function: Loads the contents of the controller's memory address register into bits 0-15 of the specified AC.

Accumulator Format:

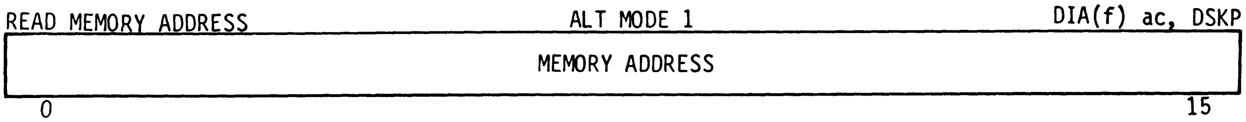


Chart of Accumulator Bit Functions:

Bits	Function
0-15	Memory address of next data channel transfer.

3.2.8 Read Extended Memory Address Instruction

Instruction Mnemonic: **DIB[f] ac,DSKP** (in Alternate Mode 1)

Instruction Function: Loads the contents of the controller's extended memory address register (optional) into bits 12-15 of the specified AC.

Accumulator Format:

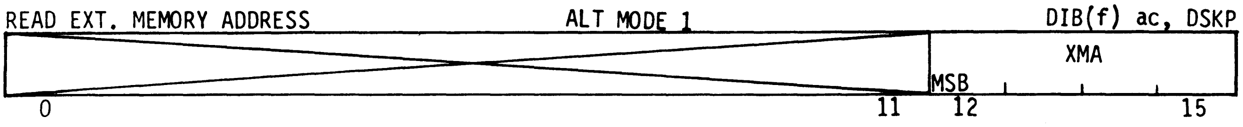


Chart of Accumulator Bit Functions:

Bits	Function
0-11	Unused.
12-15	Extended Memory Address (optional).

3.2.9 Read High-Order ECCR Bits Instruction

Instruction Mnemonic: **DIA**[f] ac,**DSKP** (in Alternate Mode 2)

Instruction Function: Loads the high-order bits of the controller's ECC remainder register into bits 0-15 of the specified AC.

Accumulator Format:

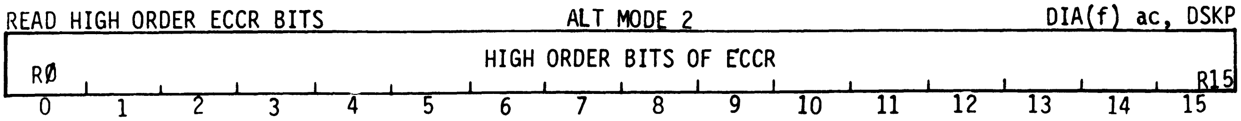


Chart of Accumulator Bit Functions:

Bits	Function
0-15	High-order bits of ECC remainder following a read or verify operation.

3.2.10 Read Low-Order ECCR Bits Instruction

Instruction Mnemonic: **DIB[f] ac,DSKP** (in Alternate Mode 2)

Instruction Function: Loads the low-order bits of the controller's ECC remainder register into bits 0-15 of the specified AC.

Accumulator Format:

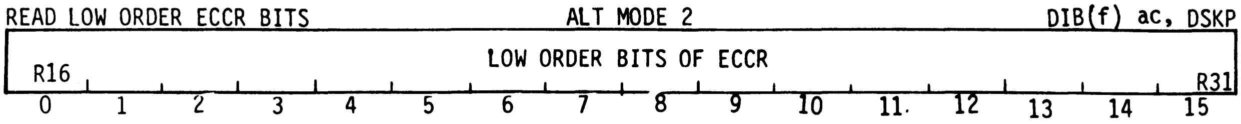


Chart of Accumulator Bit Functions:

Bits	Function
0-15	Low-Order bits of ECC remainder following a read or verify operation.

3.2.11 Read Surface, Sector, Count Instruction

Instruction Mnemonic: **DIC[f] ac,DSKP**

Instruction Function: Loads the contents of the controller's surface address register into bits 1-5 of the specified AC. Loads the contents of the controller's sector-address register into bits 6-10 of the specified AC. Loads the contents of the controller's sector-count register into bits 11-15 of the specified AC.

Accumulator Format:

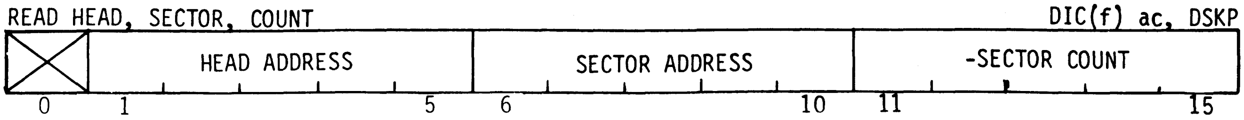


Chart of Accumulator Bit Functions:

Bits	Function
0	Unused.
1-5	Surface address of the sector addressed by bits 6-10.
6-10	Sector address of the sector following the last sector that was transferred.
11-15	Two's complement of the number of sectors remaining to be transferred.

3.3 DISC FORMATTING

Before data transfer operations can take place the surfaces on the disc must be formatted. Formatting records a unique header at the beginning of each sector on the disc. This header contains the information necessary to identify each sector. Figure 3-1 represents the configuration of a formatted sector.

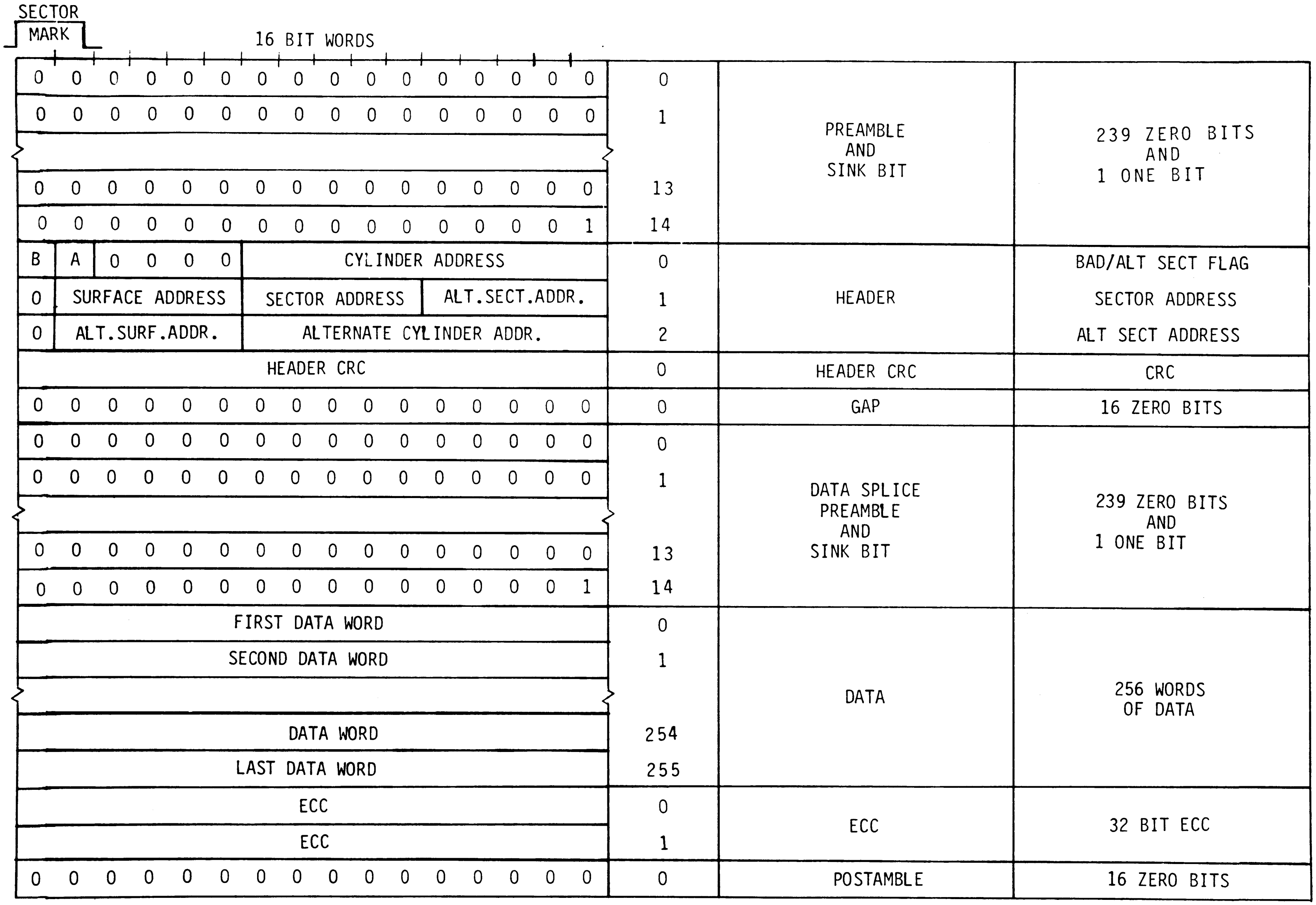
3.3.1 Sector Format

The formatted sector consists of:

- 30 bytes of preamble and sink bit
- 6 bytes of header information
- 2 bytes of header CRC (cyclic redundancy code)
- 2 bytes of gap
- 30 bytes of data splice preamble and sink bit
- 512 bytes of data storage
- 4 bytes of ECC (error correction code)
- 2 bytes of postamble

The 6 bytes of header information contain address information used by the controller to locate the desired sector for a data transfer operation. The cylinder address occupies 10 bits of the first word of the header. The first word of the header also contains a bad sector flag and an alternate sector flag of one bit each. Surface address (5-bit), sector address (5-bit) and alternate sector address (5-bit) occupy the second word of the header. Word three of the header contains a 5-bit alternate surface address and a 10-bit alternate cylinder address. The header is completed by a word of CRC that ensures error-free reading of the header.

If a sector cannot support error-free data, the Format/Surface Analysis Program sets the bad-sector flag or the alternate-sector flag. If the bad-sector flag is set to one (1) operation will be terminated. If the alternate-sector flag is set to one (1), an alternate-sector address is provided.



16 BIT WORDS			
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	PREAMBLE AND SINK BIT	239 ZERO BITS AND 1 ONE BIT
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1		
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	13		
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	14		
B | A | 0 0 0 0 | CYLINDER ADDRESS	0	HEADER	BAD/ALT SECT FLAG
0 | SURFACE ADDRESS | SECTOR ADDRESS | ALT.SECT.ADDR.	1		SECTOR ADDRESS
0 | ALT.SURF.ADDR. | ALTERNATE CYLINDER ADDR.	2		ALT SECT ADDRESS
HEADER CRC	0	HEADER CRC	CRC
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	GAP	16 ZERO BITS
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	DATA SPLICE PREAMBLE AND SINK BIT	239 ZERO BITS AND 1 ONE BIT
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1		
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	13		
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	14		
FIRST DATA WORD	0	DATA	256 WORDS OF DATA
SECOND DATA WORD	1		
DATA WORD	254		
LAST DATA WORD	255		
ECC	0	ECC	32 BIT ECC
ECC	1		
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	POSTAMBLE	16 ZERO BITS

Figure 3-1. Configuration of a Formatted Sector

3.3.2 Format Programming

The format command causes complete reformatting of the sector(s) specified. Header information is provided from the controller's address register. When a sector is formatted all data in that sector is lost (the sector data bytes are filled with 0's).

The write-header command is provided to set flags, write alternate sector address, or provide for interleaving of sectors. The write-header command uses three words from computer memory as a source of header information. This command does not destroy the data in that sector.

For data-transfer commands the controller reads the sector headers passing under the selected disc drive head. It checks flags, address and CRC.

- A CRC error causes a retry on the next header
- A cylinder/surface address error terminates the operation by setting the respective error flags

If the cylinder, surface and sector address match:

- A bad-sector flag terminates the operation
- An alternate-sector flag causes the controller to find the sector designated by the alternate-sector address

If neither flag is set, data transfer commences. See Figure 3-2 for a flowchart of this operation.

For format, write-header, and read-format operations the controller synchronizes to the index mark, and then counts the sector marks to find the desired sector.

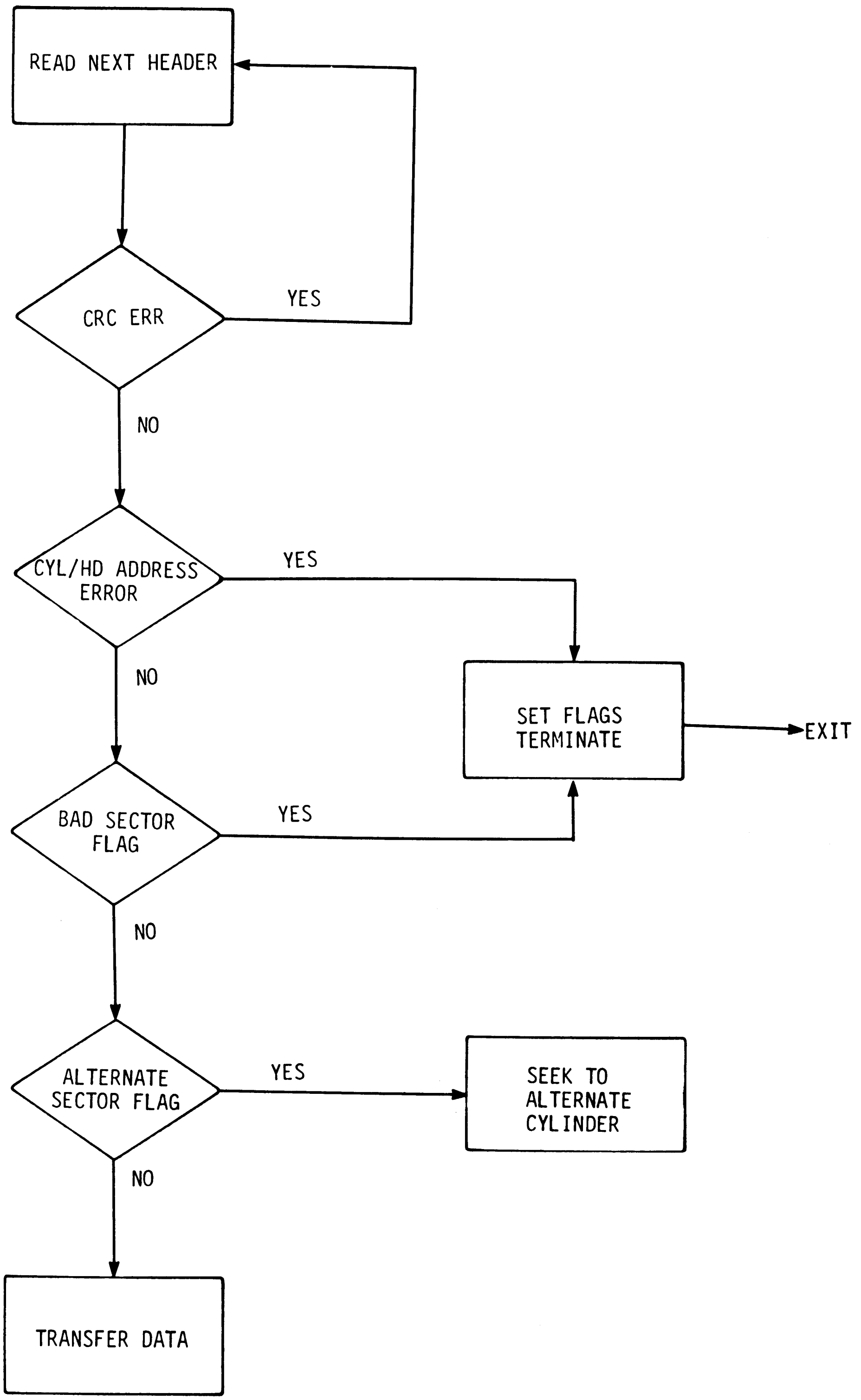


Figure 3-2. Format Programming Flowchart

3.4 DATA TRANSFER PROGRAMMING

Execution of a data transfer between the disc and memory requires use of a series of commands to ensure that all information necessary for the transfer has been conveyed to all devices involved. Activities involved in a data transfer include:

- drive selection
- status checks on that drive and the controller
- drive head positioning
- issuing of a read or write command
- specification of disc surface and sector address
- specification of the memory address for data transfer
- start operation

3.4.1 Drive Command Operation Procedure

Drive Commands are: recalibrate, seek, trespass and release. All may be programmed using the following procedure:

1. Check the controller status by issuing a DIA (Read Controller Status) instruction. In order to proceed, the controller status transferred to the processor accumulator must contain the control-full flag (bit 0) set to zero (0), indicating that the previous drive command has been transmitted to that drive. If the control-full flag is set to one (1) all DOA instructions will be ignored.

2. Select the drive and specify the drive command by issuing a DOA (Specify Command and Drive) instruction. The processor accumulator should contain the drive number in bits 9 and 10. This information will be transferred to the controller's drive select register. Bits 5 through 8 of the accumulator should contain the drive command to be transferred to the controller's command register.

3. Read the disc drive status by issuing a DIB (Read Drive Status) instruction. The drive status information transferred from the controller indicates whether the drive is ready; whether the drive is reserved by the other processor in the system (if any); whether any drive faults exist; and whether the drive is busy with a previous drive command. The drive will reject all commands if it is not ready, is busy, or has been reserved by the other processor.

4. The desired cylinder number is specified by issuing a DOCP (Specify Cylinder) instruction. The P (Pulse) control function sets the control-full flag to one (1) and initiates the command transfer to the drive. Once the command is transferred to the drive, the control-full flag is cleared. Completion of the command sets the drive-done flag to one (1), causing an interrupt to the controller (if the controller-busy flag is not set to 1). If the drive is not ready, is busy or has been reserved for the other processor when the DOCP instruction is issued, the command will be rejected by setting the drive-done flag to one (1) and clearing the control-full flag to zero (0).

NOTE

Trespass and recalibrate commands are issued similarly.

3.4.2 Data Transfer Procedure

Read/write commands can be issued without waiting for the completion of the seek operation. Once the seek operation has been initiated, proceed to program data transfer as follows:

1. Check the device (controller) busy flag by issuing a Skip-if-BUSY-Flag-is-Nonzero-I/O instruction. Verify that no data transfer is in progress (BUSY = 0). No new data transfer may be initiated if the previous data transfer is still in progress.

2. Check controller status by issuing a DIA (Read Controller Status) instruction. In order to proceed, the controller status transferred to the processor accumulator must contain the control-full flag (bit 0) set to zero (0), indicating that the previous drive command has been transmitted to that drive. If the control-full flag is set to one (1), all DOA instructions will be ignored.

3. Select the drive and specify a read/write (or verify) command by issuing a DOA (Specify Command and Drive) instruction. The processor accumulator should contain the drive number in bits 9 and 10. This information will be transferred to the controller's drive-select register. Bits 5 through 8 of the accumulator should contain a read/write or verify command (0000/1110 or 1100) to be transferred to the controller's command register.

4. Read the disc drive status by issuing a DIB (Read Drive Status) instruction. Check the status to verify that the disc drive is ready, that it is not reserved by the other processor (if any), and that there is no drive fault. For a write operation, that drive must not be write-protected.

5. The desired starting surface, sector address and sector count is specified by issuing a DOC (Specify Surface, Sector and Sector Count) instruction. Bits 1-5 of the processor accumulator should contain the surface address to be transferred to the controller's surface-address register. Bits 6-10 of the accumulator should contain the sector address to be transferred to the controller's sector-address register. Bits 11-15 of the accumulator should contain the sector count to be transferred into the controller's sector count register.

6. Issue a DOBS to specify the memory address to be used in the first DMA transfer. The S (Start) control function sets the device-busy flag to one (1), clears to zero (0) the device-done flag, starts a one-second timer and initiates the data transfer operation.

   The controller finds the desired sector on the specified surface and performs the data transfer between memory and the disc drive. Data transfer continues until sector count overflows, at which time transfer operation terminates. Termination of data transfer causes BUSY to clear to zero (0), sets the device-done to one (1), and generates an interrupt to the processor.

3.4.3 Dual Processor Data Transfer Procedure

In a dual processor system it may be desirable to release the disc drive after a data transfer operation. This allows access to the disc drive by the other processor in the system.

To release the disc drive from reserved status, proceed as follows:

1. Check the controller status by issuing a DIA (Read Controller Status) instruction. The controller status transferred to the processor accumulator must contain the control-full flag (bit 0) set to zero (0) in order to proceed.

2. Select the desired drive and specify a release command by issuing a DOAP (Specify Command and Drive) instruction. The P (Pulse) control function sets the control-full flag to one (1) and initiates the release command to that drive. Once the release command is transferred to that drive the control-full flag is cleared to zero (0).

3.4.4 Programming Flowcharts

Figures 3-3 through 3-5 are flowcharts of programming procedures for drive command operations, data transfer operations and interrupt operations. Each procedure is flowcharted in the following figures:

Figure 3-3. Drive Command Operation Programming Flowchart

Figure 3-4. Data Transfer Operation Programming Flowchart

Figure 3-5. Interrupt Processing Programming Flowchart

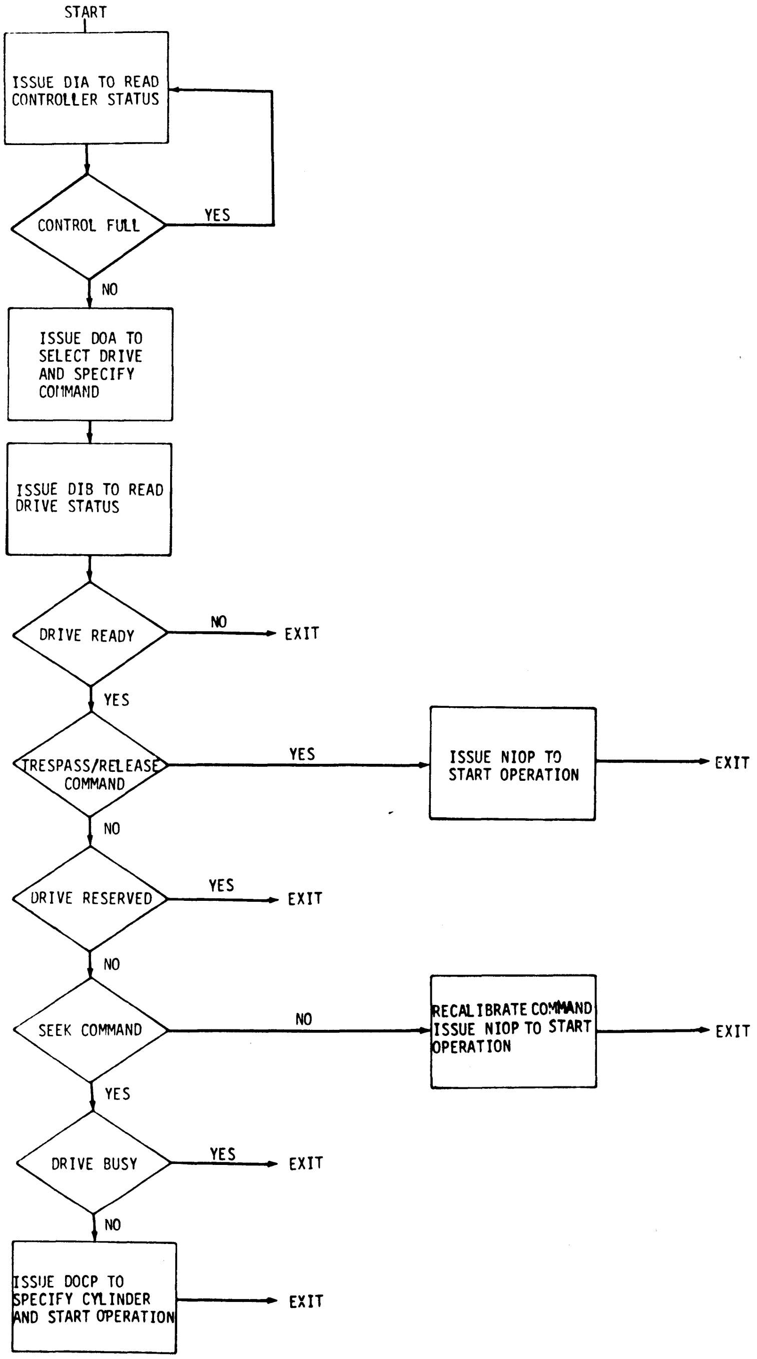

START
ISSUE DIA TO READ CONTROLLER STATUS
CONTROL FULL
YES
NO
ISSUE DOA TO SELECT DRIVE AND SPECIFY COMMAND
ISSUE DIB TO READ DRIVE STATUS
DRIVE READY
NO
EXIT
YES
TRESPASS/RELEASE COMMAND
YES
ISSUE NIOP TO START OPERATION
EXIT
NO
DRIVE RESERVED
YES
EXIT
NO
SEEK COMMAND
NO
RECALIBRATE COMMAND ISSUE NIOP TO START OPERATION
EXIT
YES
DRIVE BUSY
YES
EXIT
NO
ISSUE DOCP TO SPECIFY CYLINDER AND START OPERATION
EXIT

Figure 3-3. Drive Command Operation Programming Flowchart

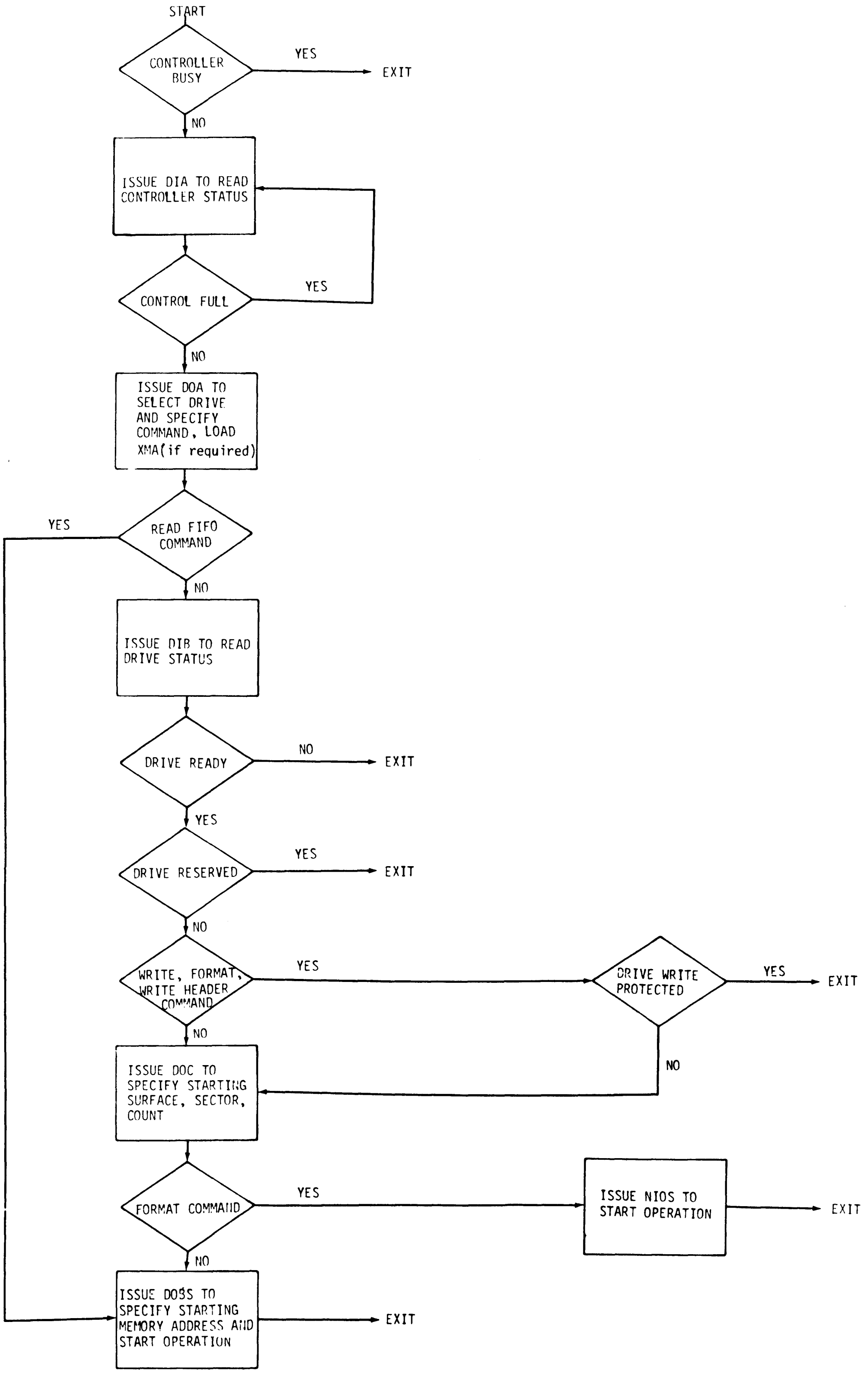

START
CONTROLLER BUSY
YES
EXIT
NO
ISSUE DIA TO READ CONTROLLER STATUS
CONTROL FULL
YES
NO
ISSUE DOA TO SELECT DRIVE AND SPECIFY COMMAND, LOAD XMA(if required)
READ FIFO COMMAND
YES
NO
ISSUE DIB TO READ DRIVE STATUS
DRIVE READY
NO
EXIT
YES
DRIVE RESERVED
YES
EXIT
NO
WRITE, FORMAT, WRITE HEADER COMMAND
YES
DRIVE WRITE PROTECTED
YES
EXIT
NO
NO
ISSUE DOC TO SPECIFY STARTING SURFACE, SECTOR, COUNT
FORMAT COMMAND
YES
ISSUE NIOS TO START OPERATION
EXIT
NO
ISSUE DOBS TO SPECIFY STARTING MEMORY ADDRESS AND START OPERATION
EXIT

Figure 3-4. Data Transfer Operation Programming Flowchart

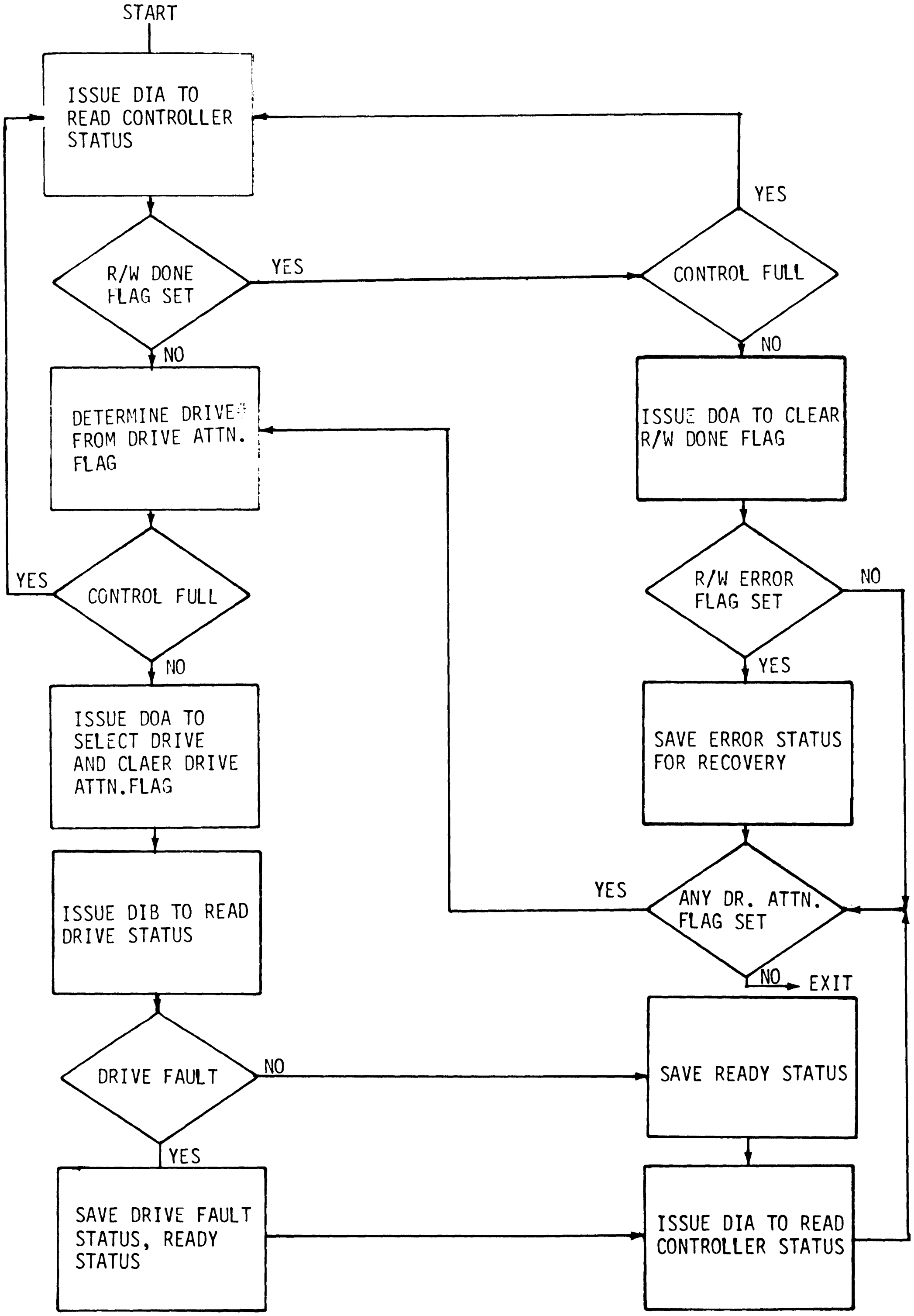

START
ISSUE DIA TO READ CONTROLLER STATUS
R/W DONE FLAG SET
YES
NO
CONTROL FULL
YES
NO
DETERMINE DRIVE# FROM DRIVE ATTN. FLAG
ISSUE DOA TO CLEAR R/W DONE FLAG
CONTROL FULL
YES
NO
R/W ERROR FLAG SET
NO
YES
ISSUE DOA TO SELECT DRIVE AND CLAER DRIVE ATTN.FLAG
SAVE ERROR STATUS FOR RECOVERY
ISSUE DIB TO READ DRIVE STATUS
YES
ANY DR. ATTN. FLAG SET
NO
EXIT
DRIVE FAULT
NO
SAVE READY STATUS
YES
SAVE DRIVE FAULT STATUS, READY STATUS
ISSUE DIA TO READ CONTROLLER STATUS

Figure 3-5. Interrupt Processing Programming Flowchart

3.5 DATA ERROR DETECTION AND CORRECTION

When the controller writes data into a sector, it generates a 4-byte Error Correction Code (ECC) checkword immediately after the last data word (see Figure 3-1, Configuration of a Formatted Sector). When the controller reads data from a sector it generates an ECC remainder from the 512 bytes of data and the four bytes of ECC checkword. If the remainder generated is nonzero, an error has occurred in data transfer. The ECC error flag is set to one (1) in the controller.

The ECC feature detects all errors contained within 21 contiguous bits. It also provides for software correction of errors within 11 contiguous bits. A number of additional errors may also be detected; however, correction is not guaranteed since there is a small possibility that the correction algorithm will produce erroneous correction on data strings containing errors that exceed 11 bits. There is also a very small class of errors that cannot be detected by this ECC-generation routine.

3.5.1 Error Detection Procedure

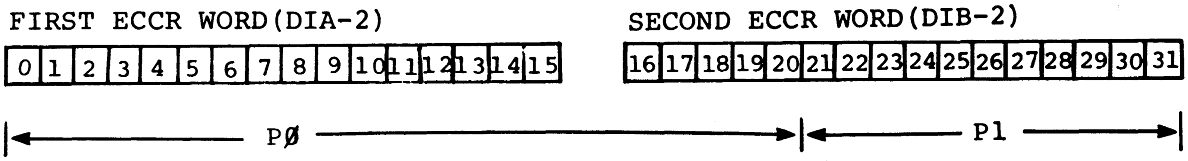
To detect errors using the ECC remainder (ECCR) generated when the data transfer took place, proceed as follows:

A) Use a DIA instruction in Alternate Mode 2 (Read-High-Order-ECCR bits) to read the high-order word of the controller's ECCR Register into bits 0-15 of the specified processor accumulator.

B) Use a DIB instruction in Alternate Mode 2 (Read-Low-Order-ECCR bits) to read the low-order word of the controller's ECCR Register into bits 0-15 of the specified processor acccumulator.

C) To check for a detected data transfer error and for correctability proceed as follows:

1) The 32 bits of the two ECCR words are grouped using the identifiers P0 and P1. P0 consists of the first 21 bits of the two ECCR words and P1 consists of the last 11 bits of the two ECCR words as illustrated below:



2) Test for the following conditions in P0 and P1.

   a) If both P0 and P1 are EQUAL to zero (0) then no error has been detected. Exit the routine.

   b) If P0 is NOT EQUAL to zero (0) and P1 is EQUAL to zero (0) or if P0 is EQUAL to zero (0) and P1 is NOT EQUAL to zero (0) the error is not correctable. Exit the routine.

   c) If both P0 and P1 are NOT EQUAL to zero (0) proceed to step 3 below.

3) Rotate P0 left N times until P0 (bits 0-9) is equal to zero (0). See Figure 3-6 for an illustration of this procedure.

4) If N is greater than or equal to zero and less than or equal to 21 ($0 \leq N \leq 21$) save the value of N and the error pattern. Then proceed to step 5 below. If N is greater than 21 the error is not correctable; exit the routine.

5) Rotate P1 and "exclusive OR" bit 30 with bit 21 M times until P1 (bits 21-31) are equal to the error pattern. See Figure 3-7 for an illustration of this procedure.

6) If M is greater than or equal to zero and less than or equal to 2047 ($0 \leq M \leq 2047$) proceed to the error correction procedure below. If M is greater than 2047 the error is not correctable; exit the routine.

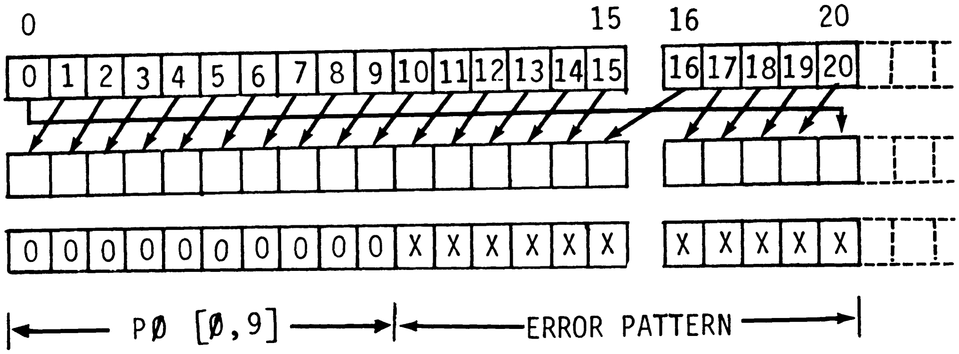


Figure 3-6. Rotation of P0 Left N Times

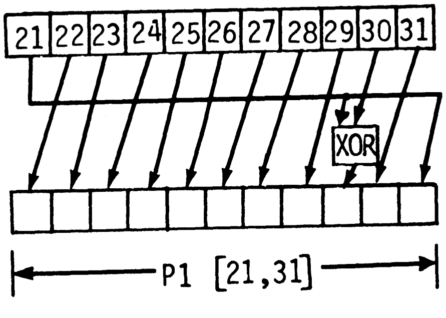


Figure 3-7. Rotation and Exclusive OR of P1

3.5.2 Error Correction Procedure

To compute bit displacement of the eleven-bit error burst, determine location of the error, and make calculated corrections proceed as follows:

1) If the value of M, calculated in steps 5 and 6 above, is greater than or equal to the value of N, calculated in steps 3 and 4 above, proceed to step 3 below. If the value of M is less than the value of N proceed to step 2 below.

2) To calculate the bit offset (R) perform the following calculations:

$$R = 19 \cdot (N-M) \cdot \text{Modulo } 21$$
$$X = 2047 \cdot R + M$$

Proceed to step 4 below.

3) To calculate the bit offset (R) perform the following calculations:

$$R = 195 \cdot (M-N) \cdot \text{Modulo } 2047$$
$$X = 21 \cdot R + N$$

Proceed to step 4 below.

4) To calculate the bit displacement of the burst error from the start of the sector (D) make the following calculations:

$$D = X - 36812$$

Proceed to step 5.

5) Test the bit displacement calculated in step 4 for a greater than zero condition ($D > 0$). If the displacement is greater than zero proceed to step 6 below. If the displacement is equal to or less than zero, proceed to step 7 below.

6) Test the bit displacement for equal to or greater than 4128 ($D \geq 4128$). If the displacement is equal to or is greater than 4128, proceed to step 8 below. If the displacement is less than 4128, proceed to step 9 below.

7) Test the bit displacement for equal to or less than -11 ($D \geq -11$). If the displacement is equal to or less than -11, proceed to step 8 below. If the displacement is greater than -11, proceed to step 13 below.

8) This error is not correctable. The bit displacement indicates the error has occurred outside the sector. Exit the routine.

9) Test the bit displacement for equal to or greater than 4096 ($D \geq 4096$). If the displacement is equal to or greater than 4096, proceed to step 11 below. If the displacement is less than 4096, proceed to step 10 below.

10) The error is in the Error Correction Code. The data transferred is correct. Exit the routine.

11) Test the bit displacement for greater than 4085 ($D > 4085$). If the displacement is greater than 4085, proceed to step 12 below. If the displacement is equal to or less than 4085, proceed to step 17 below.

12) Perform the following calculation:

$$S = D - 4085$$

Set to zero "S" number of the least significant bits of the error pattern. Proceed to step 17 below.

13) Test the most significant bit of the error pattern for equal to one (MSB = 1). If the MSB equals one (1), return to step 8 above. If the MSB equals zero (0), go to step 14 below.

14) Shift the error pattern left one position. Proceed to step 15 below.

15) Perform the following calculation:

$$D = D + 1$$

Proceed to step 16 below.

16) Test the bit displacement for equal to zero ($D = 0$). If the displacement is equal to zero, proceed to step 17 below. If the displacement is not equal to zero, return to step 13 above.

17) Perform the calculation below to properly align the bit displacement:

$$D = 16Q + R$$

(where Q = Word Offset, and R = Bit Offset)

Figure 3-8 illustrates the word and bit offsets as well as replacement of the corrected data.

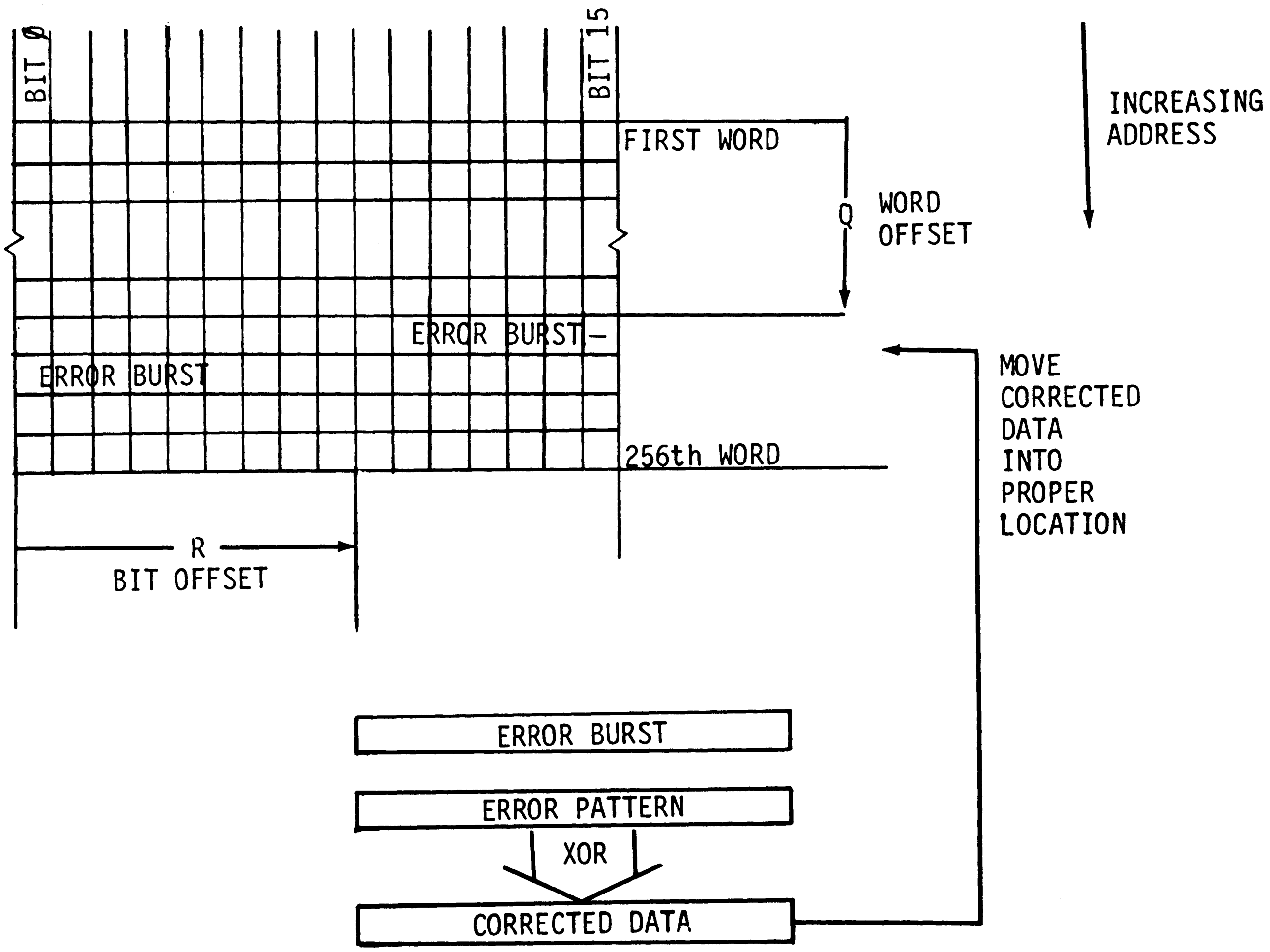


Figure 3-8. Word and Bit Offsets and Data Replacement Procedure

3.5.3 Error Detection and Correction Flowchart

Figure 3-9 is a flowchart illustrating the procedures for error detection and correction.

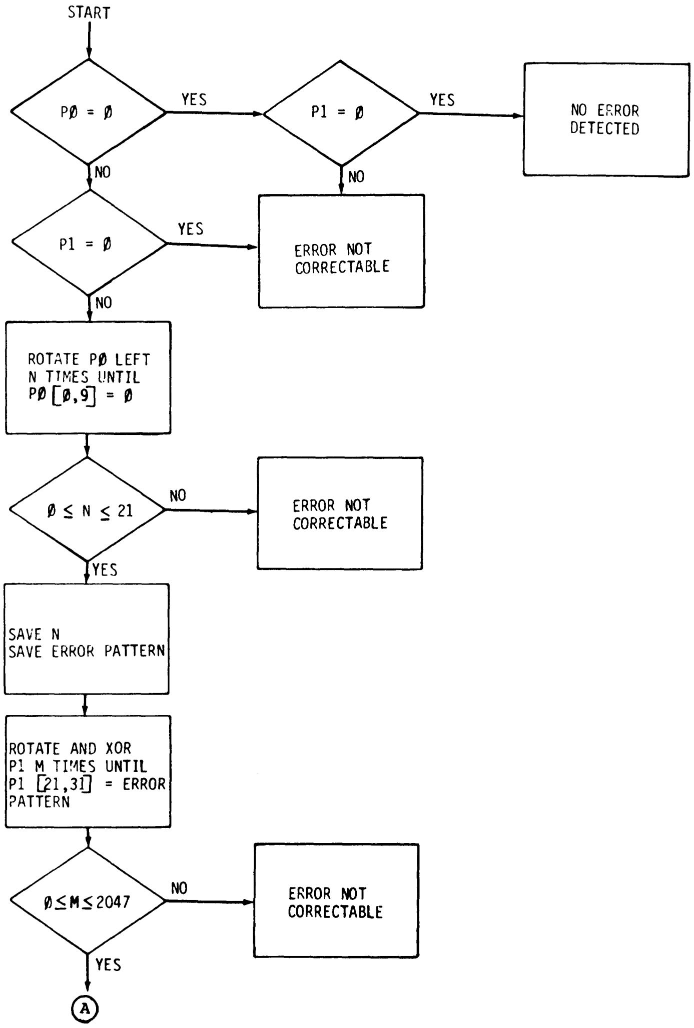
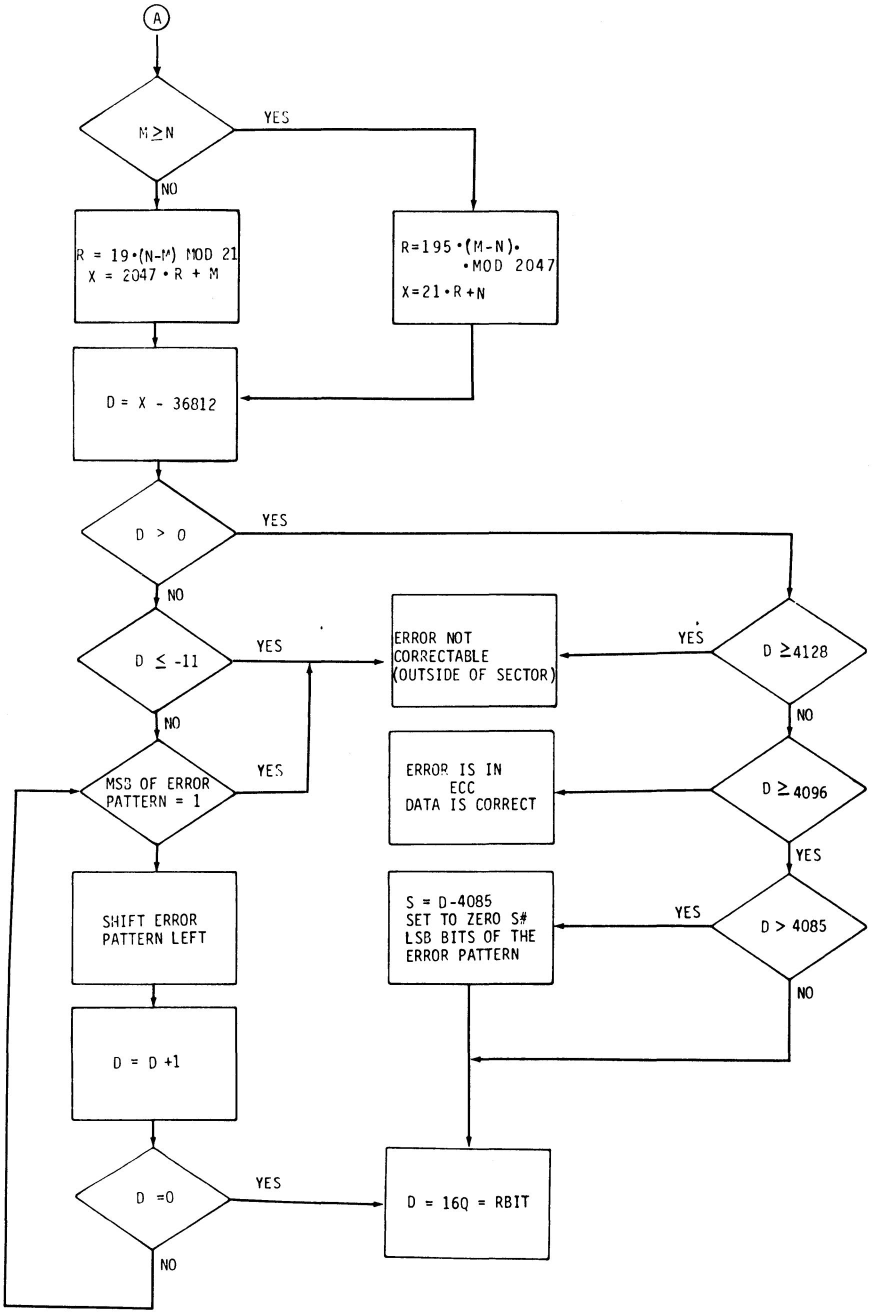


Figure 3-9. Error Detection and Correction Flowchart (Sheet 1)



D is the bit displacement of the Burst Error from the start of the sector

Q = WORD OFFSET
R = BIT OFFSET

Figure 3-9. Error Detection and Correction Flowchart (Sheet 2)

APPENDICES

Appendix A
ACCUMULATOR FORMATS

This appendix contains a summary of all the accumulator formats showing the appropriate I/O instruction and the assembly language instruction format.

SPECIFY COMMAND AND DRIVE DOA(f) ac, DSKP

CLR R/W DN	CLR DRIVE ATTN. 0 1 2 3	COMMAND	DRIVE#	VOL	XMA MSB
0	1 2 3 4	5 6 7 8	9 10	11	12 13 14 15

SPECIFY MEMORY ADDRESS DOB(f) ac, DSKP

MEMORY ADDRESS
0 15

SPECIFY CYLINDER (if seek command) DOC(f) ac, DSKP

☒	CYLINDER ADDRESS
0 5	6 15

SPECIFY HEAD,SECTOR,COUNT(if not seek command) DOC(f) ac, DSKP

☒	HEAD ADDRESS	SECTOR ADDRESS	-SECTOR COUNT
0	1 5	6 10	11 15

READ CONTROLLER STATUS DIA(f) ac, DSKP

CONTR FULL	R/W DONE	DRIVE SEEK DONE 0 1 2 3	☒	ILL SECT	ECC ERR	BAD SECT	CYL A ERR	HD A ERR	VFY ERR	R/W TIME	DATA LATE	R/W ERR
0	1	2 3 4 5	6	7	8	9	10	11	12	13	14	15

READ DRIVE STATUS DIB(f) ac, DSKP

☒	DR RES	☒	DR RDY	DR BUSY	☒	WRITE DISA	☒	ILL ADDR	ILL COM	DR FLT	DR FLT	DR FLT	☒	DR ERR
0	1	2	3	4	5	6	7	8	9	10	11	12	13 14	15

READ MEMORY ADDRESS ALT MODE 1 DIA(f) ac, DSKP

MEMORY ADDRESS
0 15

READ EXT. MEMORY ADDRESS ALT MODE 1 DIB(f) ac, DSKP

☒	XMA MSB
0 11	12 15

READ HIGH ORDER ECCR BITS ALT MODE 2 DIA(f) ac, DSKP

HIGH ORDER BITS OF ECCR RØ … R15
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

READ LOW ORDER ECCR BITS ALT MODE 2 DIB(f) ac, DSKP

LOW ORDER BITS OF ECCR R16 … R31
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

READ HEAD, SECTOR, COUNT DIC(f) ac, DSKP

☒	HEAD ADDRESS	SECTOR ADDRESS	-SECTOR COUNT
0	5	6 10	11 15

Appendix B

CONFIGURATION CHART EXAMPLES

This appendix contains Head PROM configuration chart samples for CMD32, CMD64, CMD96, MMD80, SMD80, and SMD300 disc drives, and sector configuration charts for 32 and 24 sectors.

SHEET 1

	HX	1	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	INHRST H	2	11																																
	ILL H	3	10	0																															
	LAST H	4	9																																
DRIVE 3	H4	A	5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	H3	B	6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	H2	C	7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	H1	D	4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	HØ	E	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	USA	F	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	USB	G	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	VOL	H		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CMD 96	HX	1	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	INHRST H	2	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	ILL H	3	10	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	LAST H	4	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
DRIVE 2	H4	A	5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	H3	B	6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	H2	C	7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	H1	D	4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	HØ	E	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	USA	F	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	USB	G	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	VOL	H	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CMD 64	HX	1	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	INHRST H	2	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	ILL H	3	10	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	LAST H	4	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
DRIVE 1	H4	A	5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	H3	B	6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	H2	C	7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	H1	D	4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	HØ	E	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	USA	F	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	USB	G	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	VOL	H	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CMD 32	HX	1	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	INHRST H	2	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	ILL H	3	10	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	LAST H	4	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
DRIVE Ø	H4	A	5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	H3	B	6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	H2	C	7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	H1	D	4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	HØ	E	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	USA	F	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	USB	G	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	VOL	H	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DRIVE Ø (CMD								32)				DRIVE 1 (CDM								64)				DRIVE 2 (CDM								96)				DRIVE 3											
VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX	VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX	VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX	VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX
H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1
15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12
1	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0				1
1	0	0	0	0	0	0	1	1	1	0	1	1	0	1	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	1	1	1	1	0	0	0	0	1				1
1	0	0	0	0	0	1	0	1	1	0	1	1	0	1	0	0	0	1	0	1	0	0	1	1	1	0	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0	1	0				1
1	0	0	0	0	0	1	1	1	1	0	1	1	0	1	0	0	0	1	1	1	1	0	1	1	1	0	0	0	0	1	1	0	0	0	1	1	1	1	0	0	0	1	1				1
1	0	0	0	0	1	0	0	1	1	0	1	1	0	1	0	0	1	0	0	1	1	0	1	1	1	0	0	0	1	0	0	1	0	0	1	1	1	1	0	0	1	0	0				1
1	0	0	0	0	1	0	1	1	1	0	1	1	0	1	0	0	1	0	1	1	1	0	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	1	0	0	1	0	1				1
1	0	0	0	0	1	1	0	1	1	0	1	1	0	1	0	0	1	1	0	1	1	0	1	1	1	0	0	0	1	1	0	1	1	0	1	1	1	1	0	0	1	1	0				1
1	0	0	0	0	1	1	1	1	1	0	1	1	0	1	0	0	1	1	1	1	1	0	1	1	1	0	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	1	1				1
1	0	0	0	1	0	0	0	1	1	0	1	1	0	1	0	1	0	0	0	1	1	0	1	1	1	0	0	1	0	0	0	1	1	0	1	1	1	1	0	1	0	0	0				1
1	0	0	0	1	0	0	1	1	1	0	1	1	0	1	0	1	0	0	1	1	1	0	1	1	1	0	0	1	0	0	1	1	1	0	1	1	1	1	0	1	0	0	1				1
1	0	0	0	1	0	1	0	1	1	0	1	1	0	1	0	1	0	1	0	1	1	0	1	1	1	0	0	1	0	1	0	1	1	1	0	1	1	1	0	1	0	1	0				1
1	0	0	0	1	0	1	1	1	1	0	1	1	0	1	0	1	0	1	1	1	1	0	1	1	1	0	0	1	0	1	1	1	1	0	1	1	1	1	0	1	0	1	1				1
1	0	0	0	1	1	0	0	1	1	0	1	1	0	1	0	1	1	0	0	1	1	0	1	1	1	0	0	1	1	0	0	1	1	0	1	1	1	1	0	1	1	0	0				1
1	0	0	0	1	1	0	1	1	1	0	1	1	0	1	0	1	1	0	1	1	1	0	1	1	1	0	0	1	1	0	1	1	1	0	1	1	1	1	0	1	1	0	1				1
1	0	0	0	1	1	1	0	1	1	0	1	1	0	1	0	1	1	1	0	1	1	0	1	1	1	0	0	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	0				1
1	0	0	0	1	1	1	1	1	1	0	1	1	0	1	0	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1				1
1	0	0	1	0	0	0	0	1	1	0	1	1	0	1	1	0	0	0	0	1	1	0	1	1	1	0	1	0	0	0	0	1	1	0	1	1	1	1	1	0	0	0	0	1	1	0	1
1	0	0	1	0	0	0	1	1	1	0	1	1	0	1	1	0	0	0	1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1
1	0	0	1	0	0	1	0	1	1	0	1	1	0	1	1	0	0	1	0	1	1	0	1	1	1	0	1	0	0	1	0	1	1	0	1	1	1	1	1	0	0	1	0	1	1	0	1
1	0	0	1	0	0	1	1	1	1	0	1	1	0	1	1	0	0	1	1	1	1	0	1	1	1	0	1	0	0	1	1	1	1	0	1	1	1	1	1	0	0	1	1	1	1	0	1
1	0	0	1	0	1	0	0	1	1	0	1	1	0	1	1	0	1	0	0	1	1	0	1	1	1	0	1	0	1	0	0	1	1	0	1	1	1	1	1	0	1	0	0	1	1	0	1
1	0	0	1	0	1	0	1	1	1	0	1	1	0	1	1	0	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	1	1	0	1	0	1	1	1	0	1
1	0	0	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	0	1	1	0	1
1	0	0	1	0	1	1	1	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1
1	0	0	1	1	0	0	0	1	1	0	1	1	0	1	1	1	0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	1	1	0	1	1	1	1	1	1	0	0	0	1	1	0	1
1	0	0	1	1	0	0	1	1	1	0	1	1	0	1	1	1	0	0	1	1	1	0	1	1	1	0	1	1	0	0	1	1	1	0	1	1	1	1	1	1	0	0	1	1	1	0	1
1	0	0	1	1	0	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1	1	1	0	1	0	1	1	0	1
1	0	0	1	1	0	1	1	1	1	0	1	1	0	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	0	1
1	0	0	1	1	1	0	0	1	1	0	1	1	0	1	1	1	1	0	0	1	1	0	1	1	1	0	1	1	1	0	0	1	1	0	1	1	1	1	1	1	1	0	0	1	1	0	1
1	0	0	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1
1	0	0	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	1	0	1	1	0	1	1	1	0	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1	0	1
1	0	0	1	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1

SHEET 2

DRIVE Ø							MMD SMD	30 80				DRIVE 1							SMD	300				DRIVE 2												DRIVE 3											
VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX	VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX	VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX	VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX
H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1
15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0		0		0	0	1	1	0	0	0	0	0		0		0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1				0	0	1	1	0	0	0	0	1				0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0				0	0	1	1	0	0	0	1	0				0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1	1				0	0	1	1	0	0	0	1	1				0
0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0				0	0	1	1	0	0	1	0	0				0
0	0	0	0	0	1	0	1	1	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	0	0	1	0	1				0	0	1	1	0	0	1	0	1				0
0	0	0	0	0	1	1	0	1	1	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	0	0	0	1	1	0				0	0	1	1	0	0	1	1	0				0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	1	0	0	0	1	1	1				0	0	1	1	0	0	1	1	1				0
0	0	0	0	1	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0				0	0	1	1	0	1	0	0	0				0
0	0	0	0	1	0	0	1	1	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1				0	0	1	1	0	1	0	0	1				0
0	0	0	0	1	0	1	0	1	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0	1	0				0	0	1	1	0	1	0	1	0				0
0	0	0	0	1	0	1	1	1	1	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	1	0	0	1	0	1	1				0	0	1	1	0	1	0	1	1				0
0	0	0	0	1	1	0	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0				0	0	1	1	0	1	1	0	0				0
0	0	0	0	1	1	0	1	1	1	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	1	0	0	1	1	0	1				0	0	1	1	0	1	1	0	1				0
0	0	0	0	1	1	1	0	1	1	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	1	0	0	1	1	1	0				0	0	1	1	0	1	1	1	0				0
0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	1	0	0	1	1	1	1				0	0	1	1	0	1	1	1	1				0
0	0	0	1	0	0	0	0	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0				1	0	1	1	1	0	0	0	0				1
0	0	0	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1				1	0	1	1	1	0	0	0	1				1
0	0	0	1	0	0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1	0				1	0	1	1	1	0	0	1	0				1
0	0	0	1	0	0	1	1	1	1	0	1	0	0	1	1	0	0	1	1	1	1	0	1	0	1	0	1	0	0	1	1				1	0	1	1	1	0	0	1	1				1
0	0	0	1	0	1	0	0	1	1	0	1	0	0	1	1	0	1	0	0	1	1	0	1	0	1	0	1	0	1	0	0				1	0	1	1	1	0	1	0	0				1
0	0	0	1	0	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0	1				1	0	1	1	1	0	1	0	1				1
0	0	0	1	0	1	1	0	1	1	0	1	0	0	1	1	0	1	1	0	1	1	0	1	0	1	0	1	0	1	1	0				1	0	1	1	1	0	1	1	0				1
0	0	0	1	0	1	1	1	1	1	0	1	0	0	1	1	0	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1				1	0	1	1	1	0	1	1	1				1
0	0	0	1	1	0	0	0	1	1	0	1	0	0	1	1	1	0	0	0	1	1	0	1	0	1	0	1	1	0	0	0				1	0	1	1	1	1	0	0	0				1
0	0	0	1	1	0	0	1	1	1	0	1	0	0	1	1	1	0	0	1	1	1	0	1	0	1	0	1	1	0	0	1				1	0	1	1	1	1	0	0	1				1
0	0	0	1	1	0	1	0	1	1	0	1	0	0	1	1	1	0	1	0	1	1	0	1	0	1	0	1	1	0	1	0				1	0	1	1	1	1	0	1	0				1
0	0	0	1	1	0	1	1	1	1	0	1	0	0	1	1	1	0	1	1	1	1	0	1	0	1	0	1	1	0	1	1				1	0	1	1	1	1	0	1	1				1
0	0	0	1	1	1	0	0	1	1	0	1	0	0	1	1	1	1	0	0	1	1	0	1	0	1	0	1	1	1	0	0				1	0	1	1	1	1	1	0	0				1
0	0	0	1	1	1	0	1	1	1	0	1	0	0	1	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1				1	0	1	1	1	1	1	0	1				1
0	0	0	1	1	1	1	0	1	1	0	1	0	0	1	1	1	1	1	0	1	1	0	1	0	1	0	1	1	1	1	0				1	0	1	1	1	1	1	1	0				1
0	0	0	1	1	1	1	1	1	1	0	1	0	0	1	1	1	1	1	1	1	1	0	1	0	1	0	1	1	1	1	1				1	0	1	1	1	1	1	1	1				1

SHEET 1

DRIVE Ø (SMD								80)				DRIVE 1 (SMD								300)				DRIVE 2												DRIVE 3											
VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX	VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX	VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX	VOL	USB	USA	HØ	H1	H2	H3	H4	LAST H	ILL H	INHRST H	HX
H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1
15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12
1	0	0	0	0	0	0	0	1	1	0	1	1	0	1	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0				1	1	1	1	0	0	0	0	0				1
1	0	0	0	0	0	0	1	1	1	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0	1				1	1	1	1	0	0	0	0	1				1
1	0	0	0	0	0	1	0	1	1	0	1	1	0	1	0	0	0	1	0	1	1	0	1	1	1	0	0	0	0	1	0				1	1	1	1	0	0	0	1	0				1
1	0	0	0	0	0	1	1	1	1	0	1	1	0	1	0	0	0	1	1	1	1	0	1	1	1	0	0	0	0	1	1				1	1	1	1	0	0	0	1	1				1
1	0	0	0	0	1	0	0	1	1	0	1	1	0	1	0	0	1	0	0	1	1	0	1	1	1	0	0	0	1	0	0				1	1	1	1	0	0	1	0	0				1
1	0	0	0	0	1	0	1	1	1	0	1	1	0	1	0	0	1	0	1	1	1	0	1	1	1	0	0	0	1	0	1				1	1	1	1	0	0	1	0	1				1
1	0	0	0	0	1	1	0	1	1	0	1	1	0	1	0	0	1	1	0	1	1	0	1	1	1	0	0	0	1	1	0				1	1	1	1	0	0	1	1	0				1
1	0	0	0	0	1	1	1	1	1	0	1	1	0	1	0	0	1	1	1	1	1	0	1	1	1	0	0	0	1	1	1				1	1	1	1	0	0	1	1	1				1
1	0	0	0	1	0	0	0	1	1	0	1	1	0	1	0	1	0	0	0	1	1	0	1	1	1	0	0	1	0	0	0				1	1	1	1	0	1	0	0	0				1
1	0	0	0	1	0	0	1	1	1	0	1	1	0	1	0	1	0	0	1	1	1	0	1	1	1	0	0	1	0	0	1				1	1	1	1	0	1	0	0	1				1
1	0	0	0	1	0	1	0	1	1	0	1	1	0	1	0	1	0	1	0	1	1	0	1	1	1	0	0	1	0	1	0				1	1	1	1	0	1	0	1	0				1
1	0	0	0	1	0	1	1	1	1	0	1	1	0	1	0	1	0	1	1	1	1	0	1	1	1	0	0	1	0	1	1				1	1	1	1	0	1	0	1	1				1
1	0	0	0	1	1	0	0	1	1	0	1	1	0	1	0	1	1	0	0	1	1	0	1	1	1	0	0	1	1	0	0				1	1	1	1	0	1	1	0	0				1
1	0	0	0	1	1	0	1	1	1	0	1	1	0	1	0	1	1	0	1	1	1	0	1	1	1	0	0	1	1	0	1				1	1	1	1	0	1	1	0	1				1
1	0	0	0	1	1	1	0	1	1	0	1	1	0	1	0	1	1	1	0	1	1	0	1	1	1	0	0	1	1	1	0				1	1	1	1	0	1	1	1	0				1
1	0	0	0	1	1	1	1	1	1	0	1	1	0	1	0	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1				1	1	1	1	0	1	1	1	1				1
1	0	0	1	0	0	0	0	1	1	0	1	1	0	1	1	0	0	0	0	1	1	0	1	1	1	0	1	0	0	0	0	1	1	0	1	1	1	1	1	0	0	0	0	1	1	0	1
1	0	0	1	0	0	0	1	1	1	0	1	1	0	1	1	0	0	0	1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1
1	0	0	1	0	0	1	0	1	1	0	1	1	0	1	1	0	0	1	0	1	1	0	1	1	1	0	1	0	0	1	0	1	1	0	1	1	1	1	1	0	0	1	0	1	1	0	1
1	0	0	1	0	0	1	1	1	1	0	1	1	0	1	1	0	0	1	1	1	1	0	1	1	1	0	1	0	0	1	1	1	1	0	1	1	1	1	1	0	0	1	1	1	1	0	1
1	0	0	1	0	1	0	0	1	1	0	1	1	0	1	1	0	1	0	0	1	1	0	1	1	1	0	1	0	1	0	0	1	1	0	1	1	1	1	1	0	1	0	0	1	1	0	1
1	0	0	1	0	1	0	1	1	1	0	1	1	0	1	1	0	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	1	1	0	1	0	1	1	1	0	1
1	0	0	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	0	1	1	0	1
1	0	0	1	0	1	1	1	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1
1	0	0	1	1	0	0	0	1	1	0	1	1	0	1	1	1	0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	1	1	0	1	1	1	1	1	1	0	0	0	1	1	0	1
1	0	0	1	1	0	0	1	1	1	0	1	1	0	1	1	1	0	0	1	1	1	0	1	1	1	0	1	1	0	0	1	1	1	0	1	1	1	1	1	1	0	0	1	1	1	0	1
1	0	0	1	1	0	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1	1	1	0	1	0	1	1	0	1
1	0	0	1	1	0	1	1	1	1	0	1	1	0	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	0	1
1	0	0	1	1	1	0	0	1	1	0	1	1	0	1	1	1	1	0	0	1	1	0	1	1	1	0	1	1	1	0	0	1	1	0	1	1	1	1	1	1	1	0	0	1	1	0	1
1	0	0	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1
1	0	0	1	1	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	0	1	1	1	0	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1	0	1
1	0	0	1	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1

SHEET 2

32 SECTORS												24 SECTORS											
DRIVE Ø												DRIVE 1											
ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S	ILL S			ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S	ILL S		
H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1
15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9	10	11	12
0	0	0	0	0	0	0	0	0	1			0	0	1	0	0	0	0	0	0	1		
0	0	0	0	0	0	0	1	0	1			0	0	1	0	0	0	0	1	0	1		
0	0	0	0	0	0	1	0	0	1			0	0	1	0	0	0	1	0	0	1		
0	0	0	0	0	0	1	1	0	1			0	0	1	0	0	0	1	1	0	1		
0	0	0	0	0	1	0	0	0	1			0	0	1	0	0	1	0	0	0	1		
0	0	0	0	0	1	0	1	0	1			0	0	1	0	0	1	0	1	0	1		
0	0	0	0	0	1	1	0	0	1			0	0	1	0	0	1	1	0	0	1		
0	0	0	0	0	1	1	1	0	1			0	0	1	0	0	1	1	1	0	1		
0	0	0	0	1	0	0	0	0	1			0	0	1	0	1	0	0	0	0	1		
0	0	0	0	1	0	0	1	0	1			0	0	1	0	1	0	0	1	0	1		
0	0	0	0	1	0	1	0	0	1			0	0	1	0	1	0	1	0	0	1		
0	0	0	0	1	0	1	1	0	1			0	0	1	0	1	0	1	1	0	1		
0	0	0	0	1	1	0	0	0	1			0	0	1	0	1	1	0	0	0	1		
0	0	0	0	1	1	0	1	0	1			0	0	1	0	1	1	0	1	0	1		
0	0	0	0	1	1	1	0	0	1			0	0	1	0	1	1	1	0	0	1		
0	0	0	0	1	1	1	1	0	1			0	0	1	0	1	1	1	1	0	1		
0	0	0	1	0	0	0	0	0	1			0	0	1	1	0	0	0	0	0	1		
0	0	0	1	0	0	0	1	0	1			0	0	1	1	0	0	0	1	0	1		
0	0	0	1	0	0	1	0	0	1			0	0	1	1	0	0	1	0	0	1		
0	0	0	1	0	0	1	1	0	1			0	0	1	1	0	0	1	1	0	1		
0	0	0	1	0	1	0	0	0	1			0	0	1	1	0	1	0	0	0	1		
0	0	0	1	0	1	0	1	0	1			0	0	1	1	0	1	0	1	0	1		
0	0	0	1	0	1	1	0	0	1			0	0	1	1	0	1	1	0	0	1		
0	0	0	1	0	1	1	1	0	1			0	0	1	1	1	0	1	1	1	1		
0	0	0	1	1	0	0	0	0	1			0	0	1	1	1	0	0	0	1	0		
0	0	0	1	1	0	0	1	0	1			0	0	1	1	1	0	0	1	1	0		
0	0	0	1	1	0	1	0	0	1			0	0	1	1	1	0	1	0	1	0		
0	0	0	1	1	0	1	1	0	1			0	0	1	1	1	0	1	1	1	0		
0	0	0	1	1	1	0	0	0	1			0	0	1	1	1	1	0	0	1	0		
0	0	0	1	1	1	0	1	0	1			0	0	1	1	1	1	0	1	1	0		
0	0	0	1	1	1	1	0	0	1			0	0	1	1	1	1	1	0	1	0		
0	0	0	1	1	1	1	1	1	1			0	0	1	1	1	1	1	1	1	0		

DRIVE 2												DRIVE 3											
ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S	ILL S			ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S	ILL S		
H	G	F	E	D	C	B	A	4	3	2	1	H	G	F	E	D	C	B	A	4	3	2	1
15	1	2	3	4	7	6	5	9	10	11	12	15	1	2	3	4	7	6	5	9			
0	1	0	0	0	0	0	0		1			0	1	1	0	0	0	0	0		1		
0	1	0	0	0	0	0	1					0	1	1	0	0	0	0	1				
0	1	0	0	0	0	1	0					0	1	1	0	0	0	1	0				
0	1	0	0	0	0	1	1					0	1	1	0	0	0	1	1				
0	1	0	0	0	1	0	0					0	1	1	0	0	1	0	0				
0	1	0	0	0	1	0	1					0	1	1	0	0	1	0	1				
0	1	0	0	0	1	1	0					0	1	1	0	0	1	1	0				
0	1	0	0	0	1	1	1					0	1	1	0	0	1	1	1				
0	1	0	0	1	0	0	0					0	1	1	0	1	0	0	0				
0	1	0	0	1	0	0	1					0	1	1	0	1	0	0	1				
0	1	0	0	1	0	1	0					0	1	1	0	1	0	1	0				
0	1	0	0	1	0	1	1					0	1	1	0	1	0	1	1				
0	1	0	0	1	1	0	0					0	1	1	0	1	1	0	0				
0	1	0	0	1	1	0	1					0	1	1	0	1	1	0	1				
0	1	0	0	1	1	1	0					0	1	1	0	1	1	1	0				
0	1	0	0	1	1	1	1					0	1	1	0	1	1	1	1				
0	1	0	1	0	0	0	0					0	1	1	1	0	0	0	0				
0	1	0	1	0	0	0	1					0	1	1	1	0	0	0	1				
0	1	0	1	0	0	1	0					0	1	1	1	0	0	1	0				
0	1	0	1	0	0	1	1					0	1	1	1	0	0	1	1				
0	1	0	1	0	1	0	0					0	1	1	1	0	1	0	0				
0	1	0	1	0	1	0	1					0	1	1	1	0	1	0	1				
0	1	0	1	0	1	1	0					0	1	1	1	0	1	1	0				
0	1	0	1	0	1	1	1					0	1	1	1	0	1	1	1				
0	1	0	1	1	0	0	0					0	1	1	1	1	0	0	0				
0	1	0	1	1	0	0	1					0	1	1	1	1	0	0	1				
0	1	0	1	1	0	1	0					0	1	1	1	1	0	1	0				
0	1	0	1	1	0	1	1					0	1	1	1	1	0	1	1				
0	1	0	1	1	1	0	0					0	1	1	1	1	1	0	0				
0	1	0	1	1	1	0	1					0	1	1	1	1	1	0	1				
0	1	0	1	1	1	1	0					0	1	1	1	1	1	1	0				
0	1	0	1	1	1	1	1					0	1	1	1	1	1	1	1				

SHEET 1

SHEET 2

DRIVE Ø 32 SECTORS											
ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S	ILL S		
H	G	F	E	D	C	B	A	4	3	2	1
15	1	2	3	4	7	6	5	9	10	11	12
1	0	0	0	0	0	0	0	1	0		
1	0	0	0	0	0	0	1	1	0		
1	0	0	0	0	0	1	0	1	0		
1	0	0	0	0	0	1	1	1	0		
1	0	0	0	0	1	0	0	1	0		
1	0	0	0	0	1	0	1	1	0		
1	0	0	0	0	1	1	0	1	0		
1	0	0	0	0	1	1	1	1	0		
1	0	0	0	1	0	0	0	1	0		
1	0	0	0	1	0	0	1	1	0		
1	0	0	0	1	0	1	0	1	0		
1	0	0	0	1	0	1	1	1	0		
1	0	0	0	1	1	0	0	1	0		
1	0	0	0	1	1	0	1	1	0		
1	0	0	0	1	1	1	0	1	0		
1	0	0	0	1	1	1	1	1	0		
1	0	0	1	0	0	0	0	1	0		
1	0	0	1	0	0	0	1	1	0		
1	0	0	1	0	0	1	0	1	0		
1	0	0	1	0	0	1	1	1	0		
1	0	0	1	0	1	0	0	1	0		
1	0	0	1	0	1	0	1	1	0		
1	0	0	1	0	1	1	0	1	0		
1	0	0	1	0	1	1	1	1	0		
1	0	0	1	1	0	0	0	1	0		
1	0	0	1	1	0	0	1	1	0		
1	0	0	1	1	0	1	0	1	0		
1	0	0	1	1	0	1	1	1	0		
1	0	0	1	1	1	0	0	1	0		
1	0	0	1	1	1	0	1	1	0		
1	0	0	1	1	1	1	0	1	0		
1	0	0	1	1	1	1	1	1	0		

DRIVE 1 24 SECTORS											
ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S	ILL S		
H	G	F	E	D	C	B	A	4	3	2	1
15	1	2	3	4	7	6	5	9	10	11	12
1	0	1	0	0	0	0	0	1	0		
1	0	1	0	0	0	0	1	1	0		
1	0	1	0	0	0	1	0	1	0		
1	0	1	0	0	0	1	1	1	0		
1	0	1	0	0	1	0	0	1	0		
1	0	1	0	0	1	0	1	1	0		
1	0	1	0	0	1	1	0	1	0		
1	0	1	0	0	1	1	1	1	0		
1	0	1	0	1	0	0	0	1	0		
1	0	1	0	1	0	0	1	1	0		
1	0	1	0	1	0	1	0	1	0		
1	0	1	0	1	0	1	1	1	0		
1	0	1	0	1	1	0	0	1	0		
1	0	1	0	1	1	0	1	1	0		
1	0	1	0	1	1	1	0	1	0		
1	0	1	0	1	1	1	1	1	0		
1	0	1	1	0	0	0	0	1	0		
1	0	1	1	0	0	0	1	1	0		
1	0	1	1	0	0	1	0	1	0		
1	0	1	1	0	0	1	1	1	0		
1	0	1	1	0	1	0	0	1	0		
1	0	1	1	0	1	0	1	1	0		
1	0	1	1	0	1	1	0	1	0		
1	0	1	1	0	1	1	1	1	0		
1	0	1	1	1	0	0	0	1	0		
1	0	1	1	1	0	0	1	1	0		
1	0	1	1	1	0	1	0	1	0		
1	0	1	1	1	0	1	1	1	0		
1	0	1	1	1	1	0	0	1	0		
1	0	1	1	1	1	0	1	1	0		
1	0	1	1	1	1	1	0	1	0		
1	0	1	1	1	1	1	1	1	0		

DRIVE 2											
ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S	ILL S		
H	G	F	E	D	C	B	A	4	3	2	1
15	1	2	3	4	7	6	5	9	10	11	12
1	1	0	0	0	0	0	0	1	0		
1	1	0	0	0	0	0	1	1	0		
1	1	0	0	0	0	1	0	1	0		
1	1	0	0	0	0	1	1	1	0		
1	1	0	0	0	1	0	0	1	0		
1	1	0	0	0	1	0	1	1	0		
1	1	0	0	0	1	1	0	1	0		
1	1	0	0	0	1	1	1	1	0		
1	1	0	0	1	0	0	0	1	0		
1	1	0	0	1	0	0	1	1	0		
1	1	0	0	1	0	1	0	1	0		
1	1	0	0	1	0	1	1	1	0		
1	1	0	0	1	1	0	0	1	0		
1	1	0	0	1	1	0	1	1	0		
1	1	0	0	1	1	1	0	1	0		
1	1	0	0	1	1	1	1	1	0		
1	1	0	1	0	0	0	0	1	0		
1	1	0	1	0	0	0	1	1	0		
1	1	0	1	0	0	1	0	1	0		
1	1	0	1	0	0	1	1	1	0		
1	1	0	1	0	1	0	0	1	0		
1	1	0	1	0	1	0	1	1	0		
1	1	0	1	0	1	1	0	1	0		
1	1	0	1	0	1	1	1	1	0		
1	1	0	1	1	0	0	0	1	0		
1	1	0	1	1	0	0	1	1	0		
1	1	0	1	1	0	1	0	1	0		
1	1	0	1	1	0	1	1	1	0		
1	1	0	1	1	1	0	0	1	0		
1	1	0	1	1	1	0	1	1	0		
1	1	0	1	1	1	1	0	1	0		
1	1	0	1	1	1	1	1	1	0		

DRIVE 3											
ILL H	USB	USA	SØ	S1	S2	S3	S4	LAST S	ILL S		
H	G	F	E	D	C	B	A	4	3	2	1
15	1	2	3	4	7	6	5	9	10	11	12
1	1	1	0	0	0	0	0	1	0		
1	1	1	0	0	0	0	1	1	0		
1	1	1	0	0	0	1	0	1	0		
1	1	1	0	0	0	1	1	1	0		
1	1	1	0	0	1	0	0	1	0		
1	1	1	0	0	1	0	1	1	0		
1	1	1	0	0	1	1	0	1	0		
1	1	1	0	0	1	1	1	1	0		
1	1	1	0	1	0	0	0	1	0		
1	1	1	0	1	0	0	1	1	0		
1	1	1	0	1	0	1	0	1	0		
1	1	1	0	1	0	1	1	1	0		
1	1	1	0	1	1	0	0	1	0		
1	1	1	0	1	1	0	1	1	0		
1	1	1	0	1	1	1	0	1	0		
1	1	1	0	1	1	1	1	1	0		
1	1	1	1	0	0	0	0	1	0		
1	1	1	1	0	0	0	1	1	0		
1	1	1	1	0	0	1	0	1	0		
1	1	1	1	0	0	1	1	1	0		
1	1	1	1	0	1	0	0	1	0		
1	1	1	1	0	1	0	1	1	0		
1	1	1	1	0	1	1	0	1	0		
1	1	1	1	0	1	1	1	1	0		
1	1	1	1	1	0	0	0	1	0		
1	1	1	1	1	0	0	1	1	0		
1	1	1	1	1	0	1	0	1	0		
1	1	1	1	1	0	1	1	1	0		
1	1	1	1	1	1	0	0	1	0		
1	1	1	1	1	1	0	1	1	0		
1	1	1	1	1	1	1	0	1	0		
1	1	1	1	1	1	1	1	1	0		

COMMENT SHEET

MANUAL TITLE 700 Disc Controller User Manual

PUBLICATION NO. HM-121-0012 REVISION A

FROM: NAME/COMPANY: ______

BUSINESS ADDRESS: ______

CITY/STATE/ZIP: ______

COMMENTS:

Your evaluation of this manual will be appreciated by POINT 4 Data Corporation. Notation of any errors, suggested additions or deletions, or general comments may be made below. Please include page number references where appropriate.

NO POSTAGE NECESSARY IF MAILED IN U.S.A.

Fold on Dotted Lines and Tape

NO POSTAGE
NECESSARY
IF MAILED IN
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 5755 SANTA ANA, CA.

POSTAGE WILL BE PAID BY ADDRESSEE:

POINT 4 Data Corporation
PUBLICATIONS DEPARTMENT
2569 McCabe Way
Irvine, CA 92714

CUT ON THIS LINE

POINT 4 DATA CORPORATION

2569 McCabe Way / Irvine, California 92714 / (714) 863-1111