POINT 4

DATA CORPORATION

IRIS R8.2C1

INSTALLATION/ CONFIGURATION MANUAL

Revision 12

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) 863-1111

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 package incorporating R8.1 enhancements	08/10/82
06	Complete revision incorporating new material relating to R8.1	03/01/83
07	Reissue incorporating R8.2 enhancements	08/01/83
08	Update package incorporating additional R8.2 changes	10/01/83
09	Update package incorporating corrections, additions, deletions relating to R8.2	12/01/83
10	Update package incorporating R8.2B enhancements, corrections, additions, and deletions	02/01/84
11	Update package incorporating R8.2C enhancements (including MARK 9 support), corrections and additions	06/10/84
12	Update package incorporating general and R8.2C1 enhancements, corrections, additions, and deletions	11/10/84

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. The effective revision for each page is shown below.

Page	Rev	Page	Rev	Page	Rev
Cover	-	4-32 thru 4-34	08	5-92 thru 5-98	07
Title	12	4-35	11	5-99	10
ii	08	4-36, 4-37	09	5-100	07
iii, iv	12	5-1 thru 5-3	07	5-101 thru 5-103	10
v	11	5-4	11	5-104 thru 5-120	12
vi	12	5-5	07	6-1	07
vii thru ix	11	5-6, 5-7	11	6-2	08
x	09	5-8	08	6-3	07
xi	12	5-9	07	6-4	08
xii	08	5-10	11	6-5	07
xiii	11	5-11 thru 5-14	09	6-6	08
xiv, xv	12	5-15 thru 5-17	10	6-7	07
1-1	06	5-18	12	6-8	08
1-2, 1-3	08	5-19 thru 5-31	10	6-9	10
1-4	06	5-32, 5-33	07	6-10	08
1-5 thru 1-8	07	5-34	12	6-11, 6-12	07
1-9 thru 1-16	11	5-35, 5-36	11	6-13, 6-14	08
1-17	07	5-37 thru 5-47	07	6-15 thru 6-30	07
2-1, 2-2	07	5-48, 5-49	08	6-31	08
2-3 thru 2-40	11	5-50	07	6-32 thru 6-34	07
3-1	07	5-51	08	6-35 thru 6-37	08
3-2, 3-3	08	5-52	10	6-38	07
3-4 thru 3-8	11	5-53	07	Appendix Title	-
3-9, 3-10	12	5-54	08	A-1	11
3-11	11	5-55 thru 5-59	07	A-2, A-3	12
4-1	09	5-60, 5-61	09	A-4 thru A-6	11
4-2	08	5-62 thru 5-70	07	B-1 thru B-29	07
4-3 thru 4-5	06	5-71	09	C-1 thru C-10	09
4-6	08	5-72	11	D-1 thru D-6	11
4-7, 4-8	06	5-73	07	Comment Sheet	12
4-9	08	5-74	11	Mailer	-
4-10 thru 4-12	06	5-75	09	Back Cover	-
4-13	12	5-76 thru 5-78	08		
4-14,4-15	11	5-79 thru 5-82	07		
4-16	09	5-83	10		
4-17	08	5-84, 5-85	07		
4-18	07	5-86	08		
4-19 thru 4-30	08	5-87 thru 5-90	07		
4-31	11	5-91	12		

PREFACE

The IRIS Installation and Configuration Manual has two functions. It is a guide to the initial installation and configuration of the IRIS Operating System. The manual is also a reference guide for the customization of the IRIS Operating System beyond the initial configuration process. To accommodate the dual purpose of this manual, Section 1 serves as a general introduction to the components that make up the IRIS Operating System. Sections 2, 4 and 6 describe the use of the various configuration "tools" supplied by POINT 4 which include debugging packages, utility programs, and the system configurator called SETUP. Section 3 covers recommended methods for loading the software from various media such as disc, diskette, and streamer tape. Section 5 details configuration requirements and procedures 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:

IRIS Component Checklists	Appendix A
Software Definitions	Appendix B
LPTD Driver File Listing	Appendix C
Contents of the First Four Blocks of CONFIG	Appendix D

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 lower case, 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 string represents a variable such as a filename, password, etc.
{option}	Lowercase string enclosed in braces represents an optional parameter.

Related Manuals

For related information see the following:

Title	Pub. Number
IRIS R8 Operations Manual	SM-030-0010
IRIS R8 Peripherals Handbook	SM-030-0015
IRIS R8 User Manual	SM-030-0011
IRIS R8 LCM Installation Document	
IRIS R8 Release Notes	
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 (MARK 5) Computer User Manual	
LOTUS DISCUTILITY Manual	SM-035-0018

CONTENTS

APPENDICES

INDEX

TO BE SUPPLIED

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. It draws power from pins A97 through A100. For a POINT 4 MARK 3 system, the Pico-N is installed on the CPU board (see Figure 1-1). For all other CPUs, the Pico-N is installed on the computer backplane (see Figure 1-2).

The Pico-N may be destroyed if installed incorrectly, e.g., shifted either right or left. The procedure for installing the Pico-N is as follows:

1. Turn off CPU power.

2. Install the Pico-N using step a for a MARK 3 CPU or step b for any other CPU:

   a. On a MARK 3 system, seat the Pico-N on the CPU board by pushing it onto the P2 connector, aligning pin A1 with board pin 1.

   b. Push the Pico-N's connector over the 'A' side pins of any slot except the slot which contains the CPU. It does not matter whether the selected slot contains a board.

3. Be sure the Pico-N is oriented properly, and all 100 pins are in their connectors.

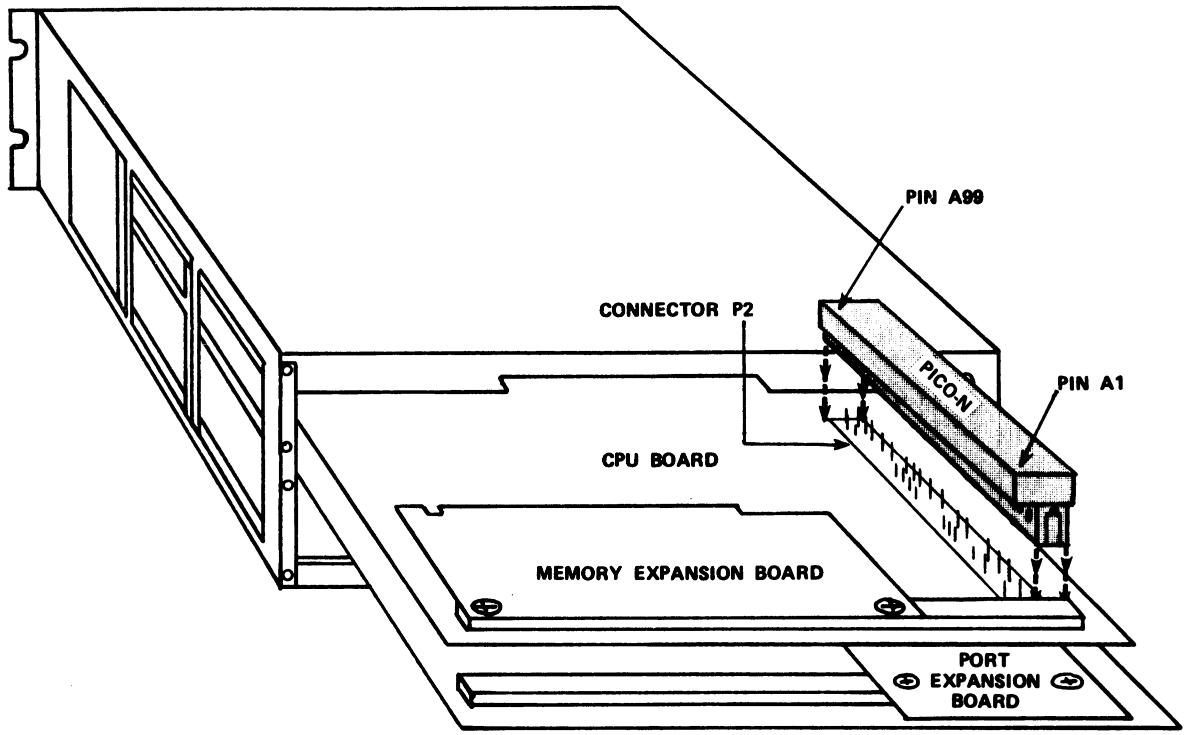


Figure 1-1. Installation of Pico-N on a MARK 3 CPU Board

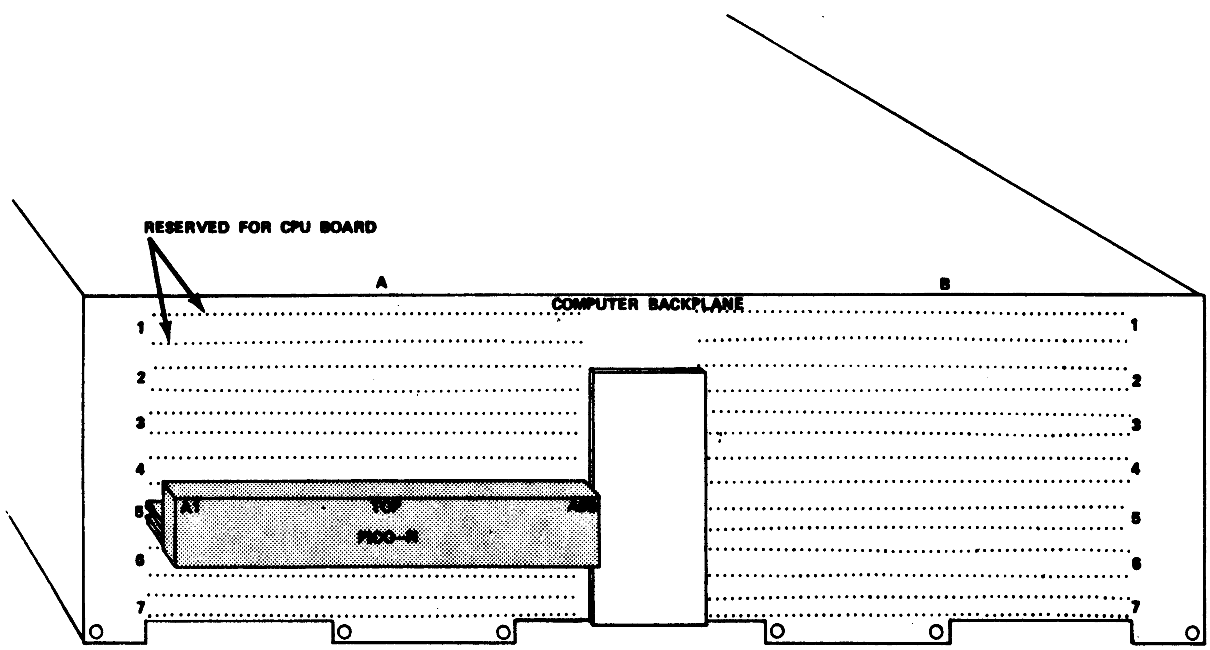


Figure 1-2. Installation of Pico-N on the Computer Backplane

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. All disc controllers
2. All magnetic tape controllers
3. POINT 4 MIGHTY MUX
4. 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. In a POINT 4 system, the CPU board occupies the top slot in the chassis. However, this location does not relate to 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 supplied on logical unit zero (LU/0), some on logical unit 5, some are stand-alone programs while others are created when the system is configured. Refer to Appendix A for checklists of IRIS components residing on LU/0 and LU/5.

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 with one supplementary module (ACTUTIL.1) used for the maintenance of the account directory (i.e., ACCOUNTS file).
ALOAD	BASIC program used to merge binary files.
ANALYPF	BASIC program used to analyze polyfile volume structures.
ASSEMBLE (ASM)	Absolute Assembler processor.
ASSIGNPF	BASIC program with four supplementary modules (ASSIGNPF1, ASSIGNPF2, ASSIGNPF2A, and ASSIGNPF3) used to move polyfile volumes from one logical unit to another.
BAKUP	Backup utility program with three additional modules (BAKUPMAIN, BAKUPCONFIG, and BAKUPINIT) used for on-line disc backups.
BASIC	BASIC language editor and lister.
BASICTEST	BASIC readiness test.

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

Name	Function
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. Does not appear on a LIBR listing.
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.
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. Does not appear on a LIBR listing.
CALLTBL	System driver containing the call table that links subroutine names and numbers to discsub numbers.
CHANGE	Processor used to change file characteristics.
CLEANUP	Processor used to realign block usage on a logical unit.

(Table continues on next page)

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

Name	Function
CLEANUPX	Similar to CLEANUP; allows movement of data between two logical units that are not alike.
COMA	IRIS interface driver for an Airland-type protocol converter.
COMD	IRIS interface driver for the Datalynx protocol converter.
CONFIG	System file which contains information about the system's current configuration.
COPY	Processor used to copy files (except polyfiles).
COPYPF	BASIC program with five supplementary modules (COPYPF1, COPYPF2, COPYPF2A, COPYPF3, and COPYPF4) used to copy polyfiles; includes option to change polyfile names.
COREMAP	BASIC program used after an IPL to display memory allocations performed by SIR.
CTR	System software diagnostic routine used to analyze the activity at an installation.
CTUS	Physical interface handler for cassette tape units.
DBUG	Stand-alone debugging utility package for the POINT 4 MARK series and Nova*-type computers. Does not appear on a LIBR listing.

*Nova is a trademark of Data General Corporation.

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

Name	Function
DC700	Diagnostic program for a POINT 4 LOTUS 700 Controller.
DDCOPY	Disc-to-disc copy utility - Unique for each disc controller.
DEC	Decimal arithmetic system driver.
DEFS	IRIS software definitions (see Appendix B).
DGMX	Driver for a Data General 4060-type multiplexer.*
DISCSUBS	File containing disc-resident IRIS system subroutines.
DISCUTILITY	Disc utility for POINT 4 LOTUS disc subsystems.
DISCUTILITY3.2	Disc utility for POINT 4 MARK 3 disc subsystem.
DISPLAY	BASIC program used to display a text file.
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.
*Not shipped for a POINT 4 MARK 3.	

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

Name	Function
EIS	Driver for MARK 8/9 Extended Instruction Set. Supersedes $MK8.
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.
FINDFILE	BASIC program used to search all installed logical units for a specified file.
FLBOOT	Utility program used to boot software from diskettes. For POINT 4 MARK 3 Systems only.
FOREIGN	Disc driver used to read/write a specified sector from or to any disc (regardless of file system) that POINT 4 hardware can support.
FORGE	IRIS BASIC program editor with seven supplementary modules (FORGE1, FORGE2, FORGE21, FORGE22, FORGE23, FORGE3, and FORGE4).
FORMAT	Processor used to create a formatted or contiguous data file.
GUARD	BASIC program used to manipulate the DOOM bits in the special access control word for File Maintenance procedures.
GUIDE	BASIC program with three supplementary modules (GUIDE.LU, GUIDE.LPT, and GUIDE.BLKCPY) that give directions for the configuration of logical units, line printer drivers, the use of BLOCKCOPY, etc.

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

Name	Function
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.
LCM	Driver for the LOTUS Cache Memory.*
LCMACTIVATE	Processor used to bring the LOTUS Cache Memory on-line.*
LCMC	BASIC program with three supplementary modules (LCMC.1, LCMC.2, and LCMC.3) used to configure a LOTUS Cache Memory.*
LCMCHECK	BASIC program used to tabulate statistics on LOTUS Cache Memory activity.*
LCMDIAG1.3	POINT 4 diagnostic program for POINT 4 LOTUS Cache Memory.*
LCMREMOVE	Processor used to place the LOTUS Cache Memory off-line.*
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.*
*Not shipped for a POINT 4 MARK 3.	

TABLE 1-1. STANDARD IRIS R8 SYSTEM COMPONENTS

Name	Function
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).*
M2DISCUTILITY	Disc utility for a POINT 4 MARK 2 disc subsystem.
M2TAPEDIAG	Tape diagnostic program for a POINT 4 MARK 2 System only.
M3DISCDIAG	Disc diagnostic program for a POINT 4 MARK 3 System only.
M3FLOPPYDIAG	Diskette diagnostic program for a POINT 4 MARK 3 System only.
M3GUIDELPT	BASIC program used to configure a line printer for a MARK 3 CPU.
M3MUXDIAG	Diagnostic program for a POINT 4 310 Multiplexer.
M3TAPEDIAG	Tape diagnostic program for a POINT 4 MARK 3 System only.
M8EXERCISER	Diagnostic program for a POINT 4 MARK 5 with EIS or MARK 8 System only.
MAGTAPE	Utility program with five supplementary modules (MAGTAPE.LOAD, MAGTAPE1, MAGTAPE11, MAGTAPE2, and MAGTAPE21) used to spool files disc-to-tape and tape-to-disc.
*Not shipped for a POINT 4 MARK 3.	

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

Name	Function
MAKEBIN	BASIC program used to convert MAKEHEXed files to binary format.
MAKEHEX	BASIC program used to convert a binary file to a hexadecimal representation thereby omitting control characters.
MAIL	Processor for sending messages from one port to another.
MAPACTIVATE	BASIC program used to activate the SYSMAP driver (for a POINT 4 MARK 9 only).
MAPCHECK	BASIC program that tabulates the activity of the SYSMAP driver (for a POINT 4 MARK 9 only).
MESSAGES	File containing standard messages.
MMUX	POINT 4 310 Multiplexer driver.
MONITOR	POINT 4 utility program. Used by Customer Support to tune and diagnose the system.
MTA0	BASIC program interface driver for nine-track magnetic tape or cassette tape unit.
MTAS	Interface driver for a magnetic tape physical unit.*
PHA	Phantom port driver.
PLOAD	Program loader used to load new files from paper tape.*
*Not shipped for a POINT 4 MARK 3.	

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

Name	Function
PORT	Processor used to change port attributes and display port activity.
PROTECT	Processor used to save BASIC programs in an unlistable format.
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.*
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.
RECEIVE	BASIC program used to receive a text file from another system using a tri-tail switch.
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.
*Not shipped for a POINT 4 MARK 3.	

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

Name	Function
RENUMBER	BASIC program with three supplementary modules (RENUMBER1, RENUMBER2, and RENUMBER3) used to renumber the line numbers of a BASIC program that is in text file format.
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 (#) indicates that the system is active and ready for input.
SETTIME	BASIC program used to set system date and time.
SETUP	BASIC utility program with the following supplementary modules and parameter files: SU1 SU114 SU23 SU34 SU.DSUBS SU11 SU114A SU24 SU4 SU.ENTRIES SU111 SU11A SU3 SU41 SU112 SU2 SU31 SU42 SU112A SU21 SU32 SU113 SU22 SU33 SETUP is used to configure the System INFO Table, Port Definition Table (PDT), Disc Driver Table, and the Discsubs Table.

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

Name	Function
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. Does not appear in a LIBR listing.
STBOOTM3	Utility program used to boot software from streamer tape. May also be used in conjunction with DISCUTILITY to copy disc-to-streamer tape. For a POINT 4 MARK 3 System only.
STBOOTM5	Utility program used to boot software from streamer tape. For a POINT 4 MARK 5 or 9 System only.
SYMBOLS	File containing the Assembler symbols.
SYS.SCHED	Driver for the IRIS system scheduler.
SYSMAP	System driver for a POINT 4 MARK 9 only.
TERM.name	Terminal Translation Module - Contains translation tables and subroutines; unique for each type of terminal. Refer to Appendix A for a list of specific modules.
TERMS	Terminal Translator system subroutine module.
TRANSMIT	BASIC program used to transmit a text file from one system to another using a tri-tail switch.

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

Name	Function
TTY	Driver for a secondary Teletype or CRT using device code 50/51.
U.CHANGE	BASIC program with one supplementary module (U.CHANGE1) used to change the protection level of selected files residing on a logical unit in one job stream.
U.CONVERT	BASIC program with one supplementary module (U.CONVERT1) used to convert selected BASIC programs on a given account from R7 format to R8 format in one job stream.
U.COPY	BASIC program with one supplementary module (U.COPY1) used to copy selected files residing on a logical unit in one job stream.
U.KILL	BASIC program with one supplementary module (U.KILL1) used to delete selected files residing on a logical unit in one job stream.
U.PROTECT	BASIC program with one supplementary module (U.PROTECT1) used to make selected BASIC source code modules unlistable in one job stream.
U.SAVE	BASIC program with one supplementary module (U.SAVE1) 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 with eight supplementary modules (XREF1, XREF2, XREF3, XREF4, XREF5, XREF6, XREFA, and XREFB) 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
BAKUP	- Disc-to-disc on-line copy utility for POINT 4-supported disc controllers/drivers
DISCUTILITY	- Disc utility for POINT 4-supplied disc subsystems
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 2.5).

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 4 310 Mux (with master terminal mode) at a rate other than 9600 baud, then the BPCON at word 375 of BTUP and the master port's Port Definition Table in both REX and $MMUX must be changed. BPCON must contain 5036x 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

To change BPCON, use a terminal set at 9600 baud and proceed as follows:

1. IPL the system.

2. Log on to the Manager account.

3. Enter DSP and issue the following commands:

```
G2          (get BTUP)
375:5036x   (enter new value in BPCON; the default is 0)
X           (exit)
```

4. Change the baud rate in the master port's Port Definition Table in REX (the pointer to PDT is at location 200 in REX) and in $MMUX.

5. SHUTDOWN the system.

6. Change the Mux hardware default baud rate.*

7. Set the master terminal to the new baud rate.

---

*Baud rates on a POINT 4 MARK 3 System are hardware jumperable. For information on baud rate selection procedures, refer to the POINT 4 MARK 2/3 Peripherals Interface Manual or the POINT 4 MARK 5/9 Computer Reference Manual as appropriate.

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 and APL on a MARK 5/8/9 front panel. On a MARK 2/3, press RESET.
- 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).

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.

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 in CTU mode.

2.2.1.1 Re-Entry to DBUG

To re-enter DBUG manually, APL 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 Memory Locations for DBUG

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.

2.2.1.5 Changing Baud Rate for DBUG

To use DBUG with a POINT 4 310 Mux (with master terminal mode) at a baud rate other than its default rate, enter the desired PCB and PCW 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 310 Mux, and PCW is the port control word which specifies the desired baud rate and parity mode (e.g., 50057 for 9600 baud, even parity). To disable the 310 Mux setup, put a 0 in word 2 of DBUG. The first eight words of DBUG are described in Table 2-2.

TABLE 2-2. FIRST EIGHT WORDS IN DBUG

Word	Description
0	Main entry point.
1	Re-entry point - Retains the value of previously saved accumulators, carry, and breakpoint status.
2	POINT 4 310 Mux setup - Contains Port Control Block (PCB) address; set to zero for non-310 Mux.
3	Port Control Word (PCW) for POINT 4 310 Mux setup.
4	PCB address for CTU port.
5	PCW word for CTU port.
6	Default RETURN delay (see Y command in Table 2-3).
7	Pointer to auxiliary bulk-memory driver (see Section 2.2.1.6).

2.2.1.6 DBUG and System Drivers

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) or other bulk-memory driver having BZUD-compatible calling sequences. DBUG may be interfaced to BZUP or another compatible driver by entering a pointer to the driver's Read-entry point in word 7 of DBUG (see Table 2-2). The BZUD-compatible drivers for systems using an LCM or MARK 9 mapped memory are $LCM and $SYSMAP respectively. Refer to the DEFS listing in Appendix B for the current location of drivers in the REX file.

If the G or W command is used and BZUD is not 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 or BZUP determine which physical unit is to be used.

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 2-3. 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.

TABLE 2-3. 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 2-3. DBUG FUNCTIONS (Cont)

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>B0,n</td><td>Remove breakpoint n (0 if n is omitted), restoring the instruction at that location. Note that a breakpoint cannot be put at location zero.</td></tr>
<tr><td>B</td><td>Remove all breakpoints that have been established.</td></tr>
<tr><td>Ca</td><td>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.</td></tr>
<tr><td>Cx,y
(x≤5)</td><td>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.</td></tr>
<tr><td>Dx,a
(a≤3)</td><td>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.</td></tr>
<tr><td>Dx,n
(n>3)</td><td>Dump memory in octal, beginning at location x, and displaying n words per line with the address of the first word on each line.</td></tr>
</table>

(Table continues on next page)

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+1) 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 2-3. 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 or other bulk-memory device. Block number (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 Section 2.1.2.6). 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 2-3. 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 2-3. 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 2-3. 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 2-3. 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 or other bulk-memory device to disc. Locations y through y+377 are written to block number (real disc address) x. Wx will write page zero on the disc (see Section 2.1.2.6). If disc error is detected, a bell, a backslash (\), and the disc controller status word are output.

TABLE 2-3. 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 SUBL instruction in order to detect a change (e.g., 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 2-3. 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 2-4 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 2-5 is a quick-reference guide to the commands used for data transfer from a source to a destination.

TABLE 2-4. 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 = 999 for 1000 blocks.

TABLE 2-4. 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 2-5. 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

2.2.4 CHANGING THE PORT CONTROL BLOCK IN DBUG

If the PSIZ was increased on a system with 32K-word memory by moving the first port control block (PCB) location in $MMUX, and if DBUG will be used for debugging, the PCB in DBUG must also be moved accordingly. The PCB in DBUG is at word address LDBUG plus two and the port control word (PCW) is at LDBUG plus three.

For example, assume that DBUG resides at memory address 73000 (octal) and PSIZ was increased by 2000 octal as described in Section 5.12.2.2; then the following change must be made in DBUG:

- Using DSP, change the location of DBUG's PCB by adding 2000 (octal) to the contents of LDBUG+2 (i.e., the contents at location 73002).

- If the PCW requires modification (e.g., baud rate), make the desired changes at LDBUG+3.

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 data 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.

F$filename may be used to select an enabled driver. The Dx' command may then be used to display the memory-resident copy of that driver including the current value of the local temporary cells.

where
- x - the address in the disc file; corresponds to the Assembly language listing
- ' - (apostrophe) selects the memory-resident copy of the driver instead of the disc file

The memory-resident copy of the driver does not reside at address x but address translation is handled by DSP automatically.

Similarly, FDISCSUBS allows x' to display the memory-resident copy of a memory-resident discsub.

When a symbolic instruction such as a user defined function is entered via an insert (x:v or E) or an append (Ax) command, the system translates it into Assembly language instruction format. For example, the user enters

SEQ 0,1

When the L command is used to check the entry, DSP displays

SUB# 0,1,SZR

Commands may be entered in lower case letters with the exception of N in the LxN command which must be upper case.

For a description of the commands used in DSP see Table 2-6.

TABLE 2-6. 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 (i.e., user-defined function) 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.

Command	Description
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.
Bxcond'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.
Ccommand	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>.

TABLE 2-6. DSP FUNCTIONS (Cont)

Command	Description
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.
Ex	Enter octal or symbolic instruction (i.e., user-defined instruction) 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.
Ffilename	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.

TABLE 2-6. DSP FUNCTIONS (Cont)

Command	Description
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.
Gu/x or Gx	Select, on logical unit u (where u is in octal), the disc block at real disc address x to be examined and/or modified. In this mode, only cells 0 through 377 (octal) will be accepted. The simple form Gx asumes 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 as a <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>.

TABLE 2-6. DSP FUNCTIONS (Cont)

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>LxN</td><td>Same as Lx except only the Assembly language instructions are printed.</td></tr>
<tr><td>Mx,y,z</td><td>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.</td></tr>
<tr><td>Nx,y,z</td><td>Search location x through y inclusively for a location <u>not equal</u> to the octal constant z. If found, displays the location and its content in octal and symbolic form.</td></tr>
<tr><td>Nx,y,z,m</td><td>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

applies the mask, 101777, to the contents of locations 400 through 1120 and checks for any value not equal to octal 53.</td></tr>
<tr><td>Ox</td><td>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.</td></tr>
<tr><td>Px,y</td><td>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.</td></tr>
</table>

TABLE 2-6. DSP FUNCTIONS (Cont)

Command	Description
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.
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.

TABLE 5-2. INFO TABLE

Location (octal)	Label	Description
600	SDAT	System creation date (hours after BASEYEAR). DO NOT CHANGE!
601	SPED	Average CPU speed in instructions per millisecond: Computer — Speed (octal) POINT 4 MARK 9 — 2500 POINT 4 MARK 5 — 2000 POINT 4 MARK 3 — 1200 NOVA — 302 NOVA 1200 or D-116 — 653 NOVA 2 or D-116H — 770 NOVA 800 — 1325 NOVA 3 — 770 SUPER NOVA — 1255 SUPER NOVA SC — 1762
602	MILU	Maximum number of installed logical units - The total number of physical disc partitions defined in the Disc Driver Table. See Section 5.4.1.
603	NDCH	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.
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.

(Table continues on next page)

TABLE 5-2. INFO TABLE (Cont)

Location (octal)	Label	Description
605	TNAP	Total number of active ports - If the value in TNAP represents less than the total number of interactive ports contained in all driver's attributes tables, SIR increases 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 increase the number of interactive ports automatically.
606	SPCF	Special conditions flags - These are flags which control certain system functions and options:
		Bit 13 - Temporary Dirty Page Flag (TDPF) writes to disc at end of a user's time slice (see Section 5.14.3).
		Bit 14 - Suppress Error Message Flag (SEMF). 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.)
		All other bits are reserved.
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!

TABLE 5-2. INFO TABLE (Cont)

Location (octal)	Label	Description
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	UDSB	Number of user discsubs - The minimum value is one greater than the largest subroutine number in the DISCSUBS.USER file.
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.
616	SDSB	Number of System discsubs - The minimum value is one greater than the largest subroutine number in the DISCSUBS file.

TABLE 5-2. INFO TABLE (Cont)

Location (octal)	Label	Description
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		Default (application) logical unit - Used by the system when searching for a specified program. If invoked from SCOPE, the system searches for the program in the following sequence: ● LU/0 ● assigned LU ● default LU If invoked from CHAIN, the system searches for the program in the following sequence: ● the default LU ● assigned LU ● LU/0 Value of 177777 indicates no default LU.
621		Reserved.
622	SZLNK	Pseudo-device linkage table size - A pseudo-device has no device code (e.g., $CTUS).
623 to 631		Reserved.
632 to 777		Reserved.

TABLE 2-6. DSP FUNCTIONS (Cont)

Command	Description
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.
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.

TABLE 2-6. DSP FUNCTIONS (Cont)

Command	Description
Xx,y	Compute and display a "rotating" checksum over memory locations x through y. The checksum is produced by an SUBL instruction in order to detect a change (e.g., if two words in memory are swapped). Useful for testing if a change has occurred anywhere in a section of memory or on disc.
Xx',y	Checksum the memory-resident copy of a discsub or driver as selected by a Ffilename command.
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.
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 DISC UTILITIES

IRIS supports four disc utilities which offer a variety of functions:

- BAKUP - on-line disc-to-disc
- DISCUTILITY - copies disc-to-disc and disc-to/from-other media, etc.
- DDCOPY - copies disc-to-disc only
- BLOCKCOPY - copies selected blocks from one location to another or to another disc (see the IRIS Operations Manual for BLOCKCOPY procedures).

BAKUP may be used on POINT 4-supported disc controllers/drivers. DISCUTILITY is available for systems using the POINT 4-supplied disc subsystems. DDCOPY is available for most systems which do not have a POINT 4 controller.

2.4.1 BAKUP

BAKUP is an on-line, user-configurable, disc-to-disc copy utility that is recommended for performing backups. Copying can be done between any two surfaces (fixed or removable) on one or more drives connected to as many as four different controllers. Any logical unit (LU) may be copied to any other LU provided both have been defined in a parameter file called BAKUPPARAM.

The number of blocks that are copied is controlled by the smaller of the two logical units, be it source or destination logical unit.

Two optional procedures are available:

- Read after write verify on all blocks copied.
- DMAP reconstruction. This feature permits the copying of different sized logical units. It should not be used if the LUs specified are not IRIS LUs.

POINT 4 supplies a BASIC program called BAKUP that may be modified to meet the requirements of a particular installation. This program, or one like it, must be used to run the actual disc-to-disc copier (BAKUPMAIN). The system must not be SHUTDOWN but all timesharing operations are suspended while the copying procedures are performed. When the BAKUPMAIN program has completed the specified disc-to-disc copy procedure, timesharing is automatically restored.

Refer to the IRIS Operations Manual for information on using BAKUP.

2.4.2 DISCUTILITY

DISCUTILITY is a stand-alone utility package for POINT 4-supplied disc subsystems. 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.

2.4.3 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 as a halt occurring under IRIS but the cause is different. Refer to the IRIS R8 Operations Manual for a list of IRIS system halts.

   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 must 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.

2.5 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
LOADING SOFTWARE

This section discusses the recommended methods for loading software from disc, diskette, and streamer tape. The procedure for loading software from cassette tape is described in Section 4. A tech memo is supplied with the paper tapes for installations that require a paper tape sysgen.

3.1 PREPARATIONS FOR BOOTING THE SYSTEM

Before the IRIS Operating system is loaded, hardware diagnostics should be run. 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.

3.2 LOADING SOFTWARE AND CONFIGURING THE SYSTEM

The software supplied by POINT 4 consists of a standard IRIS Operating System including stand-alone utility programs, and any optional application packages that were ordered. This system should not be loaded or configured directly from the media (disc, diskette, or streamer tape) supplied by POINT 4. The software should first be copied to a scratch disc pack if supplied on disc or streamer tape. It should be copied to scratch diskettes if supplied on diskette.

The procedure for loading an IRIS Operating System and making it operational is as follows:

1. Copy the POINT 4-supplied template to a scratch disc. For a POINT 4 MARK 3 system, scratch diskettes may be used.

    a. If supplied on disc, refer to Section 3.2.1.

    b. If supplied on diskettes, refer to Section 3.2.2.

    c. If supplied on streamer tape, refer to Section 3.2.3.

    d. If supplied on cassette tapes, refer to Section 4.

2. If the system has more than an 8-port POINT 4 Mux, bring the system up into a minimum IPL (refer to the IRIS Operations Manual) and establish the correct number of ports using DSP (see Section 5.8.3).

    NOTE

    Be sure to establish the correct number of ports at ATRIB-1 (see Section 5.8.3) and define the correct number of ports in the Port Definition Table (PDT) (see Section 5.8.1.4).

3. Shutdown the system.

4. IPL the system into a full configuration using the cold startup procedure described in the IRIS Operations Manual.

5. Configure the system.

    a. Enable the appropriate drivers (see Section 3.3).

    b. Use the POINT 4-supplied configurator (SETUP) for the following:

        - The system INFO table (see Section 6.2.2)

        - The port definition table in the appropriate drivers (see Section 6.2.3)

        - The disc driver table (see Section 6.2.5)

- If necessary, change the preset memory-resident discsubs table (see Section 6.2.4)

If DSP is used instead of SETUP, configuration includes:

- Adjusting partition requirements (see Section 5.12)

- Setting up ports (see Section 5.8.1.4)

- Defining the disc driver table (see Section 5.4)

- If necessary, modifying the preset memory-resident discsubs table (see Section 5.3)

c. Enable the appropriate terminal translation modules (see Section 5.9)

d. Use ACCOUNTUTILITY to set up user accounts (see Section 5.7).

e. Use GUIDE.LPT to set up line printers (see Section 5.8.7).

f. Use DSP to make certain processors accessible from selected accounts (see Section 5.11).

6. Shutdown and IPL the configured system.

7. Install LU/5.

8. Run LIBR @ to obtain a list of system components on LU/0 and LU/5. Compare the listings with the checklists given in Appendix A. If these do not correspond, call Customer Support.

9. Test the system.

10. Make a copy of the configured system disc pack or diskette. Use the copy to run the system and keep the disc pack or diskette used for the configuration process as a back-up.

NOTE

BAKUP is an on-line disc-to-disc copy program that may be used for performing back-ups for disc packs. Refer to the IRIS Operations Manual for information on BAKUP procedures.

3.2.1 LOADING SOFTWARE FROM DISC

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. Copy the POINT 4-supplied disc pack onto a scratch disc pack which has been formatted 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 not be IPLed or INSTALLed so that the original version of the software is always available in its original condition and may be recopied for future reconfigurations.

It may be necessary to load a stand-alone disc-to-disc copy program into memory without doing an IPL. POINT 4 recommends that a cassette or streamer tape unit be available for that purpose.

3.2.2 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 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.

3.2.2.1 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.

2. Press RESET to load MANIP into memory.

3. Insert the diskette into floppy disc drive 0.

4. 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.

5. Use the F (format) command to format a minimum of four scratch diskettes.

6. Copy the POINT 4-supplied diskettes to the formatted scratch diskettes using the copy option of the DISCUTILITY program.

7. Store POINT 4's diskettes in a safe place.

8. Configure the system using the scratch diskettes.

3.2.2.2 Writing from Memory to Diskette on a MARK 3

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 2.4.2) may be used. If you have a stand-alone program such as DISCUTILITY on an IRIS logical unit and wish to create a diskette of this program so that it can be booted directly into memory from MANIP, FLBOOT may be used as follows:

1. Shutdown the system to the program to be written out to diskette with FLBOOT as the second program. Using DISCUTILITY as an example, the command format is

   SHUTDOWN <CTRL-E>key<CTRL-E> DISCUTILITY,FLBOOT

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

2. Insert a formatted scratch diskette into drive 0.

3. Jump to location 70000 by entering

   J70000

   A loader is written as block 1 onto the diskette followed by blocks containing the selected program from memory locations 0-67777. When the transfer is completed, control is returned to MANIP.

To make another copy of the program, repeat the procedure starting at step 2.

3.2.3 SOFTWARE SUPPLIED ON STREAMER TAPE

Software supplied on streamer tape for a POINT 4 Computer System may be one of the following:

- Stand-alone programs such as LOTUS DISCUTILITY which can be booted into memory using MANIP

- One or more logical units (could include LU/0) which may be RESTOREd using the appropriate utility program (e.g., LOTUS DISCUTILITY or MARK 3 DISCUTILITY)

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.

3.3 ENABLING DRIVERS

An IRIS Operating System is delivered with all drivers supported by IRIS as standard components. Depending on the system configuration, some drivers must be enabled and others disabled. A driver is enabled by adding a $-sign to the filename (e.g., changing DGMX to $DGMX). When the system is IPLed, SIR makes all $files memory resident. For efficient use of memory, only those drivers should be enabled that are required for a particular installation.

Table 3-1 is an annotated list of drivers that are supported under IRIS. Refer to Section 5.8 for information on tables internal to the driver files. Refer to Section 6 for information on configuring the disc driver table in the CONFIG file.

TABLE 3-1. ANNOTATED LIST OF DRIVERS

Name	File Type	Remarks
COMA	36	Copy as $COM for an Airland-type protocol converter.
COMD	36	Copy as $COM for a Datalynx protocol converter.
CTR	77036	For use by POINT 4 with the MONITOR program (a diagnostic tool).
CTUS	77001	Enable for a cassette tape unit.
$DEC	77001	Disable for a POINT 4 MARK 9 CPU and a MARK 5 with extended instruction set.[1]
DGMX	77001	Enable for a Data General 4060-type multiplexer only.
EIS	77001	Enable for a POINT 4 MARK 5/9 Computer System with extended instruction set (supersedes MK8).
FOREIGN	77036	Enable for reading/writing non-IRIS-generated discs and diskettes.
LCM	77001	Enable for a POINT 4 LOTUS Cache Memory Subsystem.
LPTD	36	Copy as $LPT for a line printer on a Data General 4060-type Mux (see Section 5.8.7).

[1]All IRIS systems shipped on disc, diskette, or cassette tape have $DEC enabled. If the CPU is a MARK 3 or a MARK 5 without extended instruction set, no further action is required. For a MARK 9 CPU or a MARK 5 with extended instruction set, use CHANGE to disable $DEC and to enable EIS (see also Section 5.15).

TABLE 3-1. ANNOTATED LIST OF DRIVERS (Cont)

Name	File Type	Remarks
LPTM	36	Copy as $LPTn for a POINT 4 310 or MARK 3 Mux (see Section 5.8.7).
LPTP	36	Copy as $LPT for a line printer on a parallel interface, device code 17 (see Section 5.8.7).
$MMUX	77001	Used for a POINT 4 Mux only.
MTA0	36	Enable for a magnetic tape or cassette tape unit (see also Section 5.8.10).
MTAS	77001	Enable for a magnetic tape unit (see also Section 5.8.10).
PHA	77001	Enable for phantom ports.
PTM	36	Enable for TTY paper tape punch/reader.
PTP	36	Enable for a paper tape punch.
PTR	36	Enable for a high-speed paper tape reader.
RTC	77001	Enable for a Data General 4060-type multiplexer only.
SYSMAP	77001	Enable for a MARK 9 CPU.
TERMS	77001	Enable for terminal translation modules (see Section 5.9).
TTY	77001	Enable for an additional terminal with its own computer interface.

3.4 BASIC UTILITY PROGRAMS

BASIC utility programs on logical unit zero (LU/0) and the applications logical unit (LU/5) are shipped as SAVEd BASIC programs and are ready for use. Refer to Appendix A (Table A-1) for a complete list of IRIS components residing on LU/0. Appendix A (Table A-2) provides a listing of BASIC utility programs residing on LU/5.

Section 4
CTUTILITY
Cassette Tape Utility Program

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

A disc image copy of an IRIS system is delivered on cassette tape. CTUTILITY provides an easy method for loading the system and 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 Sections 5 and 6.

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 designated as a universal template. However, each template is designed for the particular CPU (MARK 3 or MARK 5/8) and a particular version of IRIS. An attempt to load software from the wrong template causes an appropriate error message to be displayed (see Section 4.7.3). A list of other error messages is provided in Section 4.7.4.

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.
- CTUTILITY requires that ports 0 and 1 be set to 9600 baud.
- For a POINT 4 310 Mux, master terminal mode must be enabled.

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.

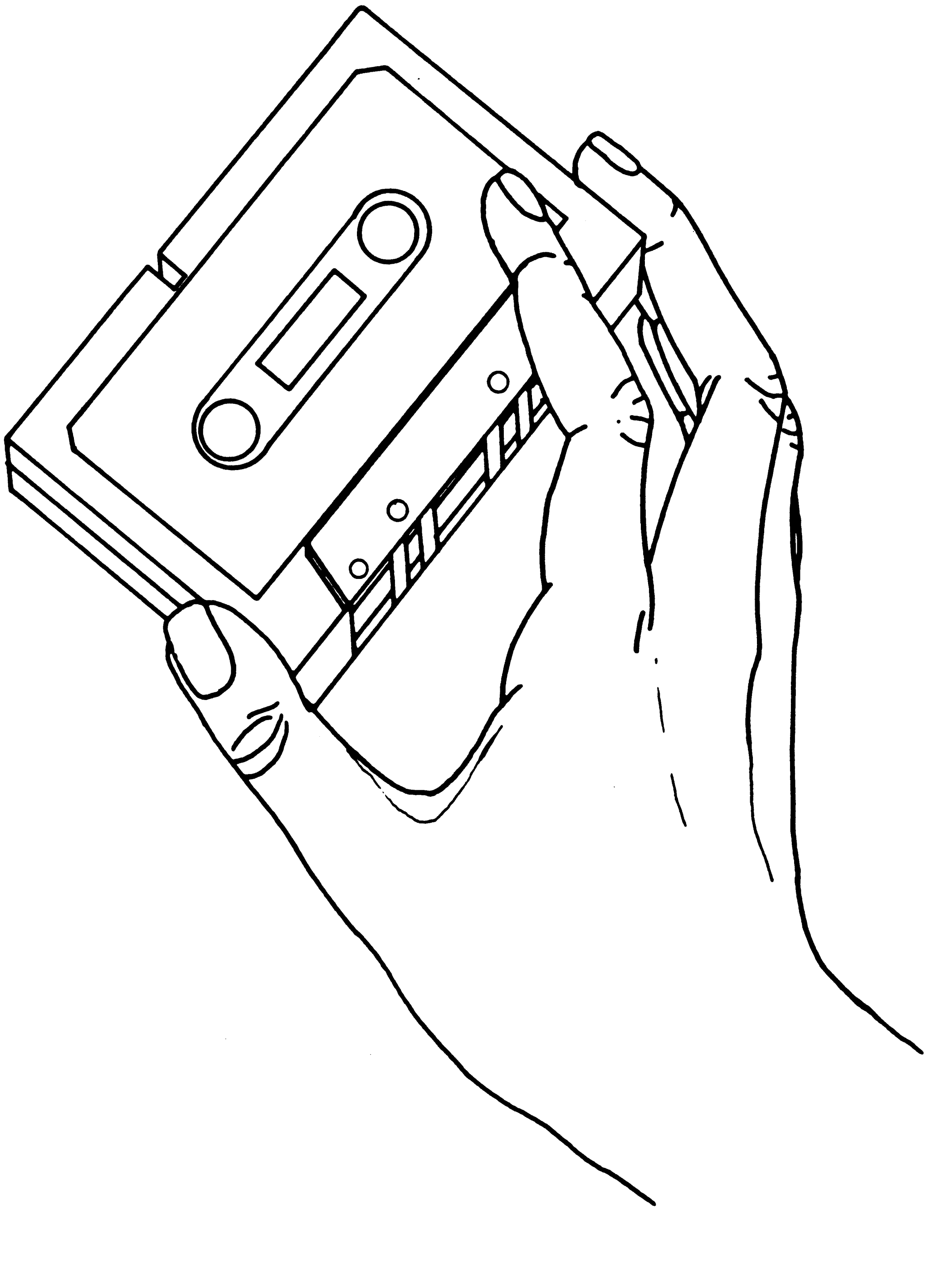


Figure 4-1. Mounting the Cassette

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 PREPARATION FOR LOADING SOFTWARE FROM CASSETTE TAPE

Before loading the software from the POINT 4-supplied cassette tapes, 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 3.2), and that hardware diagnostics have been run as discussed in Section 3.1.

The following are required:

- Diagnostic programs
- Tape-set including: CTUTILITY, appropriate template, and utilities
- One disc pack to set aside as a master copy
- Appropriate media for backup
- Pico-N
- CPU memory (must be 32K or more)

4.3.1 PROCEDURE FOR LOADING THE SYSTEM DISC

The template for the system disc (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. Load the IRIS Operating System 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 Sections 5 and 6 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

This section provides five procedures for loading 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. IPL the system using the cold startup procedure described in the IRIS Operations Manual.

3. Start CPU execution and load DBUG at location 73000. At the system command prompt (#), enter

   <u>SHUTDOWN <CTRL-E>key<CTRL-E> @73000 X73000</u>

4. Rewind the tape by entering

   <u><CTRL-Z><RETURN></u>

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

5. Select Track 0 by entering

   <u><CTRL-T>0</u>

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. Move DBUG to the location listed on the label of the tape cartridge using the format

    M(CTU DBUG addr),(CTU DBUG addr+3000),nnnnn

    where
    nnnnn - address of DBUG listed on tape cartridge label

    For example

    M30000,33000,73000

    Enter DBUG as follows:

    J73000

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. Select 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 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 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

(Continues on next page)

(Contd)

Location	Contents
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. CTUTILITY displays an error message if the tape is not appropriate for a particular configuration, or if it is not an IRIS tape (see Section 4.7.3).

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.

CTUTILITY can not be used to merge two logical units directly. The recommended procedure for merging two logical units is described in Section 4.8.5.

If an error is made when entering a command or parameter, <CTRL-X> may be used to abort the entry.

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 load 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 the procedure for loading LU/0 from tape to disc. Some values shown here are dependent on the configuration.

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 loading the software 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 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 specification sheet for the disc controller and drive in the IRIS R8 Peripherals Handbook. The entry number is printed in the upper right corner of the sheet, as shown in Figure 4-2. Enter this number and press <RETURN>.

The system displays

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

where SOV is the System Overlay Disc Driver and BZUD is the Block Zero Utility Driver. If the wrong entry number is entered, the system displays

```
DISC DRIVER INDEX IN CONFIG IS NOT CORRECT!
```

If the entry number is correct, the system requests the following parameters:

```
PLEASE ENTER THE DEVICE CODE FROM THE PERIPHERALS HANDBOOK SPEC. SHEET:

FROM THE DISC SPECIFICATION SHEET, PLEASE ENTER
NTRS=
```

After the number of IRIS tracks (NTRS) has been entered, the system displays

```
PLEASE WAIT...
```

CTUTILITY then prompts for PHYU (physical unit select constant) and FCYL (first physical cylinder).

```
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 in the IRIS R8 Peripherals Handbook.

R8 DISC SPECIFICATION

CONTROLLER: POINT 4 LOTUS 700[1]

ENTRY NO.: 38

DISC ID: P480MB

DATE: 08-01-83

DRIVE	Total Cyls On Disc	Max Cyls Other LUs
Ampex DM-940 (40MB)	626	631[3]
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 62356

BZUD ADDR 62004

LRC 240

NPTC 5

DFLG 40500

NTRS 1220

PHYU D + 20024
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.
[3]LU MAY NOT exceed total cylinders on disc.

Figure 4-2. Sample Disc Specification Sheet

After the value of PHYU is entered, the system requests

FCYL=

After FCYL 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!

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>.

The system then prompts for the following parameters, one at a time:

FROM THE DISC SPECIFICATION SHEET, PLEASE ENTER
DFLG=

FROM THE DISC SPECIFICATION SHEET, PLEASE ENTER
NPTC=

The following prompt occurs only if the specified disc has a limited capacity (e.g., a diskette).

FROM THE DISC SPECIFICATION SHEET, PLEASE ENTER
NO. CYLS IN LU/0:

CTUTILITY then displays

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)

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)

where xxxxxxx is the number of cylinders required for the destination logical unit (i.e., the number of cylinders on disc to be overwritten).

If the scratch logical unit does not have enough cylinders, enter N. The prompt is then repeated and the user may mount another disc pack containing a logical unit with the required number of cylinders.

If the scratch logical unit contains enough cylinders, enter Y.

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

While the tape is read, the system displays the following progress messages:

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

If a read/write or hardware 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 LU/0 was loaded, IPL into a minimum configuration and customize the system as described in Section 5. Then 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 logical unit 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, shutdown and 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 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 specification sheet for the disc controller 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 <u><RETURN></u>.

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 requests the following parameters

```
PLEASE ENTER THE DEVICE CODE FROM THE PERIPHERALS HANDBOOK SPEC. SHEET:

FROM THE DISC SPECIFICATION SHEET, PLEASE ENTER
NTRS=
```

where NTRS is the number of IRIS tracks. After NTRS has been entered, the system displays

```
PLEASE WAIT...
```

CTUTILITY then requests the values for PHYU (physical select unit constant) and FCYL (first physical cylinder).

```
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 999.

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, read, write or hardware errors are 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.
```

Warning messages are displayed if the cassette tape is not an IRIS tape or if the template is not appropriate for a particular CPU (see Section 4.7.3).

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.

4.7.3 SYSTEM LOADING ERROR EXAMPLES

If an attempt is made to load the IRIS Operating System from a template not appropriate to a particular CPU, the following warning is displayed:

```
WARNING:  YOU ARE USING AN LU/0 TEMPLATE ON THE WRONG TYPE OF CPU
THE RESULTANT LU CAN BE INSTALLED BUT NOT IPLED
```

If the revision of IRIS on the template is not compatible with the version of CTUTILITY, the following message is displayed:

```
CURRENT TAPES ARE NOT THE CORRECT LU/0 TEMPLATE FOR YOUR CPU
AND THIS VERSION OF CTUTILITY.
IF YOU PROCEED TO CREATE AN LU/0 DISC,
IT MAY BE INSTALLED AS A NON-LU/0
BUT IT WILL NOT BE ABLE TO IPL AND RUN IRIS.
```

If the wrong tape (e.g., the utility tape) is being used to load IRIS, the following message is displayed:

```
NOT AN ACCEPTABLE IRIS LU TAPE...
```

4.7.4 ERROR STATUS CODES

If CTUTILITY encounters an error in accessing the CTU, the program displays an error message (see Section 4.7.2) which includes a status code. Each error status code is described in Table 4-1.

TABLE 4-1. ERROR STATUS CODES

Status Code	Description
1	Reserved
2	Incorrect echo of command - Character echoed by the CTU was not the character sent.*
3	Reserved
4	Incomplete read acknowledgment - CTU returned BELL RETURN but no LINEFEED after read operation.*
5	Read error - Number of characters received from CTU was greater than or less than expected.*
6	Incorrect echo - In writing data to CTU, character echoed by CTU was not the character sent.*
7	Incorrect acknowledgment - CTU did not return the expected BELL RETURN LINEFEED acknowledgment after a write operation.*
10	Incomplete write acknowledgment - CTU returned BELL RETURN but no LINEFEED after write operation.*
77	Syntax error in command string.
*Error detected by the CTUTILITY software - Indicates that the CTU electronics are not responding as expected (see also **).	

TABLE 4-1. ERROR STATUS CODES (Cont)

Status Code	Description
110	Tape header block not correct - May be due to bad tape initialization or tape media problem.
115	Tape motion failure - Usually occurs as a result of a jam, mechanical malfunction, or as a result of incorrect tape handling. If the tape is not seated correctly, the CTU does not know the tape location and thus runs into the stops.**
120	Write-protect error - A write operation was attempted on a write-protected tape.
122	Read Error - Not correctable. An excessive number of read errors usually indicates noise interference, faulty system ground, a defective tape, or a CTU hardware problem.**

**Check the CTU and tape as follows:

- Turn the CTU off and on.
- Reseat the tape (make sure the tape is positioned with the correct side up).
- Check that all cables are securely connected.
- Retry the operation.

If the operation fails again, the tape media may be bad. Try a tape that has been successfully run before. If this tape works, then the original tape probably is bad and should be replaced. If the good tape fails, the CTU may be the problem. Try a different CTU or call POINT 4 Hardware Support.

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 4-2 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 4-2. 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 X73000

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

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

4.8.3.2 Preparing to Load Utilities on POINT 4 MARK 5/9 CPUs With CTU PROMs Without Running IRIS

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 <u>without</u> 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 ID	STARTING ADDRESS	PROGRAM 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. Example of a Cassette Tape Directory

4.8.5 MERGING TWO LOGICAL UNITS

The tape-to-disc procedure described in Section 4.6.1 can not be used to merge two logical units via a scratch logical unit because the destination logical unit parameters requested for PHYU and FCYL reference one logical unit only. This causes the pointers to the first logical unit to be lost; only the second logical unit is found when an INSTALL is done.

The following is the recommended procedure for merging two logical units:

1. Transfer each logical unit to a separate scratch logical unit as described in Section 4.6.1.

2. Use U.COPY to merge the files from each scratch logical unit into a third logical unit.

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>999<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.

Configuration procedures for the POINT 4 LOTUS Cache Memory (LCM) are described in the R8 LCM Installation Tech Memo.

5.1 AIDS FOR CONFIGURATION

POINT 4 provides two interactive programs to assist in the configuration of a system: GUIDE and SETUP. POINT 4 also provides an IRIS 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.

5.1.2 SETUP

SETUP is a system configurator which may be used to set up and modify the following:

- System INFO Table
- Port Definition Table (DFT)
- Disc Driver Table
- Memory-resident Discsubs Table

The program provides access to two parameter files:

SU.ENTRIES - A formatted file containing listings of disc controller information based on the R8 Peripherals Handbook.

SU.DSUBS - A text file containing a list of system discsubs.

SETUP operates on a user-named control file. This accommodates the setup of multiple configurations at a given installation. Only when the update function is executed is the particular configuration applied to the installation. Refer to Section 6 for information on using SETUP.

NOTE

SU.ENTRIES must be updated by the user when changed pages are issued for the R8 Peripherals Handbook.

5.1.3 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 SETUP utility (see Section 6). 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. Refer to Appendix D for a listing of the first four blocks of the CONFIG file.

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.
1400-2777	Disc Driver Table. See Section 5.4.1.
3000-13377	Reserved.
13400-13577	Specific IPL sequences.
13600-13777	Reserved.
14000-15777	BZUD and R/W entry addresses of the disc drivers.
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 following:

Location (octal)	Label	Description
400	PSIZ	Partition Size. The size of each memory partition.
401	NPART	Number of memory-resident partitions. For a MARK 9, include partitions in mapped memory.
402	MTYPE	Memory type: 0 = standard MARK 3 or 5 memory 1 = MARK 9 or Nova 3-type mapped memory

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 IPL 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.

5.3 DISCSUBS

Discsubs are subroutines which normally reside on disc and are divided into two general classes: IRIS system discsubs and user discsubs.

5.3.1 IRIS DISCSUBS

IRIS system discsubs are divided into standard and optional discsubs. Standard discsubs are an intrinsic part of the IRIS Operating System while optional discsubs support POINT 4-supplied optional hardware and software packages. Both types of system discsubs reside in the DISCSUBS file.

Some discsubs are more important because they are used more frequently than others. These should be made memory resident to ensure optimum system performance. Some discsubs require one disc block, others are extended and require two disc blocks; some may include another discsub when memory resident.

Each discsub has an identifying keyword consisting of a number and special flags as defined in DEFS. Numbers of discsubs that are to be memory resident are entered into the memory-resident 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 system discsubs that are most important and most frequently used by the IRIS Operating System (see Section 5.3.3). Depending on the memory available, this list may be expanded and/or modified by the user. A complete list of IRIS system discsubs is also given in Appendix B.

5.3.2 USER DISCSUBS

User discsubs are supplied by POINT 4's OEMs as part of application packages developed by them. The user discsubs reside in the DISCSUBS.USER file.

5.3.2.1 User Discsub Keywords

The discsub keywords in the DISCSUBS.USER file are totally independent of the system discsub keywords used by POINT 4. However, the discsub keyword must include the special flags for discsubs as defined in DEFS (e.g., the U-bit (U=2000) for a user discsub; the X-bit (X=40000) for an extended discsub).

5.3.2.2 The DISCSUBS.USER File

The rules for creating a system discsub apply to user discsubs (see the IRIS System Programmers Manual).

DISCSUBS.USER is a contiguous file. If it does not exist on your system, it may be created from the manager account by using the following procedure:

1. Create a contiguous file using the FORMAT command

   ```
   #FORMAT [n:25610/DISCSUBS.USER
   ```

   where n is the number of blocks desired.

2. Use DSP to set word 175 (CORA) in the header block of the DISCSUBS.USER file to 400 as shown in the following example:

   ```
   #DSP<CTRL-E>key<CTRL-E>
   F DISCSUBS.USER
   H
   175:400
   ```

   where key is the password.

   For each block in the DISCSUBS.USER file which does not contain a user-supplied discsub, enter 177400 at the beginning of the block as shown in the following example:

   ```
   F.
    400:177400
   1000:177400
   1400:177400
   2000:177400
       .
       .
       .
       .
       .
   ```

3. Update UDSB in the INFO table. As user discsubs are added, UDSB (location 612) in the CONFIG file's INFO table must be adjusted. The minimum value in UDSB is one greater than the largest number assigned to a subroutine in the DISCSUBS.USER file.

5.3.3 MAKING DISCSUBS MEMORY RESIDENT

In order to make a discsub memory resident, it is necessary to enter the number of the discsub into the memory-resident DISCSUB table at location 1000 in the CONFIG file. Discsubs do not have to be entered in any particular order but the table must terminate with 177777. This may be done via the SETUP program (see Section 6) or by the use of DSP.

Some discsubs are included within another discsub. Appendix B gives a complete list of IRIS system 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 and must not be specified separately.

For example: OPEN is to be moved to memory. Appendix B-12 shows its number is 40022 (i.e., the X-bit (X=40000) in the first word of the discsub is set indicating an extended subroutine). Enter 22, since only the lower three digits of an IRIS system discsub number are entered into CONFIG's DISCSUB Table. Because Discsub 22 includes Discsubs 23, 24, and 25, they are included in the transfer.

5.3.3.1 Making an IRIS Discsub Memory Resident

POINT 4 recommends that SETUP (see Section 6.2.4) be used to modify the preset DISCSUB table. For those who wish to use DSP, the last two digits of the discsub number must be entered into the memory-resident DISCSUB table. The sequence of commands is as follows (user response is underlined):

```
#DSP<CTRL-E>key<CTRL-E>CONFIG
```

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

```
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.

5.3.3.2 Making a User-Supplied Discsub Memory Resident

To make a user (OEM-supplied) discsub memory resident, the U-bit must be included in the number entered into the memory-resident DISCSUB table (i.e., 2000 is added to the discsub's number). For example, to make user discsub number 20 memory-resident, enter 2020 into the DISCSUB table.

5.3.4 STANDARD SYSTEM DISCSUBS

Standard system discsubs are the most frequently used subroutines that are required by the IRIS Operating System for its general functions. POINT 4 presets them in the DISCSUB Table starting at location 1000 (octal) in the CONFIG file. Table 5-3 shows a list of those preset, standard discsubs arranged in order of priority, giving location, discsub number, and name. They may appear in the CONFIG file in any order.

If the system manager desires to remove discsubs from the CONFIG file list in order to free up memory, the discsubs with the lowest priority (i.e., at the bottom of Table 5-3) should be removed first. Such action does not prevent any system functions, but it does have an impact on system performance since those routines are now disc resident.

TABLE 5-3. PRESET STANDARD SYSTEM DISCSUB LIST

Priority	CONFIG Location (octal)	Discsub Number	Name
1	1000	67	AFSET
2	1001	100	LINKP
3	1002	101	LOADP
4	1003	3	FFILE
5	1004	15	ACNTLOOKUP
6	1005	22	OPEN&377 (2 blocks)
7	1006	26	CLOSE
8	1007	30	GETRR&377 (2 blocks)
9	1010	33	READITEM
10	1011	1	ALLOCATE
11	1012	40	CHARGE
12	1013	36	READCONTIG
13	1014	61	SEARCH&377 (2 blocks)
14	1015	62	SHUFFLE
15	1016	63	DEKEY
16	1017	27	CLEAR
17	1020	46	SPECIAL
18	1021	57	SIGPAUSE
19	1022	41	SYSCO
	1023	177777	-1

Refer to the Software Definitions (DEFS) file for a listing of all system discsubs arranged by discsub number. System discsub numbers range from 0-177. Refer to Section 5.3.5 for information on specific IRIS discsub subsystems which may be made memory resident to achieve the best performance for the particular subsystem.

TABLE 5-4.

TABLE 5-4 DELETED

5.3.5 SUBSYSTEM DISCSUBS

Subsystem discsubs are associated with many of POINT 4's optional hardware and software packages. For maximum performance, as many as possible of the appropriate discsubs may be made memory resident. Use SETUP to modify the system DISCSUB table as described in Section 6.2.4. DSP may be used to make these changes as described in Section 5.3.3.1.

A list of discsub priorities for each subsystem is provided in the following subsections.

5.3.5.1 TAPE Subsystem Discsubs

The discsubs listed for the TAPE subsystem are used by both the MAGTAPE and CTU subsystems.

Although none of the discsubs for TAPE needs to be made memory resident, tape processing is more efficient when as many discsubs as possible (based on their priority) are made memory resident.

If the CTU is to be used as a simulated magnetic tape unit, the discsubs listed in Section 5.3.5.2 are used. They are not used by CTUTILITY.

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

5.3.5.2 CTU Subsystem Discsubs

If a CTU is used under IRIS to emulate magtape, special discsubs are used. These discsubs are not used with magtape. The special discsubs may be made memory resident for better performance.

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

5.3.5.3 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 (i.e., only SEARCH but not READ or WRITE), efficiency will not be affected if neither discsub is memory resident.

Polyfile discsubs with priorities 1 through 12 must be memory resident to achieve optimum polyfile performance. This is important even on systems which have an LCM.

5.3.5.4 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 equally important that the standard IRIS discsubs which process these files be memory resident.

5.3.5.5 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. Discsub number 163 is used infrequently; there is little advantage in making it memory resident.

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

Priority	Discsub Number
1*	160
1*	164
2	171
3	162
4	163

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

*Discsubs 160 and 164 are of equal importance.

5.4 DISC PARTITIONING

Disc partitioning is based on the type of hardware and the software used for a particular installation.

5.4.1 INTRODUCTION TO DISC PARTITIONING UNDER IRIS

A disc controller interfaces between one or more disc drives 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).

One or more disc drives on the same controller form a physical unit. Under IRIS, a physical unit contains one or more physical partitions. 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. These are listed in the IRIS R8 Peripherals Handbook.

NOTE

> It is essential that IRIS logical units do not extend into the area on disc designated for alternate tracks or sectors. If a logical unit extends into the area reserved for these alternate tracks, it could result in bad data being written to any track on the disc. Refer to the IRIS R8 Peripherals Handbook for the total number of available cylinders on disc. This number has been calculated to avoid the alternate track area.

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.

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

1. System driver used in normal system generation.

2. BZUD (Block Zero Utility Driver)

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

5.4.1.1 Logical Units Under IRIS

Each disc partition maps to a logical unit. A foreign unit is similar to a logical unit but it may only be accessed by input/output procedures using a special ($FOREIGN) driver (see Section 5.4.1.2). Logical unit numbers need not be related to disc partition numbers.

The IRIS Operating System always resides on LU/0. While the parameters for IRIS system files (LU/0) are preset, all other IRIS logical and foreign units have to be defined by creating the Disc Driver Table in the CONFIG file via the SETUP program (see Section 6.2.5). MILU (location 602 in CONFIG) must be incremented by the maximum number of foreign units that may be installed at any one session (i.e., MILU = IRIS logical units + foreign units).

Logical units may be numbered from 1 to 127. However, a foreign unit and an IRIS logical unit may not use the same number if both are to be INSTALLed at the same time. If the system is to support any foreign units, they may be numbered from 1 to 17 (octal).

5.4.1.2 Foreign Units

The foreign unit capability is provided so that IRIS can access data on disc or diskette not generated under IRIS. Any disc controller and drive combination supported by IRIS can be defined as a foreign unit.

A disc driver table (see Figure 5-1) is made up of a disc controller table and a number of disc partition tables. Word 3 (FUN) in a disc partition table (see Table 5-6) is used to designate a partition as a foreign unit by assigning it a unique number.

5.4.2 DISC DRIVER TABLE

After determining the size of each disc partition, use SETUP to enter these parameters into the Disc Driver Table as described in Section 6.2.5. The following is a general description of a Disc Driver Table under IRIS.

The Disc Driver Table 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 (see Figure 5-1). The Disc Driver Table must terminate with 177777 and must be contained within locations 1400 to 2377 (octal) in the CONFIG file.

Each Disc Controller Table consists of eight words and 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 consists of eight words and 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 5.4.3).

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). These eight words are written when the system is IPLed from information supplied when the system was built.

IRIS determines the parameters for partition 0.0 from information contained in the SOV in REX. SIR then stores parameters for partition 0.0 in the eight words of the Disc Driver Table in CONFIG, without regard to the previous contents, as an aid to the user in setting up the other LUs. The system manager may configure all other logical units as desired.

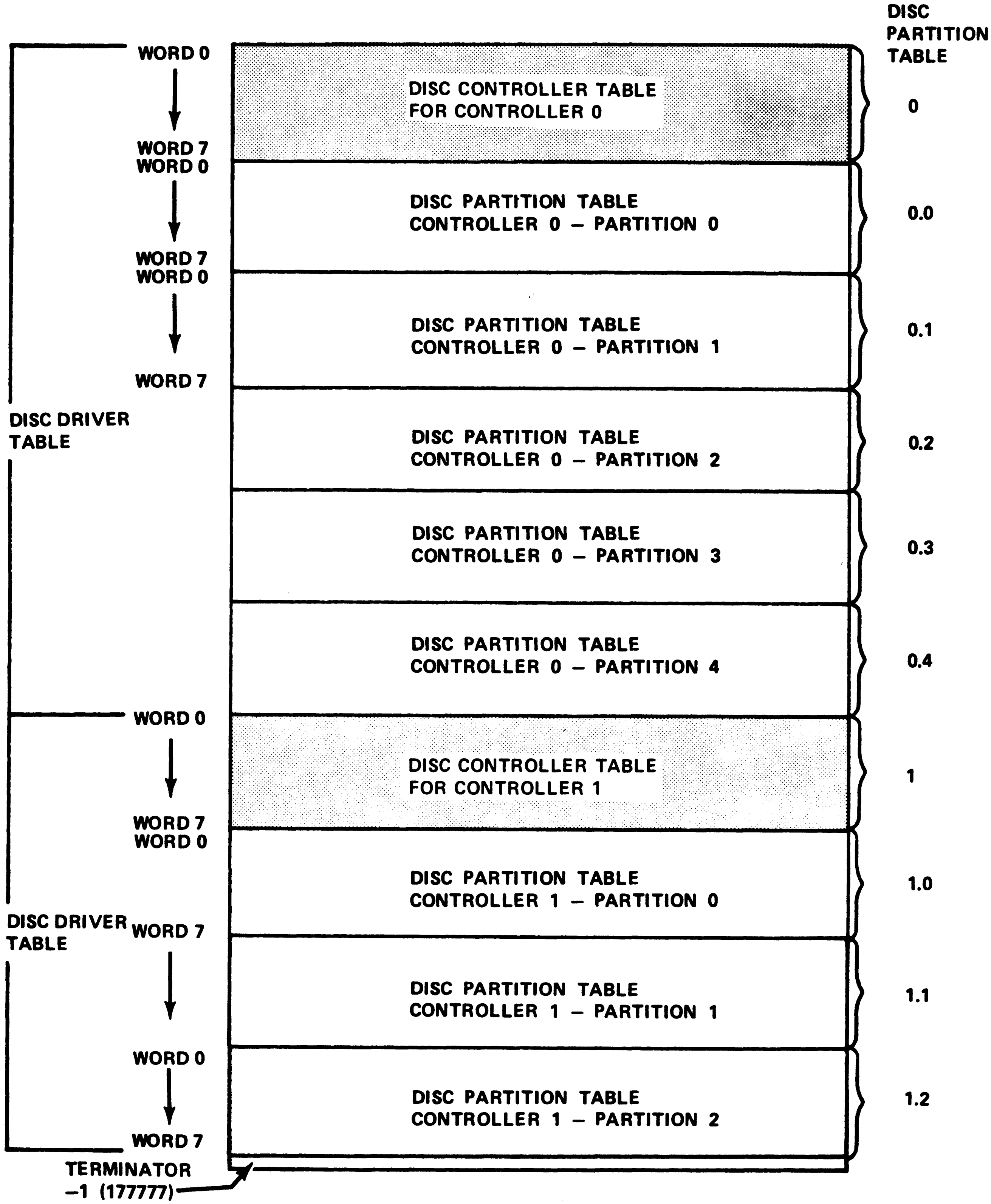


Figure 5-1. Example of a Disc Driver Table

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. A value of 52 entered into this word indicates to SIR that I/O instructions are not to be modified. 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.

Word 6 - Reserved.

Word 7 - Reserved.

1400:	26760	62356	62004	11	27	500	0	0	
1410	26725	12040	40500	0	20120	0	5	5020	
1420:	76243	24	40500	0	20120	5	141	5020	
1430:	76226	24	40500	0	20120	146	146	5020	
1440:	76211	24	40500	0	20120	314	146	5020	
1450:	76174	24	40500	0	20120	462	146	5020	
1460:	76157	24	40500	0	20120	630	146	5020	
1470:	76142	24	40500	0	20120	776	146	5020	
1500:	76125	24	40500	0	20120	1144	146	5020	
1510:	76110	24							
1520:	177777		40500	0	20120	1312	146	5020	

R8 DISC SPECIFICATION

ENTRY NO.: 34

CONTROLLER:

DISC ID: M2333

DATE: 10-26-82

DRIVE	Total Cyls On Disc	Max Cyls Other LUs
This entry # is currently NOT in use. FUJITSU M2333 DATAPLUS BOARD	1460	145
	No. Cyls in LU/0 20	

DEVICE CODE 27

DISC DRIVER ADDR 62356

BZUD ADDR 62004

LRC 1200

NPTC 24

DFLG 40500

NTRS 5020

PHYU 20120 +drive #

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

DISC COPY PROGRAM

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

CONFIG Address	OLD Contents	NEW Contents

NOTES

5.4.3 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)
1	NPTC - Number of physical tracks per cylinder
2	DFLG - Disc flag word (see Figure 5-2)
3	FUN - Foreign unit number
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 ($NT*100_8+NS$)

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 - Foreign unit number. Set to zero if the system does not have a foreign unit.

Word 4 - A physical unit select word used by BZUD and the system driver 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.

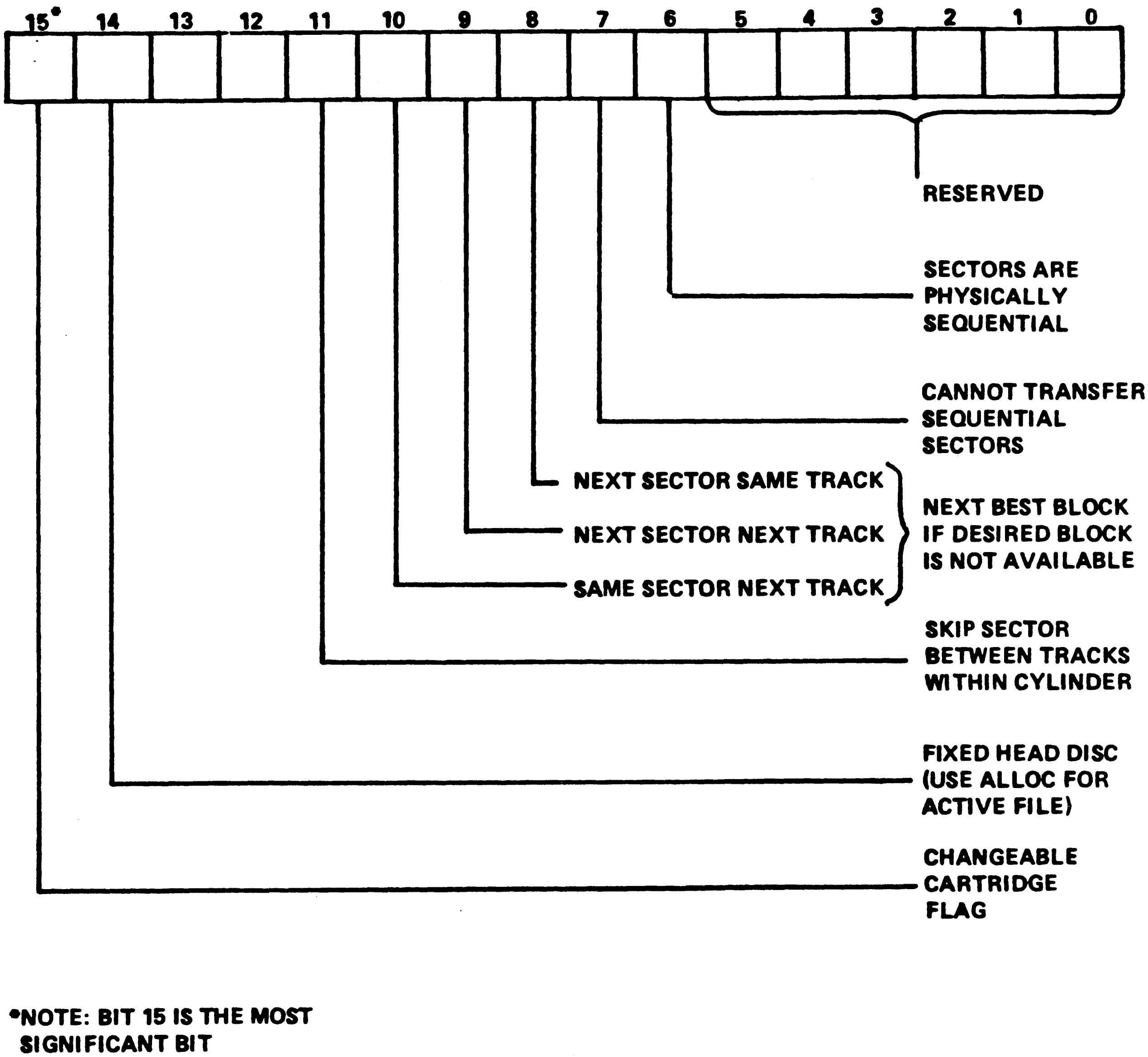


Figure 5-2. Disc Flag Word (DFLG)

Word 5 - The starting cylinder number of the partition.

NOTE

Cylinders are numbered starting at 0 on the disc.

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 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 $\geq$ 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 = 0100 001 001 100 000   in binary
        mode     group        user
         1    (9 decimal) (32 decimal)
```

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 $\leq$ 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$\leq$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 $\leq$ 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 a selected port 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.

If a user program has been specified to run automatically at log-on time (see Section 5.6.2), it preempts the initialization program. It is recommended that the initialization program be run from the first Mux port (port 1) and the utility account; the user program should be run from a nonzero port.

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 10     - (17410) program name
```

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

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
- A user program (see Section 5.6.2) has been specified to run on port 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. However, the port must be different from the port specified for the initialization program. The port specified for a user program will preempt an initialization program specified to run at IPL-time.

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 must be in the form

{lu/}filename<CTRL-Z>

(where <CTRL-Z> embeds a carriage return). If the logical unit number (lu/) is omitted, 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
16401:000002	Port 2
I16402:MENU<CTRL-Z>	Program name
16420:140000	Any user
16421:001004	Ports 4 or 5
I16422:12/INVENTORY<CTRL-Z>	Program name
16440:0	

In the first entry, if user 2,1 logs on to port 2, 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 4 or 5, the program INVENTORY residing on logical unit number 12 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

<u>ACCOUNTUTILITY</u>

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 using SETUP (see Section 6.2.3). 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 <u>must not</u> 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. <u>The Entry Table must not be changed</u>.

All driver files begin at location BPS. 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.

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). <u>The ATRIB Table must not be changed</u>.

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 177777 (-1 octal). If there is no Linkage Table, the -1 is found at ATRIB+3. The Port Definition Table immediately follows. <u>The Linkage Table must not be changed</u>.

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 PORT 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. POINT 4 recommends that the system configurator (SETUP) be used to set up the PDT (see Section 6). PDT consists of zero or more entries and must be terminated by 177777. It 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.

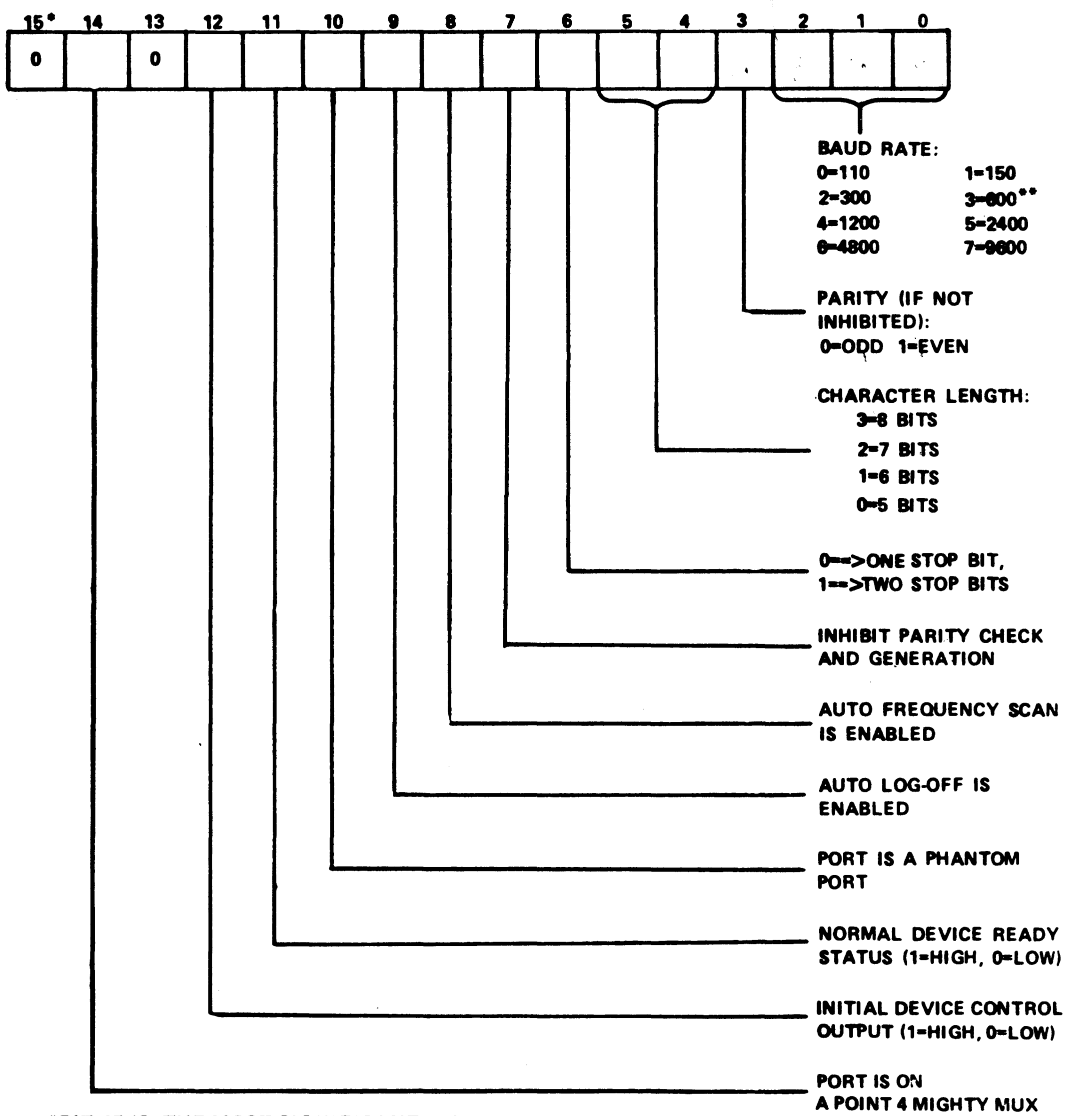
Word 6 - Size of active file on disc in blocks. The recommended size is 40 (octal) blocks.

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.



*BIT 15 IS THE MOST SIGNIFICANT BIT

**ON A POINT 4 MUX WITH THE 19200 BAUD OPTION, 3=19200, NOT 600

Figure 5-6. Port Control Word Format
(Does Not Apply to MARK 3)

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

NOTE

The PCW value for a phantom port on a MARK 3 is 2000.

5.8.2 MASTER PORT

The master port's main function is to access the system during an IPL. It is always port 0. The master port's baud rate must be set to 9600 baud.

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. POINT 4 recommends that SETUP be used to configure the PDT. However, 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 (PCW) is the second word in each set of eight words in the Port Definition Table. Refer to Section 5.8.1.4 for a description of PCW and how to set it. 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 unused port (port 15)

5. 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	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

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. 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 (if applicable).

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

<table>
<tr><th>Driver Name</th><th>Specifications</th></tr>
<tr><td>LPTM</td><td>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.</td></tr>
<tr><td>LPTP</td><td>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)</td></tr>
<tr><td>LPTD</td><td>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 C.</td></tr>
</table>

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 <u>not</u> include periods, other letters, or symbols. Examples of legal and illegal line printer driver names are as follows:

<u>Legal Name</u>	<u>Illegal Name</u>
LPT	LPTA
LPT1	LPTP
LPT2	LPT.1
.	LPT/3
.	
.	
LPT99	

To copy an IRIS line printer driver, at the system command prompt (#), enter

<u>COPY <00>LPT=LPTM</u>

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:

- 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 Using DSP to Make Appropriate Changes in the Driver Files

Before using DSP, there are a number of important things to remember:

- POINT 4 recommends that SETUP be used to modify the Port Definition Table.
- Always back up the system.
- 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
- 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.

```
1 BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789


  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789   A 2
  BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789


  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789


  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
3 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  KBCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
4
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789


  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

5 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
```

Figure 5-9. Printer Output

5.8.8 TERMINALS WITH AN INDIVIDUAL COMPUTER INTERFACE

Two terminals, each with its own computer interface, may be operated under IRIS. They may use device codes 10/11 and 50/51.

1. Master port terminal (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 terminal (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 terminal 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 terminal are controlled by a Teletype reader/punch driver ($PTM) as a noninteractive 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 terminal with its own interface (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 terminal is controlled by a secondary terminal 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., 1/2-inch 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. Refer to Section 5.8.10.2 for information on changing the buffer size.

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 contained in the CTUS file. 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.

CAUTION

> Polyfiles can NOT be transferred on a file-by-file basis using $MTA0 with either $MTAS (magnetic tape) or $CTUS (cassette tape).

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	100000+n
ATRIB-1	177777

   where n = drive 1, 2, 3, etc. corresponding to the driver filename. For the location of ATRIB, refer to Section 5.8.1.

   If a CTU is to be used as a simulated magnetic tape unit, enter the appropriate port number (origin 1) at ATRIB-1 (see Section 5.8.10.3).

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 Changing the Buffer Size in $MTAS

The reel-to-reel magnetic interface driver ($MTAS) handles any record size up to 8192 bytes. POINT 4 presets the buffer size to 400 (octal) words (i.e., 512 decimal bytes). This is the minimum size required by IRIS. Provided a particular system has enough memory available, the buffer size may be increased.

If an 8192 (decimal) byte buffer is desired, then twenty 400 (octal)-word buffers would be needed.

To change the buffer size in $MTAS, three factors must be computed:

1. Number of buffers desired
2. Size of buffers
3. ATRIB address

```
Assume n = Number of buffers desired
       S = Size of each buffer
       A = ATRIB address
```

The following algorithm will produce the buffer size required:

S(octal)= n x 400

For example, if 20 buffers are required, multiply 20 by 400; the resulting buffer size is 10000 (octal).

To compute the new address of the ATRIB table, add 33105 (octal) to the size of the buffer.

The procedure for increasing the buffer size in $MTAS is as follows:

1. Log on to the manager account.

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

   DSP <CTRL-E>key<CTRL-E> $MTAS

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

   Dump the contents at location 33077 of $MTAS by entering

   D33077

   The system should display

   20000 nnnnn nnnnn

4. Press <ESC> to stop the display. If the first number is not 20000, do not proceed because the version of $MTAS on the system is incorrect. Contact Customer Support for assistance.

5. Enter the required number of buffers:

    33104:n

    where n = the required number of buffers.

6. Define the new ATRIB table as follows:

    A   :000000

    A+1 :000040

    A+2 :000022

    A+3 :177777

    A+4 :177777

    32201:A

    For example, assuming 20 buffers for a total size of 10000, the following would be entered in steps 5 and 6.

    33104:20

    43105:0

    43106:40

    43107:22

    43110:177777

    43111:177777

    32201:43105

7. Exit DSP by entering

    X

8. Shutdown the system by entering

    SHUTDOWN <CTRL-E>key<CTRL-E>

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

9. IPL the system.

5.8.10.3 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 = cassette tape unit number (i.e., 0, 1, 2, 3, etc.)

   If the CTU is only to be used as a simulated magnetic tape unit, do not copy MTA0 but configure MTA0 by making the appropriate changes in ATRIB-1.

2. Use DSP to enter the following constants into the ATRIB table located at the end of the CTUn driver file:

Location	Enter
ATRIB-1	port # (origin 1)

   Port # is the logical system port number in octal. This port need not be dedicated exclusively to CTU. It may be configured as a normal interactive port (see Section 5.8.10.4).

   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. IPL the system.

5.8.10.4 Configuring a Port for a Cassette Tape Unit

Any port configured as an interactive port and set for 9600 baud may be used as a cassette tape unit (CTU) port. The logical port number must be entered into the CTUn driver file at ATRIB-1 (see Section 5.8.10.3). However, that port need not be dedicated exclusively to a CTU.

To use such a port for the CTU, the following steps are required:

1. Log off that port.

2. Remove the existing cable (if any) and plug the CTU cable into the mux port.

3. Run the desired tape control program from some other port.

To make the CTU port available for general use, remove the CTU cable, plug in CRT cable (if required), and log the port on. No IPL is required.

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.

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). TTC is the lower (right) byte of Word 3 in the Port Definition Table (see Section 5.8.1.4). 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.

5.9.1 ENABLING A TERMINAL TRANSLATION MODULE

POINT 4 recommends that SETUP be used to enable a required Terminal Translation Module ($TERM.name) as described in Section 6.2.3. The port is then linked automatically when the next IPL is performed.

If SETUP is not used, four steps are required to enable a TERM.name file and link a port:

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.

4. Link the port(s) to the Terminal Translation Module (see Section 5.9.2).

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. This is done automatically when SETUP (see Section 6.2.3) is used and a subsequent IPL is performed. The system links one or more ports at IPL-time after the TTC byte of Word 3 in the Port Definition Table has been modified.

If SETUP is not used, 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).

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

   <u>PORT p TYPE n</u>

   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

   <u>PORT TYPE n</u>

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

<u>PORT TYPE 0</u>

5.10 TIMESHARING

Timesharing is the method by which numerous users and jobs are serviced seemingly simultaneously by the IRIS Operating System. The scheduler allocates time based on account, program, and other parameters set by the system manager.

5.10.1 SCHEDULER FUNCTIONS

The scheduler uses a system of dynamic priorities to determine which user is to run next (i.e., the next regnant user) and which memory partitions are to be flushed to disc if contention arises.

5.10.1.1 Priority

The scheduler establishes an initial dynamic priority for each job based inversely on its effective priority. Therefore, a job with a higher effective priority 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 via the ACCOUNTUTILITY program 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 automatically set to 5 by the system. Program priority may be adjusted by the system manager using the CHANGE processor.

The scheduler maintains a dynamic priority value for each user which is decremented each time one of the jobs in the queue receives a time slice. When this value reaches zero, that job is eligible for the next time slice. Once the job has received a time slice, the scheduler resets the job's priority to its maximum value which is inversely proportional to the user's effective priority. The cycle then begins again until the job becomes dormant or an interaction is terminated. A job is considered dormant when it is no longer contending for CPU time. Usually, a job does not require CPU time during keyboard input, terminal output, or a pause.

5.10.1.1.1 INTERACTION

An interaction is the interval between the initiation of a user request at a terminal (i.e., the user leaves the dormant state) and the output of the system's response at the terminal (i.e., the user returns to the dormant state).

An interaction typically begins when one of the following conditions occurs:

- Input done is initiated by pressing the <RETURN> key
- <ESC> or <CTRL-C> is pressed
- Terminal output is completed
- A signal 3-type pause completes

An interaction ends when a program begins the next input, terminal output, or a signal 3-type pause.

When an interaction is initiated, the job receives an interactive benefit by having a larger percentage of time slices allocated to it. This is done via an automatic priority boost for the next eight time slices. The boost is reduced for each subsequent set of time slices until it reaches zero.

When a user ends an interaction, the scheduler sets the priority as if the user had just started a new interaction. This priority is decreased normally on the assumption that the job is ready to run. This feature prevents highly interactive programs from receiving an unfair number of time slices. It maintains good throughput while avoiding the danger of thrashing. The scheduler checks the status of the job only when the priority reaches zero. If the job is ready, it receives the next time slice. If the job is not ready (i.e., input or output is not complete), the priority is set to a value of 512 which indicates a currently dormant state. When a dormant job starts an interaction, it receives the next time slice.

5.10.1.1.2 ALLOCATION OF MEMORY PARTITIONS

Memory partitions are also allocated based on the dynamic priority system. If all partitions are currently assigned, then the partition with the highest priority value (i.e., farthest from zero and especially if the priority equals 512) is written to disc. It then becomes available for the job that is ready to run.

When a job starts input or output, its partition is not immediately freed up. The job is assigned a priority value which indicates the start of an interaction and it must fall to zero before the job can run again. If a user's program does short input or output frequently, its priority will not indicate a dormant state (i.e., set to 512) immediately. Thus, it will tend to keep its partition which reduces thrashing and swapping overhead.

5.10.1.2 Foreground and Background Mode

A job running 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 throughput may be defined in terms of an interaction (see Section 5.10.1.1.1) and whether the computer is in an interactive or compute-bound state.

Response time is the time required to complete a short interaction.

Throughput is a measure of the work done by a program in a given amount of time.

The state of a job is either interactive or compute-bound 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 a calculated number of time slices or the interaction completes.

The system, as a whole, is said to be in an interactive state while one or more jobs are in an interactive state. Thus the initiation of a single interaction causes the system, as a whole, to be in an interactive state.

Conversely, 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 or disc-bound.

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, all jobs receive long time slices. Short time slices are desirable because they create the potential for improving apparent response time but long time slices are more efficient because they reduce swapping overhead.

The overall system goal is to maximize throughput and minimize response time. Frequently the parameters set to increase the rate of throughput adversely effects response time and vice versa. Recommendations for achieving a proper balance are given in Section 5.10.3.

5.10.1.3.1 RESPONSE TIME

An example of good response time would be an immediate response even if the system is heavily loaded (i.e., many users are actively running).

Response time is primarily affected by two factors:

- Number of users on the system
- System scheduler

Once a user initiates an interaction, the system scheduler controls the number of slices given to other jobs before a given job gets enough time to complete its interaction. The time required to complete an interaction determines the response time. The scheduler controls the allocation of time slices based on the dynamic priority discussed in Section 5.10.1.1. Setting the parameters that control the scheduler is discussed in Section 5.10.2.

5.10.1.3.2 SYSTEM THROUGHPUT

Work done by the system (i.e., processing) may involve many interactions or one long interaction.

Good throughput is characterized by only a minimum increase in processing time for a given user as the system load increases. For example, a program might require five minutes processing time before it generates a report when the system has only one active user. The same program generating the report in 5.5 minutes when the system is heavily loaded would represent good throughput.

Throughput is primarily affected by four factors in the following order:

1. Number of users on the system
2. Amount of overlapped processing
3. System scheduler
4. System overhead

It is natural to expect that the throughput seen by each user decreases as the number of users on the system increases. However, the throughput decrease depends on how much of the processing done by these jobs can be overlapped (i.e., done in parallel). Output to a terminal, $LPT, and magtape, and input to a terminal can all be overlapped. Whereas, compute- or disc-bound programs cannot be overlapped.

For example, during terminal input and output, six users run very quickly. Each user gets good response and good throughput as if no other users were on the system. However, if the same size jobs were compute- or disc-bound, they could not overlap and each job would take six times longer to complete (i.e., throughput goes down by a factor of six).

Throughput is also affected by the scheduler. A job having a higher priority than another, gets more time slices, completes sooner, hence has better throughput than the job with a lower priority. Throughput would be adversely affected if the scheduler gave too many time slices to one job and not enough to another. The IRIS scheduler gives extra time slices to users beginning an interaction but it makes sure that all jobs get a fair allocation of time slices overall.

System overhead also has a impact on throughput. Overhead is the time the system requires to swap users and to handle interrupts. As system overhead rises, throughput decreases. By setting longer time slices (see Section 5.10.2.1), swapping is reduced. This results in less overhead and, therefore, greater throughput. Refer to Section 5.10.3 for recommendations on setting time slice parameters.

5.10.2 TIMESHARING PARAMETERS

Timesharing parameters that may be modified by the system manager include:

- Time slice parameter word (KTSL) in CONFIG
- Job priority in the ACCOUNTS file
- Program priority in the program file header

The scheduler uses the default values set by the system if these parameters are not modified. The following subsections describe the possible range of values that may be set, the default value, and how to modify the parameters. All values are given in octal.

5.10.2.1 Time Slice Parameter (KTSL)

The time slice parameter word (KTSL) is located in the CONFIG file's System Information Table (see Table 5-2). Refer to Section 5.10.3 for recommendations on time slice parameter settings.

KTSL (time slice parameters) is located at 617 in CONFIG. The upper eight bits are used for setting the long time slice parameter, the lower eight bits are used for the short time slice parameter as shown in Figure 5-10.

Contents of the long time slice parameter (bits 15-8) are:

```
UNIT    :    Tenths of a second
RANGE   :    1-377
DEFAULT:     50
```

Contents of the short time slice parameter (bits 7-0) are:

```
UNIT    :    Tenths of a second
RANGE   :    1-377
DEFAULT:     3
```

Figure 5-10. Time Slice Parameter Word (KTSL)

5.10.2.2 Job Priority

Two parameters affect job priority (see Section 5.10.1.1). 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
```

5.10.3 RECOMMENDATIONS FOR SETTING TIME SLICE PARAMETERS

Setting of the time slice parameter word (KTSL) in CONFIG concerns the proper balance of long and short time slices to achieve the best possible response time and throughput for a particular group of users and system configuration. The following subsections describe the recommended approach to these considerations and give examples of KTSL settings for specific system configurations.

5.10.3.1 Deciding Between Response Time And Throughput

The particular characteristics of an application at the user site are important factors in determining whether to emphasize response time over throughput. Setting KTSL for better response time results in the fast completion of short time slices at peak periods but may cause a delay in completing longer interactions. Setting KTSL for better throughput, causes long interactions to complete faster but delays response time for shorter interactions. Some experimentation may be required to establish a satisfactory balance.

As a starting point, let short interactions be the ones which take less than a short time slice to print a response once <RETURN> is pressed. Let long interactions take more than one short time slice to print a response.

Where short interactions that produce almost instantaneous responses at peak periods are not an important consideration, set KTSL for the best throughput. Such a situation applies to a system where the majority of users are experienced data entry people who use type-ahead to enter answers to questions before they are asked.

On other systems, operators may require virtually instantaneous responses to short interactions much as a touch typist requires a responsive keyboard. On these systems, response time tends either to be satisfactory or unsatisfactory for short interactions (i.e., if it is satisfactory, it does not make any difference if response time is improved). If there are several possible values for KTSL after taking the system configuration into consideration, experiment with them to determine which offer satisfactory response time for short interactions with a typical number of users on the system. Of those values, choose the one which also gives the best throughput.

5.10.3.2 Time Slice Parameters and System Configuration

The system configuration is an important consideration in setting the time slice parameters as described in the following subsections.

5.10.3.2.1 NUMBER OF MEMORY PARTITIONS EQUALS INTERACTIVE USERS

On a system where the number of memory partitions is the same as the number of interactive users, all users can remain memory resident and there will be no swapping. In this case, system overhead is negligible and the recommended setting for the long time slice parameter is 5 (.5 seconds) and the short time slice parameter is 1 (.1 seconds). Set

```
KTSL = 2401
```

This configuration gives both excellent response time and throughput. However, if there are more users than memory partitions and the system does not have an LCM, this value of KTSL would result in poor performance.

The number of memory partitions is defined at location 401 (NPART) in the CONFIG file's General Information Table (see Section 5.2.1). The number of interactive users equals the number of terminals plus the number of phantom ports in use, i.e., the total number of interactive ports. $LPT and $COM do not count as interactive users.

5.10.3.2.2 SYSTEMS WITH LOTUS CACHE MEMORY ($LCM)

The LCM transfers data at a rate of 2.5 megabytes per second. This means that a block of data (256 words) is transferred in approximately .2 milliseconds. If the active file contains 32 blocks (PSIZ=20000), then a swapout and swapin (64 blocks transferred) require 12.8 milliseconds (.0128 seconds). If the short time slice is set to one (.1 seconds), then a penalty of 12.8% overhead for swapping occurs (swapping = .0128 seconds leaving .0872 seconds for the user program).

Because many users will go dormant within their allotted time slice due to starting new input or output, the actual overhead for all short time slices will be greater than 12.8% on the average. Increasing the short time slice parameter to .2 seconds, reduces overhead to 6.4%.

For best response time, set the long time slice parameter to 5 (.5 seconds) and the short time slice parameter to 1 (.1 seconds) by setting

    KTSL = 2401

For better throughput (i.e., less overhead) set the long time slice parameter to 6 (.6 seconds) and the short time slice parameter to 2 (.2 seconds) by setting

    KTSL = 3002

Refer to section 5.10.3.1 for more information on response time and throughput.

5.10.3.2.3 SYSTEMS WITH A FLOPPY DISC DRIVER

A system that has LU/0 on a diskette, has more interactive users than memory partitions, and has no LCM, generally requires longer time slices. Such a system may require that the long time slice parameter be set to 74 (6 seconds) and the short time slice parameter to 24 (2 seconds) by setting

    KTSL = 36024

Accessing a diskette requires more time (.5 seconds average) than hard disc (.04 seconds average). Thus an order of magnitude performance difference is to be expected. Some systems may find that much smaller or much larger values for the time slice parameters may provide some benefit. It is recommended that the number of users be limited to the number of partitions available on a floppy system and KTSL=2401 can then be used.

5.10.3.2.4 OTHER SYSTEMS

The time slice parameter settings for other system configurations depend to a greater degree on the type of users supported (see Section 5.10.3.1) and the type of jobs to be run than the systems discussed in previous sections. The other major consideration is the percentage of swap time produced by each short time slice setting. Depending on the disc controller and drive combination, swap times vary from one system to another. For a general-purpose timesharing environment (not necessarily confined to IRIS) with PSIZ set to 20000, a swapin or swapout requires approximately 78.4 milliseconds as shown in Figure 5-11.

Step	Time (milliseconds)
One average seek time to get the correct track	20.0 milliseconds
One average latency to get sector 0	8.3 milliseconds
One maximum latency to write active file	16.7 milliseconds
One maximum latency to seek next active file and get sector 0	16.7 milliseconds
One maximum latency to read active file	16.7 milliseconds
Total	78.4 milliseconds

Figure 5-11. Swapping Steps

Based on an estimate of 78.4 milliseconds required for swapping, Table 5-10 shows the calculated percentage rate for selected short time slice parameter settings. These relative percentages demonstrate the vastly improved overall system performance that can be achieved by using an LCM (see also Section 5.10.3.2.2).

TABLE 5-10. SWAPPING OVERHEAD PER SHORT TIME SLICE SETTING

Short Time Slice (KTSL Bits 0-7)	Overhead Swapping Percentage Rate
2 (.2 seconds)	39.2%
3 (.3 seconds)	26.1%
4 (.4 seconds)	19.6%
5 (.5 seconds)	15.7%
6 (.6 seconds)	13.0%

Table 5-11 shows possible values for KTSL ranging from best response time and worst throughput to best throughput and worst response time.

TABLE 5-11. KTSL SETTINGS FOR GENERAL SYSTEM CONFIGURATIONS

Performance	Short Time Slice (Bits 0-7)	Long Time Slice (Bits 8-15)	KTSL
Best Response Time	2 (.2 seconds)	12 (1 second)	5002
	3 (.3 seconds)	14 (1.2 seconds)	6003
	4 (.4 seconds)	16 (1.4 seconds)	7004
	5 (.5 seconds)	20 (1.6 seconds)	10005
Best Throughput	6 (.6 seconds)	22 (1.8 seconds)	11006

5.11 BASIC PROGRAM AND PROCESSOR RESTRICTION

Certain processors (system commands) may be restricted by means of passwords or account privilege. Frequently both types of restrictions are used. Section 5.11.1 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).

The GUARD utility program provides a method for allowing access to BASIC programs containing restricted statements from any account. GUARD requires IRIS R8.2C1 or later. The utility is described in Section 5.11.3.

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.

When these processors are invoked, a password must be given, using the command format

    {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>
Ffilename
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
I540:<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-on 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 5-11.

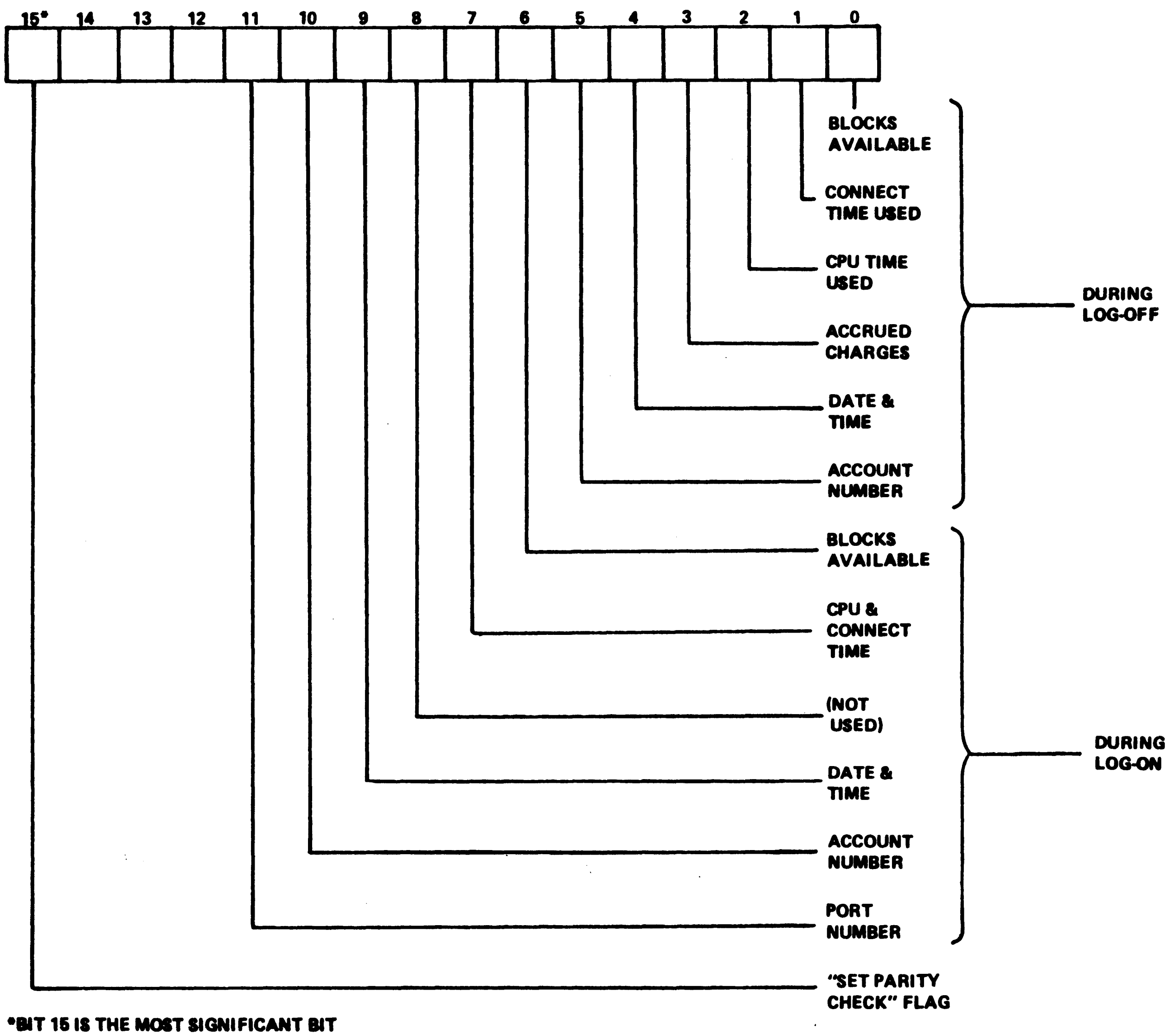


Figure 5-11. Message Flag Word (MSGFL)

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 - <u>not</u> 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, INSTALL terminates, the file is retained, and control is returned to SCOPE (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 REHASH 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 REMOVE 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 SHUTDOWN set:

- p - Allow SHUTDOWN from Port Zero (or the first Mux port) and Port p only, where p is the logical system port number in octal
- -1 - Allow SHUTDOWN from all ports

5.11.3 THE GUARD UTILITY PROGRAM

The GUARD utility program provides a method by which the system manager can allow limited access to restricted functions. These functions are desirable and useful when handled in a cautious and knowledgeable manner. However, because they are very powerful, they are also very dangerous and could cause serious problems to the operating system and user data and programs if misused.

The functions to which the system manager may allow limited access from a BASIC program include:

OPEN FILE MAINTENANCE - allows the header and data of any file to be examined and modified (similar to DSP).

CALL 93 - allows writing words into the user's program area in the user's partition.

SPC (32768 + N) - allows any address N in lower memory (i.e., below 32KW) to be read.

SPC (65536 + N) - allows any address N in lower or upper memory (i.e., below 64KW) to be read.

CALL 99 (or CALL $TIME) - allows the system clock and date to be changed.

Most of these functions are available when run from the manager's account without being GUARDed. However, a manager may find it desirable to allow another user to run a program containing one of these functions without revealing the manager password.

One example is where a junior operator does backups late at night or early in the morning. When bringing the system up, the system manager may want this junior operator to set the date and time from a BASIC program that checks all the input and makes sure that it is reasonable. The manager does not want to give out the password nor would the manager desire any user to be able to use CALL 99 ($TIME) from any account.

The GUARD program provides a solution to this dilemma. It allows the manager to write a BASIC program using CALL 99 that can be run from any account. Once GUARDed, that program cannot be modified nor listed, a CALL 99, in general, still can only be run from the manager account.

Thus curious or potentially malicious users would not be able to use CALL 99. Only those programs that have been GUARDed can use CALL 99 from any account. Before GUARDing the program, the manager can add any desired level of checking by using SPC 5 to restrict usage or by checking the desired date and time. Once GUARDed, the program cannot be listed or modified, so that CALL 99 can only be used as intended.

The header of each BASIC program contains a set of bits called the DOOM bits. The GUARD program allows the user to set certain DOOM bits as shown in Table 5-12. The DOOM bits can only be set (i.e., enabled or disabled) from the manager or utility accounts. When a program is first created, all DOOM bits are disabled.

TABLE 5-12. DOOM BIT OPTIONS

DOOM Bit Option	Description
1	Allows use of OPEN FILE MAINTENANCE to access files regardless of file type only if not prevented by the account's privilege or the file's protection levels.
2	Allows use of OPEN FILE MAINTENANCE to access files regardless of file type, privilege and protection levels.
3	Allows use of CALL 93 for writing to memory within the user partition. Also allows use of CALL 99 ($TIME) to set system time (IRIS R8.2C or later).
4	Allows use of SPC (65536 + N) or SPC (32768 + N) for unrestricted reading of memory.
5	Execute only. Prevents listing or modifying a BASIC program by anyone including the system manager.

Note that PROTECT prevents listing a program, does allow modification, and is permanent. GUARD option 5 prevents both listing and modification but is not permanent because the option can be reset.

If a program is GUARDed but option 5 is disabled, the program mode is "execute only" unless the program is run from the manager account.

5.11.3.1 GUARD Program Password

The GUARD utility program is protected by a password. The password may be changed at line number 570 in the GUARD program. The IRIS Business BASIC statement is:

```
570 LET C2$ = "X"
```

GUARD is as powerful as DSP and POINT 4 recommends that similar precautions be taken to prevent unauthorized use by changing the password in GUARD and then resaving the program.

5.11.3.2 Using The GUARD Program

The GUARD program is an interactive program displaying appropriate prompts and messages. The user may exit the program at any time by pressing <ESC> or <CTRL-C>.

The procedure for using GUARD is as follows:

1. Log on to the manager or utility account.

2. At the system command prompt, enter

   GUARD

   The program then displays

   GUARD Version n.nn

   where n.nn is the current revision number.

   The following prompt is then displayed:

   ENTER PASSWORD:

3. Enter the GUARD program's password in the following format:

   <CTRL-E>key<CTRL-E>

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

   The password is not echoed. If an incorrect password is entered, the program aborts and the system command prompt (#) is displayed.

   If the correct password is entered, GUARD requests the name of the BASIC program to be GUARDed:

   ENTER PROGRAM NAME

4. Enter the name of the BASIC program. The user may enter the name of a BASIC program residing on any logical unit by prefixing the program name with the logical unit number (e.g., 6/TESTER).

   If the name entered is misspelled or is not a BASIC program, an appropriate message is displayed and the program name prompt is repeated.

If a correct BASIC program name is entered, GUARD displays the current settings of the DOOM bits as shown in the following example (user input is underlined):

```
BASIC PROGRAM NAME 3/B

              THE GUARD WORD FOR   3/B  IS NOW:

1 .....READ-WRITE FILES OBEYING FILE PROTECTION          DISABLED

2 .....READ-WRITE FILES IGNORING FILE PROTECTION         DISABLED

3 .....WRITE TO MEMORY                                   DISABLED

4 .....READ FROM MEMORY                                  DISABLED

5 .....EXECUTE ONLY - CAN NOT BE LISTED                  DISABLED

OPTION NUMBERS TO CHANGE (#,#) :
```

5. Enter the number of the option which is to be enabled. If more than one option is to be enabled, the desired option numbers may be entered at the same time provided they are separated by commas.

   If the option numbers are not separated by commas, the numbers are rejected as shown in the following example:

```
OPTION NUMBERS TO CHANGE (#,#) : 23

23 .....IS NOT A VALID OPTION NUMBER
```

   The Option Numbers To Change prompt is redisplayed.

If the option numbers are entered correctly, the user is asked to confirm each option chosen as shown in the following example:

```
OPTION NUMBERS TO CHANGE (#,#) : 1,3

1 .....READ-WRITE FILES OBEYING FILE PROTECTION  IS NOW DISABLED

                                        ENABLE IT (Y OR N) ? Y

3 .....WRITE TO MEMORY    IS NOW DISABLED

                                        ENABLE IT (Y OR N) ? Y

              THE GUARD WORD FOR    3/B   IS NOW:

1 .....READ-WRITE FILES OBEYING FILE PROTECTION    ENABLED

2 .....READ-WRITE FILES IGNORING FILE PROTECTION           DISABLED

3 .....WRITE TO MEMORY                             ENABLED

4 .....READ FROM MEMORY                                    DISABLED

5 .....EXECUTE ONLY - CAN NOT BE LISTED                    DISABLED

                              IS THIS CORRECT (Y OR N) ? Y

THE GUARD WORD HAS BEEN WRITTEN FOR 3/B

BASIC PROGRAM NAME
```

6. Enter the name of the next program to be GUARDed and continue the procedure or press <RETURN> to exit the GUARD utility.

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.

IRIS uses multiple fixed-size partitions. If there are more users on the system than the number of user partitions available, IRIS saves the contents of the program's partition onto disc and replaces 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.

Under IRIS, four steps are required when configuring a system:

1. Set Partition Size (PSIZ) - the size of each partition.

2. Set the number of user partitions (NPART) to be in memory.

3. Identify the type of memory (MTYPE) which is either standard multiple fixed partitions or partitions in the MARK 9 map.

4. Set the proper active file size for each interactive port.

POINT 4 recommends that the system configurator (SETUP) be used to configure user partitions (PSIZ). Refer to Section 6.2.2 for information on using SETUP.

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 (PSIZ) required for a system is determined by the size of the largest BASIC program 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 5-13 for the corresponding octal value for PSIZ.

TABLE 5-13. PARTITION SIZE SELECTION TABLE

Partition Size PSIZ (Octal)	Maximum BASIC Program Size (Decimal)	Partition Size PSIZ (Octal)	Maximum BASIC Program Size (Decimal)
2000	719	21400	8655
2400	975	22000	8911
3000	1231	22400	9167
3400	1487	23000	9423
4000	1743	23400	9679
4400	1999	24000	9935
5000	2255	24400	10191
5400	2511	25000	10447
6000	2767	25400	10703
6400	3023	26000	10959
7000	3279	26400	11215
7400	3535	27000	11471
10000*	3791	27400	11727
10400	4047	30000	11983
11000	4303	30400	12239
11400	4559	31000	12495
12000	4815	31400	12751
12400	5071	32000	13007
13000	5327	32400	13263
13400	5583	33000	13519
14000	5839	33400	13775
14400	6095	34000	14031
15000	6351	34400	14287
15400	6607	35000	14543
16000	6863	35400	14799
16400	7119	36000	15055
17000	7375	36400	15311
17400	7631	37000	15567
20000	7887	37400	15823
20400	8143	40000	16079
21000	8399		

*Minimum value of PSIZ

5.12.2 DETERMINING NPART

NPART is set at location 401 in the CONFIG file. The ideal number of user partitions to be memory resident is equal to the maximum number of users actively using the system at any one time. This reduces swapping and a reduction in swapping improves system performance. However, the number of memory-resident partitions must depend on the following:

- Amount of memory available.
- LOTUS Cache Memory (LCM) - On a system with an LCM, NPART should be set to one because the swapping overhead has already been drastically reduced and memory is better used as a buffer pool (see also Section 5.10.3 and the R8 LCM Installation Document).
- On a system with a MARK 9 CPU, the number of user partitions is typically considerably larger.

5.12.2.1 PSIZ and NPART 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 for system buffers (see Section 5.13).

Six words of lower memory are required for each buffer in the buffer pool. Decreasing PSIZ and/or NPART in a 64KW system increases the number of buffers in the buffer pool and that reduces the amount of lower memory available.

If the total partition area (PSIZ * NPART) is substantially reduced, it may result in a Trap 141 on IPL, indicating a lower-memory overflow.

5.12.2.2 PSIZ For 32K-Word Memory

With a POINT 4 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.

Alternatively, PSIZ may be enlarged by moving the $MMUX port control blocks (PCBs) in multiples of 1000 (octal). Figure 5-13 is an example showing how DSP commands are used to move the $MMUX PCBs (user input is underlined). See also Section 5.8.1.2 (ATRIB Table). If DBUG will be used for debugging, the PCBs for DBUG must also be moved to correspond with the PCBs in $MMUX.

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>. Let x1 = the contents of location 32201 which is a pointer to the first Mux port address in the port control area.
Dx1	Dump contents of x1. Let x2 = the contents of x1.
x1: x2+2000	Increase the current value of x1 by 2000. Usually x2 is 66000 so x2+2000 will be 70000.
Dx1	Dump location x1 to check that the change was made correctly. It is now possible to increase PSIZ at 400 in CONFIG by up to 2000/NPART.

Figure 5-13. Example of Moving the $MMUX PCB

5.12.3 DETERMINING MTYPE

MTYPE is set at location 402 in the CONFIG file. IRIS supports two types of memory: the standard 32KW memory or 64KW memory on the POINT 4 MARK 3 and MARK 5 Computers or the mapped memory on a POINT 4 MARK 9 Computer.

- For the standard memory, set MTYPE = 0
- For the mapped memory, set MTYPE = 1 and enable $SYS.MAP

5.12.4 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.

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) and the temporary dirty page flag (TDPF) are used to control the point at which a dirty page is to be written to disc. The TDPF and NDPF are set at location 606 (bits 13 and 15) in the CONFIG file (see Sections 5.14.2 and 5.14.3 respectively). However, both cannot be set at the same time (i.e., if TDPF is set, NDPF cannot be set. Conversely, if NDPF is set, TDPF cannot be set).

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.

A compromise between total system buffering (NDPF=0) and no buffering (NDPF=1) is end-of-time-slice disc buffering (see Section 5.14.3). End-of-time-slice disc buffering is achieved by setting the temporary dirty page flag (TDPF). This guarantees that dirty pages are written to disc at the end of a user's time-slice and that related disc writes for updating a polyfile bit map or an index file are completed within the time slice.

5.14 SPECIAL CONDITIONS FLAG WORD (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 13 (temporary dirty page flag), bit 14 (suppress error message flag), and bit 15 (no dirty page flag) are used. All other bits are reserved for future use.

5.14.1 SUPPRESS 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 suppress error message flag (SEM) in the SPCF word at 606 (octal) in CONFIG.

When SEM is set to 0 (0 is the default), the descriptive text is printed with the appropriate error code (see Appendix E).

When SEM is set to 1, the error codes are printed but the descriptive text is suppressed.

5.14.2 NO DIRTY PAGE FLAG

The no dirty page flag (NDP) 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. NDP is bit 15 in the SPCF word at location 606 (octal) in CONFIG.

If NDP 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 NDP 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 NDP=0 without sacrificing data integrity.

5.14.3 TEMPORARY DIRTY PAGE FLAG

The temporary dirty page flag (TDPF) offers an alternative to the NDPF flag. TDPF is bit 13 in the SPCF word at location 606 (octal) in CONFIG. If TDPF is set (TDPF=1), all dirty pages are written to disc at the end of a user's time slice and any associated disc writes, such as updating the polyfile bit maps or an index file update are completed.

It is recommended that TDPF be set rather than NDPF because multiple disc writes within a process (e.g., index file updates) are not meaningful until the process is completed. Furthermore, polyfile performance may be severely degraded if no dirty pages (NDPF=1) are allowed.

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 CPU or MARK 5-type CPU without extended instruction set
- $EIS - For MARK 5 CPU with extended instruction set and for MARK 9 CPU; allows access to the extended instruction set

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 without the extended instruction set, no further action is required. For a MARK 9 CPU or a MARK 5 CPU with the extended instruction set, $DEC must be disabled and EIS 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 EIS

```
IF NO CHANGE, PRESS RETURN
NEW NAME:  $EIS

COST = $0.00
NEW COST? <ESC>
#
```

3. SHUTDOWN the system.

4. Re-IPL.

Section 6
A GUIDE TO USING SETUP

This section is a guide to the SETUP utility program. SETUP is an interactive system configurator which provides users with an easy method for configuring the IRIS environment.

6.1 INTRODUCTION TO SETUP

SETUP is a utility program written in BASIC consisting of a series of program modules which enable the user to configure parts of the IRIS Operating System without the use of DSP. POINT 4 recommends that SETUP be used for the following:

- The INFO tables in the CONFIG file
- The Port Definition Table (PDT)
- The memory-resident DISCSUB Table
- The Disc Driver Table

SETUP contains a separate program module for each of these tables. Each module has one or more screen displays that prompt for the required parameter entries. Numerical entries are generally made in octal (unless otherwise noted). If a decimal number is entered, it must be followed by a period.

Each table is first defined for a control file. It may then be listed separately to make sure the entries are correct. The parameters are not entered into the system files until the SETUP update function is run.

When SETUP is run for the first time this control file must be created. The control file serves as a scratch pad and record of the system configuration. Three temporary work files are used to store the various parameters entered by the user. They are:

SU.WORKnnn
SU.FRMTnnn
SU.SAVEnnn

where nnn is the port number on which the program is being run.

After the required parameters have been entered into the work files, they are saved into the control file. When SETUP is used to modify the system at a later date, the contents of the control file will be displayed as default entries for the various parameter prompts.

The SETUP utility uses two other parameter files. One is a formatted file called SU.ENTRIES. It contains disc controller-driver information arranged by entry number as published in the IRIS R8 Peripherals Handbook. The other is a text file called SU.DSUBS. It lists all IRIS discsubs by name and number.

When a new system is shipped, these two parameter files contain up-to-date information. It is the responsibility of the user to update these files when changed pages are issued for the IRIS R8 Peripherals Handbook or the system DEFS file is revised. SETUP provides a special module for updating the SU.ENTRIES file. The SU.DSUBS file may be updated and user-supplied discsubs may be added by using the IRIS editors.

Before SETUP is run, the system should be backed up. This is absolutely essential. If an error is made in configuring any of the tables and SETUP's update function is executed, it may be impossible to IPL the system.

SETUP automatically inserts the table terminator (177777) when the system files are updated by its update function.

General guidelines for using SETUP are as follows:

Menu Selection - SETUP functions are selected from various menus. Entry of the number associated with a particular function invokes the required program module.

<ESC> - May be used for any of the following purposes:

- Exit a menu or a program and return to the previous menu.
- Back up to the previous entry field of any screen.
- Exit from the main menu.

<RETURN> - A parameter is entered or a default is accepted by pressing <RETURN>.

COMMENT/COMMAND/MESSAGE lines - These lines are displayed for every screen. The comment line displays system prompts. The command line is for user input. The message line displays messages from the system.

6.2 USING THE SETUP UTILITY

To initiate the SETUP utility, log on to the Manager account and at the system command prompt (#), enter

SETUP

The System Configuration menu (i.e., the main menu) is then displayed as shown in Figure 6-1.

```
PORT n                  SYSTEM CONFIGURATION               SETUP n.n  mm/dd/yy



                  (0) EXIT THE SYSTEM CONFIGURATION

                  (1) CREATE/MAINT CONFIGURATION CONTROL FILE

                  (2) LIST CONFIGURATION CONTROL FILE

                  (3) UPDATE THE SYSTEM CONFIGURATION

                  (4) DISC DRIVE ENTRIES FILE MAINTENANCE



COMMENT:  ENTER THE NUMBER OF THE FUNCTION YOU WISH TO EXECUTE
COMMAND:
MESSAGE:
```

Figure 6-1. System Configuration Menu

Enter 0 to abort the program and return to the system command prompt (#).

Enter 1 to create, copy, or modify the control file (see Section 6.2.1).

Enter 2 to list the control file (see Section 6.2.6).

Enter 3 to update the system configuration (see Section 6.3).

Enter 4 to list or update the SU.ENTRIES file (see Section 6.4).

6.2.1 THE CONFIGURATION CONTROL FILE

If SETUP is used for the first time, a new control file must be created. On subsequent runs, the control file may be copied or modified. A user may wish to copy an existing control file when another disc controller-driver combination is added to the system and the same kind of configuration is required or the configuration is to be modified slightly.

To create, copy, or modify the control file, enter a 1 at the command line of the main menu. The Control File Maintenance screen is then displayed (see Figure 6-2).

```
PORT n          CREATE/MAINT CONFIGURATION CONTROL FILE     SU1 n.n  mm/dd/yy

                  CONTROL FILE NAME:










COMMENT:
COMMAND:
MESSAGE:
```

Figure 6-2. Initial Control File Maintenance Screen

6.2.1.1 Creating a New Control File

To create a new control file, enter any legal IRIS filename. Make sure it is not already in use. The following prompt is then displayed in the center of the menu:

    CONTROL FILE TO COPY, OR PRESS RETURN

Press <RETURN>. SETUP then prompts for type of computer and displays a list of valid computer types with a numeric code as shown in Figure 6-3. Enter the appropriate code and the Configuration Control File Maintenance menu is displayed as shown in Figure 6-4.

6.2.1.2 Copying an Existing Control File

To copy an existing control file, enter either a new name at the first prompt or use the existing name followed by an exclamation mark (e.g., CNTRNAME!). SETUP then prompts

    CONTROL FILE TO COPY, OR PRESS RETURN

Enter the name of the control file that is to be copied. The Control File Maintenance menu is then displayed (see Figure 6-4).

```
PORT n            CREATE/MAINT CONFIGURATION CONTROL FILE      SU1 n.n  mm/dd/yy

                                                           VALID TYPES
                                                           -----------
                        CONTROL FILE NAME:  name           1= MARK 2T

CONTROL FILE TO COPY, OR PRESS RETURN:                     2= MARK 3

                            COMPUTER TYPE:                 3= MARK 3T

                                                           4= MARK 4T

                                                           5= MARK 5

                                                           6= MARK 8

                                                           7= MARK 9



COMMENT:
COMMAND:
MESSAGE:
```

Figure 6-3. Computer Type Prompt and Codes

6.2.1.3 Modifying an Existing Control File

If an existing control file is to be modified, enter the name of the control file at the first prompt. If the file is found, the control file maintenance menu is displayed (see Figure 6-4). If the file cannot be opened, an appropriate error message is displayed.

```
PORT n          CONFIGURATION CONTROL FILE MAINTENANCE  SU11 n.n  mm/dd/yy


                 (0) RETURN TO SYSTEM CONFIGURATION MENU

                 (1) CONFIGURE SYSTEM INFORMATION TABLE

                 (2) CONFIGURE PORT DEFINITION TABLE

                 (3) CONFIGURE DISCSUB TABLE

                 (4) CONFIGURE DISC DRIVER TABLE




COMMENT:  ENTER THE NUMBER OF THE FUNCTION YOU WISH TO EXECUTE
COMMAND:
MESSAGE:
```

Figure 6-4. Control File Maintenance Menu

6.2.2 CONFIGURING THE SYSTEM INFORMATION TABLE

To configure the System Information Tables, enter 1 on the command line of the Control File Maintenance menu (see Figure 6-4). The System Information Table screen is then displayed as shown in Figure 6-5. Please note that parameters should be entered in octal unless otherwise noted. If a decimal number is entered, it must be followed by a period. SETUP will convert the decimal number and display its octal equivalent.

```
PORT n            CONFIGURE SYSTEM INFORMATION TABLE      SU111 n.n  mm/dd/yy

              USER PARTITION SIZE:  20000
         NUMBER OF USER PARTITIONS:
                 AVERAGE CPU SPEED:
           NUMBER OF DATA CHANNELS:
 ALLOW TEMPORARY DIRTY PAGES (Y/N):
           ALLOW DIRTY PAGES (Y/N):
DISPLAY BASIC ERROR MESSAGES (Y/N):
         MEMORY SIZE (64KB, 128KB):
             AUXILIARY BUFFER SIZE:
           NUMBER OF USER DISCSUBS:
  NUMBER OF EXTRA CHAR QUEUE NODES:
      MINIMUM NUMBER OF FREE NODES:
     NUMBER OF SIGNAL BUFFER NODES:




COMMENT:  ALL ENTRIES ARE IN OCTAL, DECIMAL INPUT MUST BE FOLLOWED BY A PERIOD
COMMAND:
MESSAGE:
```

Figure 6-5. System Information Table Screen

USER PARTITION SIZE (PSIZ) is based on BASIC program sizes. The recommended and most generally used size is 20000 (octal). This size is displayed as a default when a control file is first created. Refer to Section 5.12.1 for more information on setting PSIZ.

NUMBER OF USER PARTITIONS (NPART) is the number of user partitions in memory. The ideal number of memory-resident partitions is equal to the number of users logged on and actively using the system at any one time. Refer to Section 5.12.2 for more information on setting NPART.

AVERAGE CPU SPEED (SPED) is the average speed of an instruction per millisecond. Average speeds for a POINT 4 CPU are as follows:

MARK 5/8 = 2000 (octal)
MARK 3 = 1200 (octal)

To set SPED for other CPUs running IRIS, refer to location 601 of the INFO Table described in Section 5.2.2.

NUMBER OF DATA CHANNELS (NDCH) is the number of data channels for each port. NDCH is usually set to 10 (12 octal). The minimum is 2.

ALLOW TEMPORARY DIRTY PAGES (TDPF) is bit 13 of the special conditions flag (SPCF). When set, it forces a write-to-disc at the end of a user's time slice. Enter Y to set this flag. If an N is entered, the user may set the NDPF flag (see below). Refer to Section 5.13.4 for a discussion on setting this flag.

ALLOW DIRTY PAGES (NDPF) is bit 15 of SPCF. When set, it forces a write-to-disc of any dirty buffer pool page. This flag cannot be set if TDPF (see above) is set. To set this flag, enter N. Refer to Section 5.13.3 for more information.

DISPLAY BASIC ERROR MESSAGES (SEMF) is bit 14 of SPCF. Enter N to suppress error messages.

MEMORY SIZE (KB) is the size of CPU memory in K-bytes (entered in decimal).

AUXILIARY BUFFER SIZE (ABUF) is the size in words of the auxiliary buffer area. It must be at least 1004 words (octal) if indexed files are to be used.

NUMBER OF USER DISCSUBS is the highest number assigned to a user-supplied discsub in the DISCSUBS.USER file. For example, if user-supplied discsubs are numbered 1, 2, 3, and 15, the value 15 should be entered. The default is zero. (Does not cause discsubs to be made memory resident.)

NUMBER OF EXTRA CHAR QUEUE NODES (NCQN) is the number of extra character queue nodes required. The system initializing routine (SIR) allocates two nodes per interactive port. The minimum value for NCQN is 2 (see also Table 5-2 (location 613)).

MINIMUM NUMBER OF FREE NODES (NNOD). Each node occupies 32 decimal words.

NUMBER OF SIGNAL BUFFER NODES (NSIG) is the maximum number of signals which can be waiting to be received. Each node occupies 4 words of memory. The minimum value is 1.

To continue the configuration process, press <RETURN> and the Configuration Control File Menu is redisplayed (see Figure 6-4). Press <ESC> to back the cursor up to the previous prompt.

6.2.3 CONFIGURING OR MODIFYING THE PORT DEFINITION TABLE

To configure or modify the Port Definition Table (PDT) for a driver file, select option 2 at the Control File Maintenance menu (see Figure 6-4). SETUP then displays the Port Driver Selection menu as shown in Figure 6-6. Enter the appropriate selection number.

```
PORT n                  PORT DRIVER SELECTION              SU112 n.n  mm/dd/yy



                     (0) RETURN TO CONTROL FILE MAINT. MENU

                     (1) CONFIGURE POINT 4 MUX PORTS

                     (2) CONFIGURE PHANTOM PORTS

                     (3) CONFIGURE NON-POINT 4 MUX PORTS



COMMENT:  ENTER THE NUMBER OF THE FUNCTION YOU WISH TO EXECUTE
COMMAND:
MESSAGE:
```

Figure 6-6. Port Driver Selection Menu

6.2.3.1 Configuring the PDT for the First Time

If the PDT for the control file is being configured for the first time, the Port Configuration Screen is displayed as shown in Figure 6-7.

```
PORT n               CONFIGURE PORTS-POINT 4 MUX          SU112A n.n  mm/dd/yy

     DRIVER FILE NAME:  $MMUX
TOTAL NUMBER OF PORTS:

NO. OF            BUFFER   TERMINAL   ACTIVE FILE
PORTS     PCW      SIZE      TYPE         SIZE
======   =====    ======   ========   ===========










COMMENT:  ALL ENTRIES ARE IN OCTAL, DECIMAL INPUT MUST BE FOLLOWED BY A PERIOD
COMMAND:
MESSAGE:  ENTRY IS REDISPLAYED IN OCTAL
```

Figure 6-7. Port Configuration Screen

If option 1 or 2 was selected, SETUP defaults to the $MMUX or $PHA driver file, respectively. The name of that driver file is then displayed at the top of the Port Configuration screen ($MMUX is used as an example in Figure 6-7). If option 3 was selected, the name of the driver file must be entered (e.g., enter $DGMX if the driver has been enabled (see Section 3.3); otherwise, enter DGMX). If a driver file is modified before the driver is enabled, the SETUP update function must be used before enabling the driver.

Parameter entries should be made in octal. If a decimal number is entered, it must be followed by a period. SETUP then converts the decimal number and displays the octal equivalent.

TOTAL NUMBER OF PORTS is the total number of ports (interactive or line printer) to be configured in the driver's PDT. SETUP keeps track of the individual ports that are then specified on the detail lines. The number of ports specified on the detail lines must add up to the total number of ports specified for the system. If too many ports are specified, an appropriate error message is displayed.

NO. OF PORTS is the total number of ports to be configured with a given set of characteristics (i.e., PCW, buffer size, terminal type, and active file size).

PCW is the Port Control Word in the driver's PDT (refer to Section 5.8.14 for a complete description). When the cursor arrives at the PCW prompt (i.e., the previous parameters have been entered), a help module consisting of the most commonly used entries for the PCW is displayed at the right side of the screen (see Figure 6-8).

BUFFER SIZE is the size of the I/O buffer for the port(s) in bytes. The most commonly used buffer sizes are:

Interactive port for CRT	=	135 (decimal) bytes
Interactive port for a modem	=	85 (decimal) bytes
Non-interactive port for a line printer	=	512 (decimal) bytes

TERMINAL TYPE is the type of terminal to be used on the specified port(s). A help module is displayed at the right side of the Port Definition screen giving numeric codes for terminal types supported under IRIS (see Figure 6-9).

ACTIVE FILE SIZE is the size of the active file on disc (in blocks) for the specified port(s). If the System Information Table has already been configured for the control file (see Section 6.2.2), SETUP calculates the recommended active file size for an interactive port based on the PSIZ specified for the INFO table. The recommended value is then displayed as a default. Non-interactive ports (e.g., line-printer ports) do not require an active file and a zero should be entered.

```
PORT n              CONFIGURE PORTS-POINT 4 MUX          SU112A n.n  mm/dd/yy

     DRIVER FILE NAME:  $MMUX                         MOST COMMON ENTRIES:
TOTAL NUMBER OF PORTS:                                1ST: 5= POINT 4 MUX
                                                           1= OTHER
NO. OF             BUFFER   TERMINAL   ACTIVE FILE    2ND: 4= PRINTER
PORTS     PCW       SIZE      TYPE        SIZE             2= PHANTOM
======   =====     ======   ========   ===========         0= CRT
                                                      3RD: 3= NO PAR, 2 STOP
                                                           2= NO PAR, 1 STOP
                                                           1= PAR, 2 STOP
                                                           0= PAR, 1 STOP
                                                      4TH: 7= EVEN, 8 LEN
                                                           6= ODD, 8 LEN
                                                           5= EVEN, 7 LEN
                                                           4= ODD, 7 LEN
                                                      5TH: BAUD 7= 9600
                                                           6= 4800 5= 2400
                                                           4= 1200 3= 600
                                                           2= 300 1= 150

COMMENT:  ALL ENTRIES ARE IN OCTAL, DECIMAL INPUT MUST BE FOLLOWED BY A PERIOD
COMMAND:
MESSAGE:  ENTRY IS REDISPLAYED IN OCTAL
```

Figure 6-8. PCW Parameters

```
PORT n              CONFIGURE PORTS-POINT 4 MUX          SU112A n.n  mm/dd/yy

     DRIVER FILE NAME: $MMUX                               0-NO TRANSLATION
TOTAL NUMBER OF PORTS:                                    15-ADDS
                                                           1-ADM 1A/31
NO OF              BUFFER   TERMINAL   ACTIVE FILE         3-ADM 3A
PORTS     PCW       SIZE      TYPE        SIZE            10-BEEHIVE 100
=====    =====     ======   ========   ===========        11-DG6052/6053
                                                          17-DIALOGUE80
                                                           6-ELITE 1520A
                                                           7-ELITE 1521A
                                                           5-GE TERMINET
                                                          12-HAZELTINE 1500
                                                           9-HAZELTINE 2000
                                                          13-MT ACT-V
                                                           1-SOROC IQ120
                                                          14-TV912/920
                                                           4-TV950
                                                           8-VT100

COMMENT:  ALL ENTRIES ARE IN OCTAL, DECIMAL INPUT MUST BE FOLLOWED BY A PERIOD
COMMAND:
MESSAGE:  ENTRY IS REDISPLAYED IN OCTAL
```

Figure 6-9. Terminal Type Codes

When the last parameter (Active File Size) has been entered, SETUP displays three options at the comment line

(M)ODIFY, (D)ISPLAY, (S)AVE

If the table is to be modified, refer to Section 6.2.3.2 for the procedure. Entry of D causes the parameters just entered to be redisplayed. S must be entered to save the parameters (i.e., enter them into the the control file from the work files). If <ESC> is pressed at this point, the parameter entries are aborted. Once the entries have been saved, the main menu is redisplayed.

6.2.3.2 Modifying the PDT

To modify the PDT, specify the appropriate driver file name (see Section 6.2.3). The Port Definition Screen is then displayed as shown in Figure 6-10.

```
PORT n               CONFIGURE PORTS-POINT 4 MUX            SU112A n.n  mm/dd/yy

     DRIVER FILE NAME:  $MMUX
TOTAL NUMBER OF PORTS:  10

NO. OF             BUFFER   TERMINAL   ACTIVE FILE
PORTS     PCW       SIZE      TYPE        SIZE
======   =====     ======   ========   ===========








COMMENT:  ALL ENTRIES ARE IN OCTAL, DECIMAL INPUT MUST BE FOLLOWED BY A PERIOD
COMMAND:
MESSAGE:  ENTRY IS REDISPLAYED IN OCTAL
```

Figure 6-10. Port Definition Screen for an Existing PDT

Enter the name of the driver file to be modified. The total number of ports previously specified is then displayed as a default. If the total number of ports is to be changed, enter the appropriate number. Refer to Section 6.2.3.2.1 for information on decreasing the number of ports. Refer to Section 6.2.3.2.2 for information on adding ports.

If there is no change in the total number of ports, press <RETURN>. SETUP then displays the following options:

(M)ODIFY, (D)ISPLAY, (S)AVE

Enter D to display the current configuration. The option message is then repeated. Enter M to change any of the individual port configurations. The following message is then displayed:

ENTER STARTING PHYSICAL PORT NUMBER (ORIGIN 0)

Physical port number, origin 0, refers to the entries under No. of Ports. For example, assume the following entries:

Physical Port		NO. OF PORTS	PCW	BUFFER SIZE	TERMINAL TYPE	ACTIVE FILE SIZE
		======	===	======	========	============
0	=	1	50277	207	0	40
1	=	1	40367	1031	0	0
2	=	1	50056	400	0	0
3-7	=	5	50277	207	4	40

The port listed on the first line is physical port 0, last port on line 4 is physical port 7 (remember to count in octal).

Enter the physical port number for the port entry that is to be modified. The screen is then cleared and the cursor is positioned at the No. of Ports prompt. Enter the number of ports (i.e., the quantity). The parameters previously entered for that physical port are displayed as default entries. As the cursor moves to each parameter (PCW, Buffer Size, etc.), press <RETURN> to accept the default value or enter a new value for that parameter. SETUP then modifies the remaining number of ports to correspond with the total number specified. The modified PDT may then be displayed.

For example, assume (in the configuration shown above) that another port with the same characteristics as physical port number one is to be added. Enter 1 at the command line. When the cursor is positioned at the NO. OF PORTS prompt, enter 2 (i.e., the total number of ports required) and accept all the default entries. Then enter D to display the modified PDT. The following configuration is then displayed:

NO. OF PORTS	PCW	BUFFER SIZE	TERMINAL TYPE	ACTIVE FILE SIZE
=====	=====	======	========	===========
1	50277	207	0	40
2	40367	1031	0	0
1	50056	400	0	0
4	50277	207	4	40

When the changes are completed, use the save option to enter the new configuration into the control file. SETUP then redisplays the Control File Maintenance menu.

6.2.3.2.1 DELETING A PORT

If the total number of ports is decreased, the individual ports (i.e., the number listed under No. of Ports) are automatically adjusted.

For example, assume that the total number of ports parameter contains 10 (octal) and the individual ports are listed as follows:

```
NO. OF
PORTS
======
  1
  1
  1
  5
```

If the total number of ports is then decreased from 10 to 6, SETUP will change the entry under the No. of Ports parameter from 5 to 3. SETUP displays

    (M)ODIFY, (D)ISPLAY, (S)AVE

The modified table entries should first be displayed on the screen by entering D. The modify option may then be used to make any further changes (see Section 6.2.3.2).

6.2.3.2.2 ADDING A PORT

If the total number of ports is increased, the cursor is positioned at the No. of Ports prompt.

The parameters for the additional port(s) must be entered first. When the total number of individual ports matches the number given for the Total Number of Ports parameter, SETUP displays

    (M)ODIFY, (D)ISPLAY, (S)AVE

Enter D to display the new configuration. If any other changes are to be made, enter M and modify the individual port entries as described in Section 6.2.3.2.

6.2.3.3 Deleting a PDT from the Control File

A PDT for a driver file may be deleted from the control file by entering the appropriate option (1, 2, or 3) at the Port Driver Selection menu (see Figure 6-6). The Configure Ports screen is then displayed (see Figure 6-7). Enter the name of the driver file that is to be deleted. At the Total Ports prompt, enter

0

The program displays the following message at the comment line:

DELETE THIS DRIVER?

The default is no, if an N is entered or <RETURN> is pressed, the cursor returns to the Total Number of Ports prompt at the top of the screen.

If the deletion is confirmed by entering Y, the following message is displayed:

UPDATING control filename, DO NOT DISTURB!

When the driver file has been deleted, the program redisplays the Control File Maintenance menu (see Figure 6-4).

6.2.4 CREATING OR MODIFYING THE MEMORY-RESIDENT DISCSUBS TABLE

To create or modify the memory-resident discsubs table for the control file, select option 3 at the Control File Maintenance menu. The contents of the SU.DSUBS file are then displayed in groups of 36. Each group represents a page with the discsub names listed in sequence. These pages may be modified by using the edit commands described in Section 6.2.4.2.

If the discsub table has already been defined for the control file, an asterisk to the left of the discsub name indicates that it was made memory-resident. A $-sign to the right of a discsub name indicates that it is an IRIS system discsub and that POINT 4 has included it in the preset system DISCSUBS Table. Unless such a discsub is to be removed from the preset DISCSUBS Table, it should be made memory-resident in the control file's memory-resident discsubs table. Refer to Section 5.3.2 for a list of IRIS discsubs and their assigned priority in the preset system DISCSUBS Table.

```
PORT n                         DISCSUB SELECTION                   SU113 n.n  mm/dd/yy


NO.    *  DISCSUB NAME        NO.    *  DISCSUB NAME        NO.    *  DISCSUB NAME
===    =  ============        ===    =  ============        ===    =  ============
01.       AFSET=67  ($)       13.       SHUFF=62
02.       LINK=100  ($)       14.       DEKEY=63
03.       FFILE=3   ($)       15.       CLEAR=27
04.       ACNTL=15  ($)       16.       SPECI=46
05.       OPEN=22   ($)
06.       CLOSE=26  ($)
07.       GETRR=30
08.       READI=33
09.       ALLOC=1
10.       CHARG=40
11.       READC=36
12.       SEARC=61

COMMENT:  A=ALL, nn=SEQ NUMBER, P=PAGE, R=RESTART, U=UPDATE, W=WRAP, ?=HELP
COMMAND:
MESSAGE:
```

Figure 6-11. DISCSUB Selection Screen

6.2.4.1 Control File DISCSUBS Table Help Modules

The DISCSUBS module of the SETUP utility has two help modules which may be invoked by entering ? at the command line.

The first help module may be invoked when the DISCSUBS Selection screen is first displayed. It lists all IRIS discsubs by their assigned number grouped by type: discsubs associated with the IRIS Operating System (i.e., standard discsubs), the MAGTAPE or CTU subsystems, polyfile discsubs, etc. (see Figure 6-12). To invoke the second help module, press <RETURN>. To return to the DISCSUB Selection screen, press <ESC>.

The second help module provides a description for each edit command (see Section 6.2.4.2). To return to the DISCSUB Selection Menu, press <RETURN>.

```
PORT n                        DISCSUB SELECTION                 SU113 n.n  mm/dd/yy

SOFTWARE  ASSOCIATED DISCSUB NUMBERS
========  ==========================
  SYSTEM: 67, 100, 3, 15, 22, 26, 30, 33, 1, 40, 36, 61, 62, 63,
          27, 46, 57, 41

 MAGTAPE: 72, 76, 77, 74, 75, 71, 73

     CTU: 102, 103, 104

POLYFILE: 122, 142, 133, 134, 136, 123, 127, 130, 132, 124, 125
          126, 140, 135, 137, 141, 121, 120, 144






COMMENT:  PRESS <RETURN> TO CONTINUE
COMMAND:
MESSAGE:
```

Figure 6-12. DISCSUB Listing by Type

6.2.4.2 DISCSUB Module Edit Commands

The sequence numbers given at the left of the discsub name are used to edit the entries on the page. For example, if only the first six discsubs listed on the page shown in Figure 6-11 are to be made memory resident, the remaining discsubs must be deleted from the screen. Enter 7-16 on the command line and discsubs seven through sixteen are then erased from the screen.

Only those discsubs remaining on pages that were reviewed are entered into the control file's memory-resident discsubs table. Discsubs listed on pages that were not reviewed are not entered into the control file. A description of each edit command is given in Table 6-1.

TABLE 6-1. DISCSUB MODULE EDIT COMMAND SUMMARY

Command	Function
A	All - Erases all discsubs on the current screen. None of the discsubs erased from the screen will be entered into the control file's discsubs table.
nn	nn - Sequence number assigned to a discsub on a page display. Used to erase discsub names from the screen. Two numbers separated by a hyphen will erase a range of discsub names. Several numbers may be entered at one time provided they are separated by a comma or a space. For example, an entry of 1,7-9 25 erases numbers 1, 7, 8, 9, and 25.
P	Page - Stores for later processing the discsubs left on the screen, and displays the next 36 selected discsubs until EOF (end of file).
R	Restart - Redisplays the current screen as it was before any erasure(s).
U	Update - Enters those discsubs into the control file that were not erased from the screen.
W	Wrap - Stores the remaining discsubs for subsequent processing. It restarts the selection process by renumbering discsubs that were not deleted and wrapping to the beginning of the workfile.

6.2.5 CONFIGURING THE DISC DRIVER TABLE

The disc driver table consists of a disc controller table for each controller, followed by a number of disc partition tables. The first disc controller table is for the system disc controller. The first disc partition table is for the system logical unit (partition 0.0). The system logical unit (usually called LU/0) cannot be defined via the SETUP utility. It is initialized by the IRIS Operating System at IPL-time. Refer to Section 5.4.1 for more information on the disc driver table.

In this context, disc controller or logical disc controller refers to a driver (i.e., software). Generally, there is one driver for each computer interface. However, occasionally separate drivers are created for specific controller drive combinations.

The SETUP utility uses a parameter file called SU.ENTRIES for the disc driver table module. The parameters in this file are based on the R8 Peripherals Handbook. These parameters are used by SETUP to display default values for the number of cylinders on LU/0, maximum cylinders on other logical units, total number of cylinders, NPTC, DFLG, and a PHYU code. The contents of the SU.ENTRIES file may be displayed at the terminal or printed. Refer to Section 6.4.3 for the procedure.

NOTE

> It is vitally important that the SU.ENTRIES file be updated when changed pages are received for the R8 Peripherals Handbook (see Section 6.4).

The user may also refer to the R8 Peripherals Handbook for the proper device code, entry number, and possible numbers for fixed and removable surfaces. The PHYU code is based on the drive, fixed/removable, and surface/platter parameters. It is used by the disc driver table module to position the cursor at the appropriate prompt.

For example, if the user specifies a removable surface, the cursor skips the platter/surface prompt. Refer to Section 6.4 for more information on the PHYU code.

Each controller must be configured separately for the disc driver table because no two controllers may have the same device code except for device code 52.

If <ESC> is pressed before all the controllers have been defined and before the parameters have been saved into the control file, the module is aborted and the Control File Maintenance menu is redisplayed.

6.2.5.1 Creating The Disc Driver Table For The Control File

To configure the disc driver table for the control file, select option 4 at the Configuration Control File Maintenance menu (see Figure 6-4). The Disc Driver Table screen is then displayed as shown in Figure 6-13.

```
PORT n                CONFIGURE DISC DRIVER TABLE          SU114 n.n mm/dd/yy

                  TOTAL NUMBER OF LOGICAL CONTROLLERS:

        LOGICAL CONTROLLER NO: 0
                  DEVICE CODE:
            NO. OF PARTITIONS:

PARTITION  DRIVE  ENTRY  FIX/  PLATTER/  MAX. CYLS  STARTING  NUMBER OF
   NO.      NO.    NO.   REM   SURFACE   OTHER LUS  CYLINDER  CYLINDERS

   0.1











COMMENT:
COMMAND:
MESSAGE:
```

Figure 6-13. Disc Driver Table Screen

TOTAL NUMBER OF LOGICAL CONTROLLERS. Enter the total number (decimal) of controllers to be defined for the system.

LOGICAL CONTROLLER NO. SETUP displays the number of the particular controller being configured.

DEVICE CODE. Enter the device code given in the R8 Peripherals Handbook for the particular controller/driver combination on the system.

NO. OF PARTITIONS. Enter the number (decimal) of partitions required for that controller.

PARTITION NO. Enter the partition number to be defined for that controller. The program displays the first partition that may be defined for the system controller (i.e., partition 0.1). Partition 0.0, the system logical unit, is defined by the IRIS Operating System at IPL-time.

DRIVE NO. Enter the drive number for the system (0-7).

ENTRY NO. Enter the entry number from the R8 Peripherals Handbook (see Figure 6-14) for the particular controller-driver combination to be configured.

FIX/REM. Enter whether the surface for that partition is fixed or removable. Refer to the R8 Peripherals Handbook for information.

PLATTER/SURFACE. Enter either the platter or surface number. The platter number may be 0-n depending on the number of platters in the disc pack. Each platter may have two surfaces starting with surface 0 (i.e., the top surface of the first platter is 0, the underneath is 1 and the top surface of the next platter is number 3, etc.). The cursor skips this prompt if an entry for this parameter does not apply.

MAX. CYLS OTHER LUS. Enter the maximum number of cylinders for other logical units (except LU/0) from the R8 Peripherals Handbook. A default value based on the lowest maximum value given for an entry number in the R8 Peripherals Handbook is displayed. This value is used to ensure that values entered for starting cylinder and number of cylinders do not overflow the disc.

STARTING CYLINDER. Enter the starting cylinder number for a partition. 0 for the first partition or the first cylinder after the end of a previous partition is displayed as a default. If a user wishes to have gaps between partitions, the default may be overwritten. If the starting cylinder number entered here is less than the end of the previous partition, the program displays the warning

OVERLAPPING PARTITIONS

NUMBER OF CYLINDERS. Enter the maximum number of cylinders for that partition.

R8 DISC SPECIFICATION

CONTROLLER: POINT 4 LOTUS 700[1]

ENTRY NO.: 38

DISC ID: P480MB

DATE: 08-01-83

DRIVE	Total Cyls On Disc	Max Cyls Other LUs
Ampex DM-940 (40MB)	626	631[3]
DM-980 (80MB)	1462	
DM-9160 (160MB)[2]	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 62356

BZUD ADDR 62004

LRC 240

NPTC 5

DFLG 40500

NTRS 1220

PHYU D + 20024
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.
[3]LU MAY NOT exceed total cylinders on disc.

Figure 6-14. Sample R8 Peripherals Handbook Specification Sheet

When all parameters for the first controller have been entered, the program displays the following message:

(M)ODIFY, (D)ISPLAY, OR (S)AVE

If any of the parameters just entered need to be changed, enter M (see Section 6.2.5.2 for information on modification procedures). If all the entries are correct, enter S on the command line. This saves the entries into the SU.SAVEnnn file.

After the parameters have been saved into the work file, SETUP displays the Disc Driver Table screen again. The program displays the number of the next logical controller. The user may then proceed to configure this controller. When all the specified controllers have been defined and saved into the work file, SETUP displays

ENTER CONTROLLER TO MODIFY, OR PRESS <RETURN> WHEN DONE

To enter the controllers into the control file, press <RETURN>. This saves the configured controllers into the control file. The Control File Maintenance menu is then redisplayed. If one of the controllers just defined requires modification, enter the number of that controller.

6.2.5.2 Modifying the Disc Driver Table for the Control File

To modify the disc driver table for the control file, enter option 4 at the Control File Maintenance menu. The disc driver table screen as shown in Figure 6-13 is displayed. The previously defined total number of controllers is displayed as a default.

A new controller may be added by entering a number that is larger than the total number of controllers displayed (see Section 6.2.5.2.1). One or more controllers may be deleted from the disc driver table by entering a number that is smaller than the total number of disc controllers displayed. Refer to Section 6.2.5.2.2 for information on the procedure.

To modify the partitions of a previously defined controller, press <RETURN>. SETUP then displays the default values for the device code and the number of partitions. Press <RETURN> to enter the default values or enter new values.

If the device code is changed, the controller must be newly configured (see Section 6.2.5.1). If the device code remains the same but one or more partitions are to be added, enter the appropriate number (see Section 6.2.5.2.3). If one or more partitions are to be deleted, enter the lesser number at the number of partitions prompt (see Section 6.2.5.2.4).

If the parameters of a previously defined partition are to be changed, press <RETURN>. SETUP then displays the following options:

    (M)ODIFY, (D)ISPLAY, OR (S)AVE

Enter M at the command line. The program clears the screen and the cursor rests at the partition number parameter prompt. Enter the number of the partition that is to be changed. SETUP then displays all partition configuration up to and including the partition to be modified. Partitions that follow the specified partition number are erased from the screen.

The user may then enter the required changes as the cursor moves from one parameter prompt to the next. A <RETURN> enters the default values (i.e., the parameters previously defined). The subsequent partition must then be redefined. When the last parameter for all the partitions has been entered, the program displays

    (M)ODIFY, (D)ISPLAY, OR (S)AVE

The completed revision may be displayed by entering D. The modified entries must be saved by entering S at the command line. The program then displays

    LOGICAL CONTROLLER TO MODIFY, OR <RETURN> WHEN DONE

Another controller may then be modified by repeating the procedure. When the required modifications have been completed, press <RETURN>. This updates the control file and the Control File Maintenance menu is redisplayed.

6.2.5.2.1 ADDING A CONTROLLER TO THE DISC DRIVER TABLE

After a larger number has been entered at the Total Number of Controllers prompt, the cursor positions at the device code prompt. Enter the appropriate device code and then proceed to define the new controller as described in Section 6.2.5.1.

6.2.5.2.2 DELETING A CONTROLLER FROM THE DISC DRIVER TABLE

After the lesser number of controllers has been entered, a list of previously defined controllers is displayed as shown in Figure 6-15. SETUP then prompts for the controller number(s) to be deleted, one at a time.

Enter the number of the controller that is to be deleted. SETUP deletes the specified controller and redisplays the list of controllers indicating which was deleted with a DELETED message. SETUP continues to prompt for a controller to be deleted until the total number of controllers corresponds to the total specified. The control file is then updated automatically.

A user may press <ESC> to abort this module and return to the Control File Maintenance menu.

```
PORT n              CONFIGURE DISC DRIVER TABLE         SU114A n.n  mm/dd/yy

   LOGICAL                 DEVICE        NUMBER OF
  CONTROLLER                CODE         PARTITIONS

      0                      33              4
      1                      27              3











COMMENT:  ENTER CONTROLLER NUMBER TO DELETE
COMMAND:
MESSAGE:
```

Figure 6-15. Controller Deletion Screen

6.2.5.2.3 ADDING A PARTITION

After a larger total number of partitions has been entered, the previously defined partitions are displayed as defaults. An added partition number is displayed and the cursor is positioned at the Drive No. prompt (see Figure 6-13). Enter the appropriate parameters as described in Section 6.2.5.1. When the number of partitions matches the total number of partitions specified, SETUP displays

    (M)ODIFY, (D)ISPLAY, (S)AVE

The previously defined partitions may then be modified as described in Section 6.2.5.2. The new configuration must then be saved into the control file.

6.2.5.2.4 DELETING A PARTITION

After a smaller number of partitions have been specified, SETUP deletes the appropriate number of partitions from the previously defined configuration by truncation.

For example, if a controller was configured for four partitions (e.g., 1.0, 1.1, 1.2, and 1.3) and two partitions are to be deleted, SETUP deletes partition numbers 1.2 and 1.3.

SETUP then displays the remaining partitions with their previously defined parameters as defaults. The following message is then displayed:

    (M)ODIFY, (D)ISPLAY, (S)AVE

Modify the partition entries as described in Section 6.2.5.2 or enter S to save the configuration into the control file.

6.2.6 LISTING THE CONTROL FILE

Portions of the control file, such as the System Information table, Port Definition table, etc., may be displayed at the terminal or sent to a printer.

At the main menu, select option 2 and the screen shown in Figure 6-16 is displayed:

```
PORT n              LIST THE CONFIGURATION CONTROL FILE      SU2 n.n  mm/dd/yy


                CONTROL FILE NAME:














COMMENT:
COMMAND:
MESSAGE:
```

Figure 6-16. List Control File Screen

Enter the name of the control file and a list option screen as shown in Figure 6-17 is displayed.

```
PORT n              LIST THE CONFIGURATION CONTROL FILE      SU2 n.n  mm/dd/yy



                    (0) RETURN TO SYSTEM CONFIGURATION MENU

                    (1) LIST SYSTEM INFORMATION TABLE

                    (2) LIST PORT DEFINITION TABLE

                    (3) LIST DISCSUB TABLE

                    (4) LIST DISC DRIVER TABLE



COMMENT:  ENTER THE NUMBER OF THE FUNCTION YOU WISH TO EXECUTE
COMMAND:
MESSAGE:
```

Figure 6-17. List Option Screen

Select the listing of your choice. A screen naming the portion of the control file to be listed and the following message are then displayed:

OUTPUT REPORT TO (P)RINTER OR (S)CREEN?

The default is output to the screen. Press <RETURN> to display the list on the screen. To scroll the screen, press <RETURN>. To exit the display, press <ESC>. The program redisplays the prompt for a control file name. Enter the name of another control file or press <ESC>, the main menu is then displayed.

Enter P to send a listing to the printer. The following prompt is then displayed:

ENTER DEVICE NAME

$LPT is the default. To accept the default, press <RETURN>; otherwise specify the appropriate device name.

When the list has been printed, the main menu is displayed.

6.3 UPDATING THE SYSTEM

The system information tables (i.e., the CONFIG file) and the driver files are not updated until the parameters for the various tables have been entered into the configuration control file and the update function of SETUP is executed. Two requirements should be met before executing the update function:

- Make sure that the parameters entered into the tables of the control file are correct. This may be done by listing each table either on the terminal or printing it out and then checking the various parameters. Refer to Section 6.2.6 for information on listing the control file.
- Make a backup copy of the current system. Once the update function is executed, the system configuration is updated. An error in a parameter entry may make it impossible to IPL the newly configured system.

To update the system files with the configuration stored in the control file, select option 3 at the main menu. SETUP displays the following message on the comment line:

DO YOU HAVE A BACKUP?

Notice that the default for this question is no. This is to make sure that pressing <RETURN> accidentally will not result in a faulty system update.

If <RETURN> is pressed or N is entered, SETUP displays the following message:

BACK UP YOUR SYSTEM BEFORE UPDATING

The main menu is displayed and the user may then exit the SETUP utility to perform the required backup.

If a backup copy of the system has been made, enter Y. SETUP then asks for the control file name. Enter the name and an option screen is displayed as shown in Figure 6-18.

Each time an option has been executed, the update option screen is redisplayed. The sequence for updating the various system tables and the driver file is not important. To exit this module, select option 0 or press <ESC>.

After the system configuration has been updated, IPL the system to allow SIR to set pointers, bring the correct drivers into memory, etc.

```
PORT n                 UPDATE THE SYSTEM FILES                 SU3 n.n  mm/dd/yy



                   (0) RETURN TO SYSTEM CONFIGURATION MENU

                   (1) UPDATE SYSTEM INFORMATION TABLE

                   (2) UPDATE PORT DEFINITION TABLE

                   (3) UPDATE DISCSUB TABLE

                   (4) UPDATE DISC DRIVER TABLE



COMMENT:  ENTER THE NUMBER OF THE FUNCTION YOU WISH TO EXECUTE
COMMAND:
MESSAGE:
```

Figure 6-18. System Update Option Screen

6.4 SU.ENTRIES FILE MAINTENANCE

The SETUP utility uses a parameter file called SU.ENTRIES for the disc driver table module. It is a formatted file which contains information from the disc specification sheets in the R8 Peripherals Handbook. When an IRIS Operating System is first delivered, this file contains up-to-date information. However, disc controllers and driver specifications change from time-to-time. POINT 4 then releases changed pages for the R8 Peripherals Handbook. Such changes should be entered into the SU.ENTRIES file so that the SETUP utility functions properly when a system is reconfigured. An example of a disc specification sheet is shown in Figure 6-14.

SETUP provides IRIS users with an easy method for updating the SU.ENTRIES file with option 4 of the main menu.

Selection of option 4 from the main menu (see Figure 6-1) causes the Entry File Maintenance options to be displayed as shown in Figure 6-19.

```
PORT n                    ENTRY FILE MAINTENANCE                SU4 n.n  mm/dd/yy



                   (0) RETURN TO SYSTEM CONFIGURATION MENU

                   (1) DRIVES ENTRY FILE MAINTENANCE

                   (2) LIST THE DRIVES ENTRY FILE



COMMENT:  ENTER THE NUMBER OF THE FUNCTION YOU WISH TO EXECUTE
COMMAND:
MESSAGE:
```

Figure 6-19. Entry File Maintenance Options

Selection 1 from the option menu displays the Disc Drives Entry Maintenance screen as shown in Figure 6-20.

```
PORT n              DISC DRIVES ENTRY MAINTENANCE           SU41 n.n  mm/dd/yy
                                          ENTRY NO.:
                                        DEVICE CODE:
                             NO. OF CYLINDERS IN LU/0:
                          MAXIMUM CYLINDERS OTHER LUS:
NUMBER OF PHYSICAL TRACKS PER CYLINDER (NPTC):
                                DISC FLAG WORD (DFLG):
                         NUMBER OF IRIS TRACKS (NTRS):
                            PHYSICAL UNIT CODE (PHYU):


COMMENT:
COMMAND:
MESSAGE:
```

Figure 6-20. Disc Drives Entry Maintenance Screen

ENTRY NO. - Enter the disc specification sheet's entry number as given in the top right corner of the disc specification sheet.

DEVICE CODE - Enter the value given for the device code listed at the left side of the disc specification sheet.

NO. OF CYLINDERS IN LU/0 - Enter the value given for the number of cylinders in LU/0.

MAXIMUM CYLINDERS OTHER LUs - Enter the value given for the maximum number of cylinders for other LUs.

NUMBER OF PHYSICAL TRACKS PER CYLINDER (NPTC) - Enter the value given for NPTC listed at the left side of the disc specification sheet.

DISC FLAG WORD (DFLG) - Enter the value given for DFLG listed at the left side of the disc specification sheet.

NUMBER OF IRIS TRACKS (NTRS) - Enter the value given for NTRS listed at the left side of the disc specification sheet.

PHYSICAL UNIT CODE (PHYU) - On the disc specification sheet, the value for PHYU is specified as an expression where D = drive unit number and P = platter or surface. The PHYU code for the SU.ENTRIES file is as follows:

PHYU Code	Description
0	Enter 0 when the value for PHYU shown on the disc specification sheet is given for the drive number only (e.g., either D is the only parameter, D * an octal value, or D + an octal value)
1	Enter 1 when the value for PHYU shown on the disc specification sheet is D + P and P = 0 or 1 (e.g., (D * 40000) + (P * 1000) usually P=0 for removable, P=1 for fixed)
2	Enter 2 if the drive includes fixed discs and the value for PHYU shown on the disc specification sheet gives one or more specifications for fixed surface P. For example, the calculation for PHYU may be shown as follows: (10*P) + D + (100000 if fixed) where D = drive unit no. P = platter or surface 32MB - P=0 remov; P=0 fixed 64MB - P=0 remov; P=0,1,or 2 fixed 96MB - P=0 remov; P=0,1,2,3 or 4 fixed

6.4.1 DELETING AN EXISTING RECORD

The cursor is positioned at the first parameter. Enter the disc specification entry number. This is found at the top right corner of the disc specification sheet (see Figure 6-14). If the SU.ENTRIES file contains a record for that entry number, SETUP displays those values for the required parameters. At the comment line the following message is displayed:

DELETE THIS RECORD?

To delete the record, enter Y. The record is deleted from the SU.ENTRIES file and the disc drive maintenance screen is redisplayed.

Enter another disc specification entry number to continue the deletion procedure.

Press <ESC> to return to the Entry File Maintenance Option menu.

6.4.2 MODIFYING AN EXISTING RECORD

The cursor is positioned at the first parameter. Enter the disc specification entry number. This is found at the top right corner of the disc specification sheet (see Figure 6-14). If the SU.ENTRIES file contains a record for that entry number, SETUP displays those values for the required parameters. At the comment line the following message is displayed:

DELETE THIS RECORD?

To modify a record, press <RETURN> (the default is no). The previous parameters are erased from the screen and the cursor rests at the device code prompt. If the same device code is entered, the original parameters are displayed as defaults. If the device code is changed, enter the new parameters from the new disc specification sheet supplied by POINT 4. Press <RETURN> for any parameters that do not require change.

When the last parameter has been entered, SETUP asks

IS ALL THE ABOVE CORRECT?

If one or more parameters are incorrect, enter N. The cursor moves to the last prompt on the screen. Press <ESC> to back up the cursor to the incorrect parameter. Enter the correct value. Repeat for any other incorrect parameters. Then press <RETURN> at each correct parameter until the program repeats the 'all correct' message.

If all parameters are correct, press <RETURN> (the default is yes). The record is then written into the SU.ENTRIES file and the program redisplays the Disc Drive Maintenance screen (see Figure 6-20).

If no further modifications are required, press <ESC> to return to the Entry Options menu.

6.4.3 ADDING A RECORD

A new disc specification record may be added to the SU.ENTRIES file by entering a new disc specification entry number at the first parameter prompt. Enter the required parameters from the new disc spec sheet supplied by POINT 4.

6.4.4 LISTING THE SU.ENTRIES FILE

To display the SU.ENTRIES file at the terminal or send it to a printer, select option 2 from the Entry File Maintenance Option menu (see Figure 6-19). SETUP then displays the following message:

    OUTPUT REPORT TO (P)RINTER OR (S)CREEN

The default outputs to the screen. To display the file at the terminal, enter S or press <RETURN>. The file is displayed in the format shown in Figure 6-21. Press <RETURN> to scroll the display. Press <ESC> to exit the display and return to the Entry File Maintenance Option menu.

To output the contents of the SU.ENTRIES file to a line printer, enter P at the command line. SETUP then displays the following message:

    ENTER THE DEVICE NAME

The default is $LPT (the system printer). To print the list on the system printer, press <RETURN>. Otherwise specify the appropriate device name.

The format of the report is shown in Figure 6-21. When printing is completed, the Entry File Maintenance Option menu is redisplayed.

DATE TIME R8 DISC SPECIFICATION PAGE n

ENTRY NO.	DEVICE CODE	NO. CYLS IN LU/0	MAX. CYLS OTHER LUS	NPTC	DFLG	NTRS	PHYU
=====	======	========	=========	====	======	====	====
1	33	200	626	2	100500	214	1
5	30	200	626	2	104500	214	1
36	27	140	1462	1	500	220	2
360	52	115	115	2	500	217	0

END OF DATA

Figure 6-21. Sample SU.ENTRIES File Report

APPENDICES

Appendix A
IRIS COMPONENT CHECKLISTS

This appendix contains two tables listing IRIS components. Table A-1 is a list of IRIS components on LU/0 and their file types. Table A-2 describes the IRIS components on LU/5.

TABLE A-1. IRIS COMPONENTS ON LU/0

Name	File Type	Name	File Type
$CALLTBL	77001	FOREIGN	36
$DEC	77001	FORMAT	33401
$MMUX	77001	GUIDE	77002
$SYS.SCHED	77001	GUIDE.BLKCOPY	77002
ACCOUNTS	77031	GUIDE.LPT	77002
ACCOUNTUTILITY	77002	GUIDE.LU	77002
ACTUTIL.1	77002	INDEX	33000
ALOAD	77002	INSTALL	33401
ASM	33401	KILL	33401
BAKUPCALL	77002	LCM*	77001
BAKUPINIT	77003	LCMACTIVATE*	77401
BAKUPMAIN	77003	LCMC*	77002
BASIC	33702	LCMC.1*	77002
BASICTEST	77002	LCMC.2*	77002
BCONVERT	33401	LCMC.3*	77002
BLOCKCOPY	77003	LCMCHECK*	77002
BYE	33400	LCMDIAG1.3*	77003
CHANGE	33401	LIBR	33401
CLEANUP	77401	LPTD*	36
CLEANUPX	77401	LPTM	36
COMA	36	LPTP*	36
COMD*	36	M2DISCUTILITY**	77003
CONFIG	77001	M2FLBOOT**	77003
COPY	33401	M2TAPEDISCDIAG**	77003
CTR	77036	M3DISCDIAG**	77003
CTUS	77001	M3FLOPPYDIAG**	77003
DC700*	77003	M3MUXDIAG**	77003
DEFS	33030	M3TAPEDIAG**	77003
DGMX*	77001	M8EXERCISER*	77003
DISCSUBS	77032	MAIL	33401
DISCUTILITY	77003	MAPACTIVATE*	77401
DMAP	77000	MAPCHECK*	77002
DSP	77400	MESSAGES	77001
EDIT	33401	MTA0	36
EIS*	77001	MTAS*	77001
FAULTHISTORY	77032	PHA	77001
FAULTPRINT	33401	PLOAD	33400
FLBOOT**	77003	PORT	33401

*Shipped with POINT 4 MARK 5/9 only
**Shipped with POINT 4 MARK 2/3 only

TABLE A-1. IRIS COMPONENTS ON LU/0 (Cont)

Name	File Type	Name	File Type
PROTECT	33401	SU2	77002
PTM*	36	SU21	77002
PTP*	36	SU22	77002
PTR*	36	SU23	77002
PZ	33030	SU24	77002
QUERY	33401	SU3	77002
R82BAKUPCNFG	77002	SU31	77002
REHASH	33401	SU32	77002
REMOVE	33401	SU33	77002
RESTOREINFO	77002	SU34	77002
RESTOREINFO.S	77030	SU4	77002
REX	77000	SU41	77002
RTC*	77001	SU42	77002
RUN	33602	SYMBOLS	33030
RUNMAT	33402	SYSMAP*	77001
SAVE	33401	TERM.ACT5	77001
SBU.CHECKP	77003	TERM.ADDS	77001
SBU.GETB	77003	TERM.ADDS25	77001
SBU.SUSPENDTS	77003	TERM.ADM1	77001
SCOPE	33400	TERM.ADM3	77001
SETTIME	33002	TERM.B100	77001
SETUP	77002	TERM.B4	77001
SHUTDOWN	33403	TERM.DGC	77001
STBOOTM3**	77003	TERM.DIAL80	77001
STBOOTM5*	77003	TERM.DM1520	77001
STREAMER*	77003	TERM.DM1521	77001
SU.DSUBS	77032	TERM.H1500	77001
SU.ENTRIES	77031	TERM.H2000	77001
SU1	77002	TERM.INET	77001
SU11	77002	TERM.TV912	77001
SU111	77002	TERM.TV950	77001
SU112	77002	TERM.VT100	77001
SU112A	77002	TERMS	77001
SU113	77002	TERM.WS100	77001
SU114	77002	TTY*	77001
SU114A	77002	USERID	77031
SU11A	77002	VERIFY	33601

*Shipped with POINT 4 MARK 5/9 only
**Shipped with POINT 4 MARK 2/3 only

TABLE A-2. IRIS COMPONENTS ON LU/5

Name	Description
ABASIC	Protected BASIC program
ABASIC.1	Protected BASIC program
ABASIC.2	Protected BASIC program
ABASIC.3	Protected BASIC program
ABASIC.4	Protected BASIC program
ABASIC.5	Protected BASIC program
ABASIC.A	Text file
ABASIC.B	Text file
ABASIC.XREF	Indexed contiguous file for ABASIC
ACCOUNTS	System file created when LU/5 is installed
ACS.VERIFY	Saved BASIC program
ACS.VERIFY2	Saved BASIC program
ACS.VERIFY3	Saved BASIC program
ACS.VERIFY4	Saved BASIC program
ANALYPF	Saved BASIC program
ASSIGNPF	Saved BASIC program
ASSIGNPF1	Saved BASIC program
ASSIGNPF2	Saved BASIC program
ASSIGNPF2A	Saved BASIC program
ASSIGNPF3	Saved BASIC program
BUILDPF	Saved BASIC program
BUILPFERR	Saved BASIC program
BUILDXF	Saved BASIC program
CHECKSUM	Saved BASIC program
COPYPF	Saved BASIC program
COPYPF1	Saved BASIC program
COPYPF2	Saved BASIC program
COPYPF2A	Saved BASIC program
COPYPF3	Saved BASIC program
COPYPF4	Saved BASIC program
COREMAP	Saved BASIC program
DISPLAY	Protected BASIC program

TABLE A-2. IRIS COMPONENTS ON LU/5 (Cont)

Name	Description
DMAP	System file created when LU/5 is installed
EXERCISER	Saved BASIC program
EXTRAPORT	Saved BASIC program
FILE.SERVE	Saved BASIC program
FINDFILE	Saved BASIC program
FORGE	Saved BASIC program
FORGE1	Saved BASIC program
FORGE2	Saved BASIC program
FORGE21	Saved BASIC program
FORGE22	Saved BASIC program
FORGE23	Saved BASIC program
FORGE3	Saved BASIC program
FORGE4	Saved BASIC program
GUARD	Guarded BASIC program
INDEX	System file created when LU/5 is installed
KILLPF	Saved BASIC program
MAGTAPE	Saved BASIC program
MAGTAPE.LOAD	Saved BASIC program
MAGTAPE1	Saved BASIC program
MAGTAPE11	Saved BASIC program
MAGTAPE2	Saved BASIC program
MAGTAPE21	Saved BASIC program
MAKEBIN	Guarded BASIC program
MAKEHEX	Guarded BASIC program
MONITOR	Guarded BASIC program
QUERYPF	Saved BASIC program
R7TOR8ACTCONV	Saved BASIC program
RECEIVE	Saved BASIC program
RENUMBER	Saved BASIC program
RENUMBER1	Saved BASIC program
RENUMBER2	Saved BASIC program

TABLE A-2. IRIS COMPONENTS ON LU/5 (Cont)

Name	Description
RENUMBER3	Saved BASIC program
RETRY	Saved BASIC program
SWAPTEST	Saved BASIC program
TRANSMIT	Saved BASIC program
U.CHANGE	Saved BASIC program
U.CHANGE1	Saved BASIC program
U.CONVERT	Saved BASIC program
U.CONVERT1	Saved BASIC program
U.COPY	Saved BASIC program
U.COPY1	Saved BASIC program
U.KILL	Saved BASIC program
U.KILL1	Saved BASIC program
U.PROTECT	Saved BASIC program
U.PROTECT1	Saved BASIC program
U.SAVE	Saved BASIC program
U.SAVE1	Saved BASIC program
XREF	Saved BASIC program
XREF1	Saved BASIC program
XREF2	Saved BASIC program
XREF3	Saved BASIC program
XREF4	Saved BASIC program
XREF5	Saved BASIC program
XREF6	Saved BASIC program
XREFA	Saved BASIC program
XREFB	Saved BASIC program

Appendix B
SOFTWARE DEFINITIONS

This appendix contains a listing of software definitions for IRIS R8.2.

```
ASM ,@$LPT,R82DEFSF,R82PZA
AUG  4, 1983  11:39:22

; SOFTWARE DEFINITIONS FOR "IRIS" R8.2
; LAST EDIT 11 JUNE 83 by GKW. (FOR R8.2')

;                    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 constants     p. 10-12 Discsubs
;     2 CONFIG file                    13-14 DISCSUBS file locations
;   3-4 INFO Table                     15-16 System subroutines, Accounts
;   5-7 PCB (Port Control Block)          17 File Header block
;     8 User partition, Task Queue        18 DATAPUMP
;     9 TCN (Task Control Node)        19-20 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
       100000 .DUSR PSDEVF=100000   ;PSEUDO DEVICE FLAG
       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

; MISCELLANEOUS SYSTEM DEFAULT PARAMETERS

          200 .DUSR DFNSUB=200   ;DEFAULT MAX SYSTEM DISCSUB # USED IN CONFIG AND REX
            4 .DUSR SZLT   =4    ;DEFAULT SIZE OF PSEUDO DEV. LINKAGE TABLE

           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
        43200  .DUSR MEPS     =43200    ;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
          377  .DUSR MAXBZUP=377        ;MAX SIZE OF A BZUP DRIVER
        77000  .DUSR LBTUP    =77000    ;LOCATION FOR BTUP

; DEFINE CONFIG FILE LAYOUT

;     0 -   277      ;RESERVED
;   300 -   377      ;INITIALIZATION TABLE, RESERVED FOR USE BY SIR.
;   400 -   577      ;GENERAL INFORMATION TABLE (PARTITION INFO)
;   600 -   777      ;SYSTEM INFORMATION TABLE (USER CONFIGURABLE PORTION)
;  1000 -  1177      ;MEMORY RESIDENT DISCSUB LIST
;  1200 -  1377      ;RESERVED
;  1400 -  2777      ;DISC DRIVER TABLE
;  3000 - 13377      ;RESERVED
;
; IF THE FOLLOWING TWO ADDRESS RANGES ARE CHANGED, THEN CHANGE CTUTILITY ALSO
;
; 13400 - 13477      ;NON MCT 802-902 IPL SEQUENCE
; 13500 - 13577      ;MCT 802-902 IPL SEQUENCE
;
; 13600 - 13777      ;RESERVED
; 14000 - 15777      ;DISC DRIVER ENTRY ADDRESSES BY PERIPHERAL HANDBOOK ENTRY
; 16000 - 16377      ;LOG-ON RESTRICTIONS TABLE
; 16400 - 17377      ;LOG-ON PROGRAM STARTUP TABLE
; 17400 - 17777      ;IPL PROGRAM STARTUP TABLE
; 20000 - 77777      ;DISC DRIVERS

        14000  .DUSR DRADDRTABLE = 14000;LOC. OF DISC DRIVER ADDRESS TABLE
```

```
; DEFINE A DISC CONTROLLER TABLE ENTRY IN CONFIG
;
                0 .DUSR ALFX = 0        ;MEMORY ADDRESS OF LUFIX (SET BY SIR)
                1 .DUSR VAFX = 1        ;FILE ADDRESS OF LUFIX (AKA R/W ENTRY) IN CONFIG
                2 .DUSR VABZ = 2        ;FILE ADDRESS OF BZUD IN CONFIG
                3 .DUSR NPAR = 3        ;NUMBER OF DISC PARTITIONS THIS DRIVE
                4 .DUSR DCDE = 4        ;DEVICE ADDRESS (CODE) FOR THIS CONTROLLER
                5 .DUSR RMNB = 5        ;RATIO FOR MINIMUM BLOCK COUNT

;WORDS 6 AND 7 ARE RESERVED FOR POINT 4 USE

               10 .DUSR SZDCTBL = 10   ;SIZE OF A DISC CONTROLLER TABLE ENTRY

; DEFINE A DISC PARTITION TABLE ENTRY IN CONFIG
;
                0 .DUSR ALVR = 0        ;MEMORY ADDRESS OF LUVAR (SET BY SIR)
                1 .DUSR CNPT = 1        ;NPTC - NUMBER OF PHYSICAL TRACKS PER CYLINDER
                2 .DUSR CDFG = 2        ;DFLG - DISC FLAG WORD (SEE LUVAR DEFINITIONS)
                3 .DUSR FUN  = 3        ;      - FOREIGN UNIT NUMBER
                4 .DUSR CFYU = 4        ;PHYU - PHYSICAL UNIT SELECT WORD
                5 .DUSR FSTC = 5        ;FCYL - FIRST CYLINDER
                6 .DUSR NCYS = 6        ;NCYL - NUMBER OF CYLINDERS
                7 .DUSR CNIT = 7        ;NTRS - NUMBER OF IRIS TRACKS

               10 .DUSR SZDPTBL = 10   ;SIZE OF A DISC PARTITION TABLE

; DEFINE SPECIAL ACCESS POINTS FOR A BZUD DISC DRIVER
;
                1 .DUSR DXPU = 1        ;BZUD'S PHYU
                2 .DUSR DXFC = 2        ;BZUD'S FCYL
                4 .DUSR DXRD = 4        ;BZUD'S READ ENTRY POINT
                7 .DUSR DXWR = 7        ;BZUD'S WRITE ENTRY POINT
```

```
; DEFINITION 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

; DEFINE SPECIAL $CTR WORDS

            16  .DUSR CTRVECTOR = 16 ;PAGE ZERO VECTOR FOR MONITOR HOOKS
             1  .DUSR CTIRDA    =  1 ;SYSTEM HOOK FROM SINT AT SWAPIN
             2  .DUSR CTECALL   =  2 ;SYSTEM HOOK FROM CALLD

; DEFINE GENERAL INFORMATION TABLE DISPLACEMENTS

             0  .DUSR PSIZE = 0      ;PARTITION SIZE
             1  .DUSR NPART = 1      ;NUMBER OF PARTITIONS
             2  .DUSR MTYPE = 2      ;MEMORY TYPE NUMBER

; 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  .DUSR UDSB. =12 ;MAXIMUM NUMBER OF USER WRITTEN DISCSUBS
            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 SDSB. =16 ;MAXIMUM NUMBER OF SYSTEM DISCSUBS
            17  .DUSR KTSL. =17 ;TIME SLICE PARAMS. LONG TIME SLICE * 400 + SHORT TIME SLICE
            20  .DUSR DFLU. =20 ;DEFAULT LU # (FOR CHAIN AND SCOPE FIND FILE SEARCHES)
                                ;.DUSR      =21 ;RESERVED
            22  .DUSR SZLNK=22 ;SIZE OF PSEUDO-DEVICE LINKAGE TABLE

            32  .DUSR SZICON=32;SIZE OF AREA OF INFO DOWNLOADED FROM CONFIG

; (CONTINUED ON NEXT PAGE)
```

```
; THIS PORTION OF THE INFO TABLE DOES NOT EXIST IN THE CONFIG FILE

        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 .DUSR LINKT=76  ;POINTER TO PSEUDO DEVICE LINKAGE TABLE
        77 .DUSR LCALT=77  ;ACTUAL LOCATION OF CALL TABLE (-1 ==> NONE)
```

```
; THE CALL TABLE MAY BE ACCESSED VIA ROUTINES AT THE FOLLOWING OFFSETS
; FROM THE ADDRESS OF THE CALL TABLE

        177777 .DUSR LCFNBR. =-1 ;OFFSET FROM CALL TABLE TO ROUTINE TO RETURN THE DISCSUB
                                  ;   NUMBER ASSOCIATED WITH THE "CALL" SUBROUTINE NUMBER
        177776 .DUSR LCFNM.  =-2 ;OFFSET FROM CALL TABLE TO ROUTINE TO RETURN THE DISCSUB
                                  ;   NUMBER ASSOCIATED WITH THE "CALL" SUBROUTINE NAME
                 ;      100-105 ;(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

; CURRENTLY DEFINED PSEUDO DEVICE NUMBERS

             1 .DUSR CTDN =1   ;CASSETTE TAPE PSEUDO DEV. #1
```

```
; 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

; NOTE: URA IS SET TO SPECIAL ENTRY POINT WHEN APPROPRIATE
;       (I.E  CNTL-C OR ESCAPE ENTRIES)

;                   .DUSR ORA.  =15 ;OLD RETURN ADDRESS (RESERVED FOR POINT 4 USE)
                16  .DUSR RUA.  =16 ;CPU TENTH-SECONDS USED SINCE LOG-ON (LSB'S)
                17  .DUSR ABN.  =17 ;ABNORMAL TERMINATOR INDICATOR
                                ;   bit 15-2 RESERVED
                                ;   bit 1     CNTRL-C
                                ;   bit 0     ESCAPE
                20  .DUSR RUI.  =20 ;CPU TENTH-SECONDS USED SINCE LOG-ON (MSB'S)
                21  .DUSR AOI.  =21 ;AGE OF INTERACTION
                22  .DUSR CTC.  =22 ;CRITICAL TIME COUNTER
                23  .DUSR PF1.  =23 ;APRI*1000 + EFFECTIVE PRIORITY
                24  .DUSR PF2.  =24 ;POLICY FUNCTION WORD #2
                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)
```

```
-- PAGE 8 -

; 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
```

```
; EACH BIT IN PCW IS A FLAG AS FOLLOWS:

;  BIT*    MEANING
;   15_  0
;   14‾  THIS PORT IS ON A POINT 4 MIGHTY-MUX
;   13   0
;   12_  DEVICE CONTROL OUTPUT (1 = HIGH, 0 = LOW)
;   11‾  NORMAL DEVICE STATUS INPUT (1 = HIGH, 0 = 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 <CLTR>
;    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    }     (0 TO 17 OCTAL)
;    0_  /

; * NOTE:  BIT 15 IS THE MOST SIGNIFICANT BIT
```

```
; 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‾  FILE IS OPENED IN MAINTAINCE MODE
;   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

; VALUES FOR MUX "SEND" ROUTINES.  SET UP IN A0 BEFORE CALLING SEND.

          177777 . DUSR MXSO =-1   ;MUX'S START OUTPUT ROUTINE
          177776 . DUSR MXSI =-2   ;MUX'S START INPUT ROUTINE
          177775 . DUSR MXTA =-3   ;MUX'S TERMINATE ALL I/O ROUTINE
          177774 . DUSR MXPW =-4   ;MUX'S SEND PCW ROUTINE
          177773 . DUSR MXSN =-5   ;MUX'S START OUTPUT NO SPL  CHAR. ROUTINE
          177772 . DUSR MXSS =-6   ;MUX'S START INPUT SNGL  CHAR. ROUTINE
          177771 . DUSR MXTO =-7   ;MUX'S TERMINATE OUTPUT ROUTINE
          177770 . DUSR MXCO =-10  ;MUX'S CONVERT OCW TO MK5 FORMAT ROUTINE
          177767 . DUSR MXCI =-11  ;MUX'S CONVERT ICW TO MK5 FORMAT ROUTINE
```

```
-- PAGE 11

; PARTITION INFORMATION TABLE DISPLACEMENTS
; TABLE HEADER

                0 .DUSR NPT.  = 0   ;NUMBER OF PARTITIONS
                1 .DUSR MP0.  = 1   ;USER MAP 0                    (1)
                2 .DUSR MP1.  = 2   ;USER MAP 1                    (1)
                3 .DUSR MP2.  = 3   ;USER MAP 2                    (1)

                4 .DUSR PIHS = MP2.+1   ;PARTITION INFORMATION TABLE HEADER SIZE

; PARTITION INFORMATION BLOCK

                0 .DUSR PAD.  = 0   ;PARTITION ADDRESS
                1 .DUSR SZP.  = 1   ;PARTITION SIZE
                2 .DUSR JCP.  = 2   ;JOB CONTROL BLOCK POINTER
                3 .DUSR AFP.  = 3   ;ACTIVE FILE HEADER BUFFER
                4 .DUSR TLU.  = 4   ;TIME LAST USED
                5 .DUSR LBP.  = 5   ;LOGICAL BASE PAGE             (1)
                6 .DUSR PBP.  = 6   ;PHYSICAL BASE PAGE            (1)
                7 .DUSR NPG.  = 7   ;NUMBER OF PAGES               (1)

               10 .DUSR PIBS = NPG.+1   ;PARTITION INFORMATION BLOCK SIZE

;(1) USED BY $SYS.MAP

; 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 MODE (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
```

```
; 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
```

```
; 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.
          2000  .DUSR U =02000 ;DISCSUB IS FROM THE "DISCSUBS.USER" FILE

; DISCSUB NUMBERS FOR "CALL"

         10000  .DUSR FAULT = 0+D    ;PRINT TRAP MESSAGE, ABORT TASK
             1  .DUSR ALLOC = 1      ;ALLOCATE DISC BLOCKS
             2  .DUSR DALLC = 2      ;DEALLOCATE DISC BLOCKS
             3  .DUSR FFILE = 3      ;FIND FILE IN INDEX
             4  .DUSR EXTEN = 4      ;CHANGE TO EXTENDED FILE
             5  .DUSR ALCON = 5      ;ALLOCATE A CONTIGUOUS FILE
             6  .DUSR CDTA  = 6      ;CONVERT DRATSAB TO ASCII
             7  .DUSR CIA   = 7      ;CONVERT INTEGER TO ASCII (ANY RADIX)
            10  .DUSR CSTR  =10      ;COMPARE STRINGS
            11  .DUSR PASSC =11      ;PASSWORD COMPARE
            12  .DUSR ERROR =12      ;ERROR ROUTINE FOR BASIC
            13  .DUSR MESSA =13      ;CANNED MESSAGE TO I/O BUFFER
            14  .DUSR BREAK =14      ;BREAKPOINT SETUP FOR DSP
            15  .DUSR ACNTL =15      ;ACCOUNT LOOKUP
            16  .DUSR DELET =16      ;DELETE A FILE
         20017  .DUSR PDELE =17+N    ;DELETE A PROCESSOR OR DRIVER
         40020  .DUSR BUILD =20+X    ;BUILD A NEW FILE
         60021  .DUSR BILDD =21+N+X  ;BUILD A "$" FILE
         40022  .DUSR OPEN  =22+X    ;OPEN A FILE OR A DEVICE
         60023  .DUSR OPENU =23+N+X  ;OPEN A FILE OR A DEVICE FOR UPDATE
         60024  .DUSR OPENL =24+N+X  ;OPEN AND LOCK A FILE OR A DEVICE
         60025  .DUSR OPENR =25+N+X  ;OPEN A FILE OR A DEVICE FOR REFERENCE
            26  .DUSR CLOSE =26      ;CLOSE A CHANNEL
            27  .DUSR CLEAR =27      ;CLEAR A CHANNEL
         40030  .DUSR GETRR =30+X    ;GET RECORD FOR READ
         60031  .DUSR GETRW =31+N+X  ;GET RECORD FOR WRITE
            32  .DUSR FINDI =32      ;FIND AN ITEM (NOT IMPLEMENTED)
            33  .DUSR READI =33      ;READ AN ITEM
         20034  .DUSR WRITI =34+N    ;WRITE AN ITEM
            35  .DUSR WRITN =35      ;WRITE A NEW ITEM
            36  .DUSR READC =36      ;READ FROM CONTIGUOUS FILE
         20037  .DUSR WRITC =37+N    ;WRITE INTO CONTIGUOUS FILE
            40  .DUSR CHARG =40      ;CHARGE FOR FILE ACCESS
            41  .DUSR SYSCO =41      ;TRANSMIT A SYSTEM COMMAND (CALL 98)
            42  .DUSR CNVDA =42      ;CONVERT DATE TO ASCII
            43  .DUSR CNVAD =43      ;CONVERT ASCII TO DATE
            44  .DUSR CNVDT =44      ;CONVERT DATE AND TIME (CALL 99)
            45  .DUSR RDFHI =45      ;READ FILE HEADER INFO (CALL 97)
            46  .DUSR SPECI =46      ;SPECIAL FUNCTIONS
         10047  .DUSR RECOV =47+D    ;RECOVER FROM A STALL OR CRASH
            50  .DUSR PATNF =50      ;PSEUDO DIVIDE ARC TANGENT FUNCTION
            51  .DUSR PLOGF =51      ;PSEUDO DIVIDE NATURAL LOG FUNCTION
            52  .DUSR PSQRF =52      ;PSEUDO DIVIDE SQUARE ROOT FUNCTION

; (CONTINUED ON NEXT PAGE)
```

```
; DISCSUB NUMBERS (CONTINUED)

          40053 .DUSR PEXPF =53+X     ;PSEUDO D
          40054 .DUSR PSINF =54+X     ;PSEUDO DIVIDE SINE FUNCTION
          60055 .DUSR PCOSF =55+N+X   ;PSEUDO DIVIDE COSINE FUNCTION
             56 .DUSR PTANF =56       ;PSEUDO DIVIDE TANGENT FUNCTION
             57 .DUSR SIGPA =57       ;SIGNAL OR PAUSE
             60 .DUSR DIREC =60       ;SET UP DIRECTORIES FOR INDEXED FILE
          40061 .DUSR SEARC =61+X     ;SEARCH INDEXED FILE DIRECTORY
             62 .DUSR SHUFF =62       ;SHUFFLE DIRECTORY BLOCKS
             63 .DUSR DEKEY =63       ;DELETE KEY FROM DIRECTORY
          20064 .DUSR RELEA =64+N     ;RELEASE A DIRECTORY BLOCK
             65 .DUSR FIXDI =65     ;FIX DIRECTORIES OF MOVED INDEXED FILE
          40066 .DUSR REOPT =66+X     ;RE-OPTIMIZE INDEXED FILE DIRECTORY
             67 .DUSR AFSET =67       ;SET UP ACTIVE FILE FOR SWAP-OUT
             71 .DUSR MRC3  =71       ;MAG TAPE READ STATUS
             72 .DUSR MTASK =72       ;MAG TAPE SUPPLEMENTARY TASKS
             73 .DUSR MRFHD =73       ;MAG TAPE READ FILE HEADER
             74 .DUSR MRFIL =74       ;MAG TAPE READ INPUT FILE
             75 .DUSR MTFPE =75       ;MAG TAPE READ/WRITE TRANSFERS
          20076 .DUSR MNEXT =76+N     ;MAG TAPE GO TO NEXT DRIVE
             77 .DUSR MTAPA =77       ;MAG TAPE ALL OTHER FUNCTIONS
          40100 .DUSR LINKP =100+X    ;LINK PROGRAMS (BASIC'S "CHAIN")
          60101 .DUSR LOADP =101+N+X;LOAD A BASIC PROGRAM
          40102 .DUSR CTNXT =102+X    ;CTU POST-PROCESSING TASK
            103 .DUSR CTUSR =103      ;CTU DIR SEARCH ROUTINE
            104 .DUSR CTUWE =104      ;CTU WRITE DIR ENTRY ROUTINE
            105 .DUSR ST105 =105      ;STYLUS CALL(89)
          20106 .DUSR ST106 =106+N    ;FIND A SUBSTRING IN A STRING
            107 .DUSR ST107 =107      ;REVERSE A STRING
            110 .DUSR FINDF =110      ;FIND A FILE (CALL 96)
            111 .DUSR RDISC =111      ;READ OR WRITE WORD TO DISC (CALL 95)
            112 .DUSR CHFLT =112      ;CHANGE FILE TYPE (CALL 94)
            113 .DUSR WRWRD =113      ;WRITE WORD TO CORE (CALL 93)
            114 .DUSR OPENM =114      ;OPEN A FILE IN MAINTENANCE MODE
            115 .DUSR READM =115      ;READ A FILE IN MAINTENANCE MODE
          20116 .DUSR WRITM =116+N    ;WRITE A FILE IN MAINTENANCE MODE
            117 .DUSR RWMBIN=117      ;READ(WRITE)M non-BCD processing

; (CONTINUED ON NEXT PAGE)
```

```
; DISCSUB NUMBERS (CONTINUED)

        40120 .DUSR MODE0 =120+X   ;POLYMODE MODE 0
          121 .DUSR MODE1 =121     ;POLYMODE MODE 1
        40122 .DUSR PFSEA =122+X   ;POLYMODE SEARCH (MODES 2 - 5)
          123 .DUSR MODE4 =123     ;POLYMODE 4 (CALLED FROM PFSEA)
          124 .DUSR PRCOM =124     ;POLYFILE COMBINE BLOCKS
          125 .DUSR PFSHF =125     ;POLYMODE 4 KEY SHUFFLE
          126 .DUSR PFSHX =126     ;POLYMODE 4 KEY SHUFFLE(FOR BLK SPLIT)
          127 .DUSR MODE5 =127     ;POLYMODE 5 (CALLED FROM PFSEA)
          130 .DUSR PFABL =130     ;POLYFILE ALLOCATE DIRECTORY BLK
        20131 .DUSR PFALL =131+N   ;POLYFILE ALLOCATE (FROM BIT MAP)
          132 .DUSR PFRLS =132     ;POLYFILE RELEASE
          133 .DUSR PFSCN =133     ;POLYFILE SCAN
          134 .DUSR VOLRE =134     ;POLYFILE READ FROM VOLUME
          135 .DUSR MAPBU =135     ;POLYFILE MAP BUILD
          136 .DUSR DIRFN =136     ;POLYFILE FIND DIRECTORY
          137 .DUSR SZMAP =137     ;POLYFILE COMPUTE BIT MAP SIZE
          140 .DUSR DATCK =140     ;POLYFILE VALIDATE DATA VOL. DEFINIT
          141 .DUSR OPENP =141     ;POLYFILE OPEN (CALLED BY other OPEN)
        40142 .DUSR READP =142+X   ;POLYFILE READ
        60143 .DUSR WRITP =143+X+N;POLYFILE WRITE
        40144 .DUSR CALLP =144+X   ;POLYFILE BUILD
          145 .DUSR JULIA =145     ;JULIAN DATE ROUTINE
        40146 .DUSR CLPY1 =146+X   ;POLYFILE BUILD EXTENTION
          147 .DUSR PPWR  =147     ;
          150 .DUSR PRAND =150     ;PSEUDO RANDOM NUMBER GENERATOR
        40151 .DUSR ST151 =151+X   ;STYLUS' EDIT AND JUSTIFY
          152 .DUSR ST152 =152     ;STYLUS' PUSH OR POP A CHARACTER ARRAY
          153 .DUSR ST153 =153     ;STYLUS' CALL 56
          154 .DUSR ST154 =154     ;STYLUS' UNDERLINE
          155 .DUSR ST155 =155     ;STYLUS' PORT BUFFER LENGTH ADJ
          156 .DUSR ST156 =156     ;STYLUS' JUSTIFICATION CALCULATOR
          157 .DUSR ST157 =157     ;STYLUS' STRING LOCATOR
          160 .DUSR TIPO1 =160     ;TYPIST'S CALL 81
          161 .DUSR TIPO2 =161     ;TYPIST'S CALL 82
          162 .DUSR TIPO3 =162     ;TYPIST'S CALL 83
          163 .DUSR TIPO4 =163     ;TYPIST'S CALL 84
          164 .DUSR TIPO5 =164     ;TYPIST'S CALL 85
          165 .DUSR TIPO6 =165     ;TYPIST'S CALL 86
          166 .DUSR ATOE = 166     ;ASCII TO EBCDIC
          167 .DUSR ETOA = 167     ;EBCDIC TO ASCII
          170 .DUSR LOGIC =170     ;BASIC CALLABLE LOGIC OPERATIONS
          171 .DUSR STRING =171    ;BASIC CALLABLE STRING OPERATIONS
          172 .DUSR DOOMC =172     ;CHECK THE "DOOM" BITS
          173 .DUSR BAKU  =173     ;BAKUP DISCSUB

; 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)
```

```
; DEFINE FILE LOCATIONS FOR DISCSUBS

      400   DUSR LDSB1=400          ;"DISCSUBS" GROUP #1

      400   DUSR LREDIT = LDSB1    ;FILE LOC FOR READITEM
     1000   DUSR LBUILD = READI&X/X*400+400+LREDIT;FILE LOC FOR BUILD
     2000   DUSR LOPEN  = BUILD&X/X*400+400+LBUILD;FILE LOC FOR OPEN DISCSUBS
     3000   DUSR LCLEAR = OPEN&X/X*400+400+LOPEN;FILE LOC FOR CLEAR
     3400   DUSR LGETRR = CLEAR&X/X*400+400+LCLEAR;FILE LOC FOR GET RECORD
     4400   DUSR LFFILE = GETRR&X/X*400+400+LGETRR;FILE LOC FOR FIND FILE
     5000   DUSR LDALLC = FFILE&X/X*400+400+LFFILE;FILE LOC FOR DE-ALLOCATE
     5400   DUSR LALLOC = DALLC&X/X*400+400+LDALLC;FILE LOC FOR ALLOCATE
     6000   DUSR LDELET = ALLOC&X/X*400+400+LALLOC;FILE LOC FOR DELETE
     6400   DUSR LACNTL = DELET&X/X*400+400+LDELET;FILE LOC FOR ACCOUNT LOOKUP
     7000   DUSR LBREAK = ACNTL&X/X*400+400+LACNTL;FILE LOC FOR BREAKPOINT
     7400   DUSR LAFSET = BREAK&X/X*400+400+LBREAK;FILE LOC FOR ACTIVE FILE SETUP
    10000   DUSR LCNVAD = AFSET&X/X*400+400+LAFSET;FILE LOC FOR CONVERT ASCII TO DATE
    10400   DUSR LCNVDT = CNVAD&X/X*400+400+LCNVAD;FILE LOC FOR CONVERT DATE TO ASCII
    11000   DUSR LCIA   = CNVDT&X/X*400+400+LCNVDT;FILE LOC FOR CONVERT INTEGER TO ASCII
    11400   DUSR LFAULT = CIA&X/X*400+400+LCIA;FILE LOC FOR TRAPFAULT
    12000   DUSR LCLOSE = FAULT&X/X*400+400+LFAULT;FILE LOC FOR CLOSE
    12400   DUSR LLINKP = CLOSE&X/X*400+400+LCLOSE;FILE LOC FOR LINK(LOAD) PROGRAMS
    13400   DUSR LOPNPO = LINKP&X/X*400+400+LLINKP;FILE LOC FOR OPEN POLYFILE
    14000   DUSR LOPNMA = OPENP&X/X*400+400+LOPNPO;FILE LOC FOR OPEN MAINTAINANCE
    14400   DUSR LREDMA = OPENM&X/X*400+400+LOPNMA;FILE LOC FOR READMAINTAINCE
    15000   DUSR LRWMBIN= READM&X/X*400+400+LREDMA ;FILE LOC FOR READM BINARY
    15400   DUSR LDSE1  = RWMBIN&X/X*400+400+LRWMBIN;END OF "DISCSUBS" GROUP #1

    16400   DUSR LDSB2=2*400+LDSE1;"DISCSUBS" GROUP #2

    16400   DUSR LPSQRF = LDSB2   ;FILE LOC FOR SQUARE ROOT
    17000   DUSR LERROR = PSQRF&X/X*400+400+LPSQRF;FILE LOC FOR ERROR
    17400   DUSR LPLOGF = ERROR&X/X*400+400+LERROR;FILE LOC FOR LOG
    20000   DUSR LPEXPF = PLOGF&X/X*400+400+LPLOGF;FILE LOC FOR EXP
    21000   DUSR LPTANF = PEXPF&X/X*400+400+LPEXPF;FILE LOC FOR TAN
    21400   DUSR LPSINF = PTANF&X/X*400+400+LPTANF;FILE LOC FOR SIN
    22400   DUSR LRDFHI = PSINF&X/X*400+400+LPSINF;FILE LOC FOR READ F. H.
    23000   DUSR LSPECI = RDFHI&X/X*400+400+LRDFHI;FILE LOC FOR SPECIAL
    23400   DUSR LCHFLT = SPECI&X/X*400+400+LSPECI;FILE LOC FOR CHG. F. T.
    24000   DUSR LSYSCO = CHFLT&X/X*400+400+LCHFLT;FILE LOC FOR SYS. COMM
    24400   DUSR LRDISC = SYSCO&X/X*400+400+LSYSCO;FILE LOC FOR READ DISC
    25000   DUSR LLOGIC = RDISC&X/X*400+400+LRDISC;FILE LOC FOR BASIC LOGIC COMMANDS
    25400   DUSR LSTRING= LOGIC&X/X*400+400+LLOGIC;FILE LOC FOR BASIC STRING OPERATIONS
    26000   DUSR LPPWR  = STRING&X/X*400+400+LSTRING;FILE LOC FOR UP ARROW
    26400   DUSR LPRAND = PPWR&X/X*400+400+LPPWR;FILE LOC FOR PSEUDO-RANDOM GENERATOR
    27000   DUSR LDSE2  = PRAND&X/X*400+400+LPRAND;END OF "DISCSUBS" GROUP #2
```

```
30000  DUSR LDSB3=2*400+LDSE2; "DISCSUBS" GROUP #3

30000  DUSR LWRITN = LDSB3   ;FILE LOC FOR WRITE NEW
30400  DUSR LALCON = WRITN&X/X*400+400+LWRITN;FILE LOC FOR ALOC CONT
31000  DUSR LREDCO = ALCON&X/X*400+400+LALCON;FILE LOC FOR READ CONT
31400  DUSR LDIREC = READC&X/X*400+400+LREDCO;FILE LOC FOR DIRECTORY
32000  DUSR LSEARC = DIREC&X/X*400+400+LDIREC;FILE LOC FOR SEARCH
33000  DUSR LSHUFF = SEARC&X/X*400+400+LSEARC;FILE LOC FOR SHUFFLE
33400  DUSR LDEKEY = SHUFF&X/X*400+400+LSHUFF;FILE LOC FOR DEL. KEY
34000  DUSR LMRFIL = DEKEY&X/X*400+400+LDEKEY;FILE LOC FOR RD. TP. FILE
34400  DUSR LMTASK = MRFIL&X/X*400+400+LMRFIL;FILE LOC FOR TAPE TASK
35000  DUSR LMTAPA = MTASK&X/X*400+400+LMTASK;FILE LOC FOR TAPE MISC.
35400  DUSR LMTFPE = MTAPA&X/X*400+400+LMTAPA;FILE LOC FOR TP. FL. HNDL.
36000  DUSR LMRC3  = MTFPE&X/X*400+400+LMTFPE;FILE LOC FOR TP. EOF
36400  DUSR LMRFHD = MRC3&X/X*400+400+LMRC3;FILE LOC FOR FILE POST.
37000  DUSR LFIXDI = MRFHD&X/X*400+400+LMRFHD;FILE LOC FOR FIX DIRECTS
37400  DUSR LREOPT = FIXDI&X/X*400+400+LFIXDI ;FILE LOC FOR MODE 7
40400  DUSR LCTNXT = REOPT&X/X*400+400+LREOPT;FILE LOC FOR CTU NEXT
41400  DUSR LCTUNE = CTNXT&X/X*400+400+LCTNXT;FILE LOC FOR CTU ENTRY
42000  DUSR LDSE3  = CTUWE&X/X*400+400+LCTUWE;END OF "DISCSUBS" GROUP #3

43000  DUSR LDSB4=2*400+LDSE3; "DISCSUBS" GROUP #4

43000  DUSR LMDE0  = LDSB4   ;FILE LOC FOR POLYF MODE 0
44000  DUSR LMDE1  = MODE0&X/X*400+400+LMDE0;FILE LOC FOR POLYF MODE 1
44400  DUSR LPFSEA = MODE1&X/X*400+400+LMDE1;FILE LOC FOR POLYF SEARCH
45400  DUSR LMDE4  = PFSEA&X/X*400+400+LPFSEA;FILE LOC FOR POLYF MODE 4
46000  DUSR LPFSHF = MODE4&X/X*400+400+LMDE4;FILE LOC FOR POLYF SHUFFLE
46400  DUSR LPFSX  = PFSHF&X/X*400+400+LPFSHF;FILE LOC FOR POLYF SHFL. EXT.
47000  DUSR LPRCOM = PFSHX&X/X*400+400+LPFSX;FILE LOC FOR POLYF COMBINE
47400  DUSR LMDE5  = PRCOM&X/X*400+400+LPRCOM;FILE LOC FOR POLYF MODE 5
50000  DUSR LPFABL = MODE5&X/X*400+400+LMDE5;FILE LOC FOR POLYF ALLOCATE
50400  DUSR LPFRLS = PFABL&X/X*400+400+LPFABL;FILE LOC FOR POLYF RELEASE
51000  DUSR LMAPBU = PFRLS&X/X*400+400+LPFRLS;FILE LOC FOR POLYF MAP BL.
51400  DUSR LDATCK = MAPBU&X/X*400+400+LMAPBU;FILE LOC FOR POLYF DATA MAP
52000  DUSR LREDPO = DATCK&X/X*400+400+LDATCK;FILE LOC FOR READ POLYF
53000  DUSR LCALLP = READP&X/X*400+400+LREDPO;FILE LOC FOR POLY CALL
54000  DUSR LCLPY1 = CALLP&X/X*400+400+LCALLP;FILE LOC FOR POLY CALL DISCSUB EXT.
55000  DUSR LDSE4  = CLPY1&X/X*400+400+LCLPY1;END OF "DISCSUBS" GROUP #4

56000  DUSR LDSB5=2*400+LDSE4; "DISCSUBS" GROUP #5

56000  DUSR LS157 = LDSB5   ;FILE LOC FOR STYLUS' STRING LOCATOR
56400  DUSR LS105 = ST157&X/X*400+400+LS157;FILE LOC FOR STYLUS' CALL 89
57400  DUSR LS154 = ST151&X/X*400+400+LS105;FILE LOC FOR STYLUS' UNDERLINE
60000  DUSR LS153 = ST154&X/X*400+400+LS154;FILE LOC FOR STYLUS' CALL 56
60400  DUSR LS156 = ST153&X/X*400+400+LS153;FILE LOC FOR STYLUS' STRING JUST CALC
61000  DUSR LS152 = ST156&X/X*400+400+LS156;FILE LOC FOR STYLUS' PUSH OR POP
61400  DUSR LTP01 = ST152&X/X*400+400+LS152;FILE LOC FOR TYPIST'S CALL 81
62000  DUSR LTP02 = TIP01&X/X*400+400+LTP01;FILE LOC FOR TYPIST'S CALL 82
62400  DUSR LTP03 = TIP02&X/X*400+400+LTP02;FILE LOC FOR TYPIST'S CALL 83
63000  DUSR LTP04 = TIP03&X/X*400+400+LTP03;FILE LOC FOR TYPIST'S CALL 84
63400  DUSR LTP05 = TIP04&X/X*400+400+LTP04;FILE LOC FOR TYPIST'S CALL 85
64000  DUSR LTP06 = TIP05&X/X*400+400+LTP05;FILE LOC FOR TYPIST'S CALL 86
64400  DUSR LBAKU = TIP06&X/X*400+400+LTP06;FILE LOC FOR BAKUP DISCSUB
65000  DUSR LDSE5 = BAKU&X/X*400+400+LBAKU;END OF "DISCSUBS" GROUP #5
```

```
        65400  DUSR LDSB6=1*400+LDSE5; "DISCSUBS" GROUP #6

        65400  DUSR LATOE = LDSB6   ;FILE LOC FOR ASCII TO EBCDIC TRANSLATOR
        66000  DUSR LETOA = ATOE&X/X*400+400+LATOE;FILE LOC FOR EBCDIC TO ASCII

; 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
```

```
- PAGE 19

; "ACCOUNTS" FILE RECORD DISPLACEMENTS
                    ;               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 (DIV, 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   ; "              "

; CMR. IS DEFINED AS FOLLOWS

;  BIT
;    15     UNLIMITED DISC BLOCK ALLOTMENT FLAG (1 ==> UNLIMITED)
;    14-0   DISC BLOCK ALLOTED

; RUR. AND RUR1. ARE DEFINED AS FOLLOWS

; RUR1.
;   BIT
;     15     UNLIMITED CONNECT TIME FLAG (1 ==> UNLIMITED)
;     14     UNLIMITED CPU TIME FLAG (1 ==> UNLIMITED)
;     13-6   UNUSED
;     5-0    CPU SECONDS REMAINING (MOST SIGN. BITS)

; RUR.
;   BIT
;     15-0 CPU SECONDS REMAINING (LEAST SIGN. BITS)
```

```
; 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 USACE (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
 27   .DUSR CATR = 27  ;"CATALOG" RECORD NUMBER
 30   .DUSR CLAS = 30  ;CATALOG CLASSIFICATION (2 WORDS)
 32   .DUSR DOOM = 32  ;SPECIAL ACCESS FLAGS FOR 'DOOM'ED PROGRAMS
      ;       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 DOOM BITS

          1 .DUSR PROTDOOM   = 000001 ;enable maintaince access - obey normal protection
          2 .DUSR UNPRDOOM   = 000002 ;enable maintainance access - ignore protection
      20000 .DUSR WRCORDOOM  = 020000 ;enable unrestricted writing of core
      40000 .DUSR RDCORDOOM  = 040000 ;enable unrestricted reading of core
     100000 .DUSR EXECODOOM  = 100000 ;execute only, disallow listing of program

; DEFINE DEFAULT PROGRAM PRIORITY FOR NEW PROGRAM FILE

          5 .DUSR DPRI =5


; 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 FILREAD    = 12 ;READ ENTIRE FILE OF HEADER IN POOL
         13 .DUSR FILWRITE   = 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

         20 .DUSR D.READ      = 20 ;DIRECT READ (NOT VIA POOL)
         21 .DUSR D.WRITE     = 21 ;DIRECT WRITE (NOT VIA POOL)
         22 .DUSR DFILREAD    = 22 ;READ FILE BYPASSING POOL  } FOR REX (& $LCM)
         23 .DUSR DFILWRITE   = 23 ;WRITE FILE BYPASSING POOL } ONLY
            ;     SAVDISCSUB  = 24 ;SAVE (SSA) IN ITS POOL IMAGE, FOR REX USE ONLY
         25 .DUSR FILFFLUSH = 25 ;FLUSH A FILE (HDR. EXCLUDED).

            ; SUFFIX DEFINITIONS FOR THE COMMAND WORDS

     100000 .DUSR UL        = 100000 ;UNLATCH THE POOL BUFFER UPON COMPLETION
      40000 .DUSR CB        =  40000 ;CLEAR BUFFERS UPON COMPLETION
      20000 .DUSR CF        =  20000 ;CLEAR THE FILE BLOCKS FROM THE POOL (HDR. EXCLUDED)
```

```
; 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         *** USED ONLY IN CONFIG
    177763 .DUSR DSIZ =-15 ;SIZE OF DRIVER                 *** NOT LOADED INTO MEMORY
    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

       15   DUSR SZLVR =15 ;SIZE OF A LUVAR


        ; 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   REQUEST NCYL FOR LU 0 FROM OPERATOR FOR CTUTILITY
        ;    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" R8.2
```

```
;; "REX" == REAL-TIME EXECUTIVE FOR "IRIS" R8.2
;; WRITTEN BY DAN PAYMAR
;; MODIFIED FOR "IRIS" R8.0 BY G. DAVIE
;; LAST EDITED 30 APR 83 by CKW. (FOR R8.2)
;
;       .RDX 10
;
;MONTH  = 4
;DAY    = 30
;YEAR   = 1983
;
;;                      ALL RIGHTS RESERVED
;;       COPYRIGHT (C) 1974, EDUCATIONAL DATA SYSTEMS
;;       COPYRIGHT (C) 1979, EDUCATIONAL DATA SYSTEMS
;;       COPYRIGHT (C) 1982, POINT 4 DATA CORPORATION
;;       COPYRIGHT (C) 1983, 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 PIB:    .BLK    1

        5       1 RUP:    .BLK    1
        6       1 .BLK  1         ;REGNANT USER STORAGE POINTER (TO BECOME SPARE IN R8.3)
        7       1 RTP:    .BLK    1

       10       1 .BSA:   .BLK    1
       11       1 .HBA:   .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
```

```
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 SBA:   .BLK    1
41       1 DBA:   .BLK    1
42       1 C20:   .BLK    1
43       1 C37:   .BLK    1
44       1 C40:   .BLK    1
45       1 .BLK   1       ;RESERVED FOR STK. BASE PTR  ON MK8 CPU

46       1        .BLK    1
47       1 .BLK   1       ;AT 46 AND 47 FOR NOVA 3

50       1 C77:   .BLK    1
51       1 C100:  .BLK    1
52       1 C177:  .BLK    1
53       1 C200:  .BLK    1
54       1 C205:  .BLK    1
55       1 C215:  .BLK    1
56       1 C240:  .BLK    1
57       1 C244:  .BLK    1
60       1 C260:  .BLK    1
61       1 C271:  .BLK    1
62       1 C300:  .BLK    1
63       1 C334:  .BLK    1
64       1 C377:  .BLK    1
65       1 C400:  .BLK    1
66       1 C777:  .BLK    1
67       1 C1000: .BLK    1
70       1 C1777: .BLK    1
71       1 C2000: .BLK    1
72       1 C4000: .BLK    1

73       1 ESCF:  .BLK    1
74       1 ETSF:  .BLK    1
75       1 BSACF: .BLK    1
76       1 ERRF:  .BLK    1

77       1 .BPS:  .BLK    1
100      1 .INFO: .BLK    1
```

```
                  ; SYSTEM COMMAND CALLS
          6101 CALL     =JSR @.  ;CALL A SYSTEM SUBROUTINE
101          1          .BLK   1
          6102 FLAGC    =JSR @.  ;CHANGE OR CHECK A FLAG
102          1          .BLK   1
          6103 QCHAR    =JSR @.  ;QUEUE A CHARACTER TO BE PROCESSED
103          1          .BLK   1
          6104 QUEUE    =JSR @.  ;PUT A TASK IN THE QUEUE
104          1          .BLK   1
          6105 DQUEUE   =JSR @.  ;REMOVE REGNANT TASK FROM THE QUEUE
105          1          .BLK   1
          6106 CHANNEL  =JSR @.  ;PERFORM A CHANNEL OPERATION
106          1          .BLK   1
          6107 FREENODE=JSR @.   ;GET OR RELEASE A FREE 32-WORD NODE
107          1          .BLK   1
          6110 DATAPUMP=JSR @.   ;INITIATE A DMA DATA TRANSFER
110          1          .BLK   1


111          1 .INTR: .BLK   1
112          1 .NRET: .BLK   1
113          1 .SRET: .BLK   1
114          1 .LCM:  .BLK   1
```

```
                 ; SYSTEM SUBROUTINE CALLS
       6115 BINDIVIDE   =JSR @.
          1         .BLK    1
       6116 BINMULTIPLY=JSR @.
          1         .BLK    1
       6117 BUMPUSER    =JSR @.
          1         .BLK    1
       6120 DECIMAL     =JSR @.
          1         .BLK    1
       6121 FIX         =JSR @.
          1         .BLK    1
       6122 FLOAT       =JSR @.
          1         .BLK    1
       6123 FINDLUT     =JSR @.
          1         .BLK    1
       6124 GETBYTE     =JSR @.
          1         .BLK    1
       6125 INBYTE      =JSR @.
          1         .BLK    1
       6126 INSTBYTE    =JSR @.
          1         .BLK    1
       6127 ISA2DIGIT   =JSR @.
          1         .BLK    1
       6130 ISA2LETTER =JSR @.
          1         .BLK    1
       6131 LOADDA      =JSR @.
          1         .BLK    1
       6132 OUTBYTE     =JSR @.
          1         .BLK    1
       6133 OUTTEXT     =JSR @.
          1         .BLK    1
       6134 PUTBYTE     =JSR @.
          1         .BLK    1
       6135 READBLOCK   =JSR @.
          1         .BLK    1
       6136 RELJMPRET   =JSR @.
          1         .BLK    1
       6137 STORDA      =JSR @.
          1         .BLK    1
       6140 STINPUT     =JSR @.
          1         .BLK    1
       6141 STOUTPUT    =JSR @.
          1         .BLK    1
       6142 TRAPFAULT   =JSR @.
          1         .BLK    1
       6143 WRITBLOCK   =JSR @.
          1         .BLK    1
       6144 XGETBYTE    =JSR @.
          1         .BLK    1
       6145 XPUTBYTE    =JSR @.
          1         .BLK    1
       6146 SPINPUT     =JSR @.
          1         .BLK    1
       6147 STINTERACTION=JSR @.
          1         .BLK    1
```

```
                    ;POINTERS USED ONLY WITHIN "REX" AND "SYSGEN"

   150        1  .BRKP: .BLK    1
   151        1 JFLIO:  .BLK    1
   152        1  .FLIO: .BLK    1

   153        5 .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

           100 C600 =.INFO

   200        0 .BLK   INFO-400-.       ;OVERLAP CHECK

                    .EOT  ;PAGE ZERO FOR IRIS R8.2
```

BINDI	6115	BINMU	6116	BPI	16	BSACF	75	BUMPU	6117
C10	30	C100	51	C1000	67	C11	31	C12	32
C13	33	C14	34	C15	35	C16	36	C160	174
C163	175	C166	176	C17	37	C170A	21	C171	177
C177	52	C1777	70	C2	2	C20	42	C200	53
C2000	71	C205	54	C215	55	C240	56	C244	57
C260	60	C271	61	C3	3	C300	62	C334	63
C37	43	C377	64	C4	24	C40	44	C400	65
C4000	72	C5	25	C6	26	C600	100	C7	27
C77	50	C774C	22	C777	66	CALL	6101	CHANN	6106
CM400	23	DA	160	DAC	164	DAS	165	DATAP	6110
DB	166	DBA	41	DBC	172	DBS	173	DECIM	6120
DQUEU	6105	ERRF	76	ESCF	73	ETSF	74	FINDL	6123
FIX	6121	FLAGC	6102	FLOAT	6122	FREEN	6107	GETBY	6124
INBYT	6125	INSTB	6126	ISA2D	6127	ISA2B	6130	JFLTO	151
LOADD	6131	OUTBY	6132	OUTTE	6133	PIB	4	PUTBY	6134
QCHAR	6103	QUEUE	6104	READB	6135	RELJM	6136	RTP	7
RUP	5	SBA	40	SPINP	6146	STIMP	6140	STINT	6147
STORD	6137	STOUT	6141	TASKQ	15	TRAPF	6142	WRITB	6143
XGETB	6144	XPUTB	6145	.ABA	14	.BPS	77	.BRKP	150
.BSA	10	.DA	174	.DA3	175	.DB	176	.DB3	177
.FLTO	152	.HBA	11	.HXA	12	.INFO	100	.INTR	111
.LCM	114	.NRET	112	.SRET	113	.SSA	13		

Appendix C
LPTD DRIVER FILE LISTING

This appendix contains the listing of the LPTD driver file to aid in the installation and configuration process.

```
;       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
```



```
   32235  54513 FINIS:STA      3,RTNAD
```

```
;                               << 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 AC0 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 (0=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     @STRI       ;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
```

```
;                          << 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 D
CONFIG LISTING

This appendix contains the first four blocks of the CONFIG file which include:

Block 0 - Memory-resident processor list set up by SIR.

Block 1 - General and System INFO tables. Some values may be entered by the user when configuring the system.

Block 2 - Memory-resident DISCSUB list given in order of priority.

Block 3 - Disc Driver Table used to configure a particular controller/disc drive combination. Values that are entered into this table are provided on the Disc Specification sheets (see Section 1).

```
    PAGE 1
;                    << SI = R82CONFIGSC; BO = A.CONFIG.3162! >>

; "CONFIG" == Configuration file for "IRIS" R8.2
; "UNIVERAL" base file - Loc 0 to 1777 fits all discs


            12       .RDX 10


             4 MONTH = 4
            36 DAY   = 30
          3677 YEAR  = 1983



;                ALL RIGHTS RESERVED
;     Copyright (C) 1981, Point 4 Data Corporation
;     Copyright (C) 1982, Point 4 Data Corporation
;   This document may not be reporduced without the
; prior written permission of Point 4 Data Corporation.


; SYSTEM CONFIGURATION DATE (HOURS AFTER JAN 1 OF BASE YEAR)

         72020 SDATE = YEAR-BASEYEAR*12+MONTH-1*31+DAY-1*24


             1       .TXTM  1
            10       .RDX   8


; CONFIG file layout
;
;     0 - 277      ;Reserved
;   300 - 377      ;Initilization Table
;   400 - 577      ;General Information Table
;   600 - 777      ;System Information Table
;  1000 - 1177     ;Memory Resident Discsub List
;  1200 - 1377     ;Reserved
;  1400 - 2777     ;Disc Driver Table
;  3000 - 13777    ;Reserved

; 14000 - 15777    ;Disc Driver Index
; 16000 - 16377    ;Log On Restrictions Table
; 16400 - 17377    ;Log On Program Start Table
; 17400 - 17777    ;IPL Program Start Table
; 20000 - 77777    ;Disc Drivers


             0       .LOC   0         ;Block zero
      0 177777       -1


           300       .LOC   300       ;Driver init routine RDA list
```

```
     PAGE 2
;                 << SI = R82CONFIGSC; BO = A.CONFIG.3162! >>


            400         .LOC    400       ;GENERAL INFORMATION TABLE

     400  20000         20000   ;Partition Size
     401      1         1       ;Number of Partitions
     402      0         0       ;Memory Type


            600         .LOC    INFO      ;SYSTEM INFORMATION TABLE

     600  72020         SDATE   ;System creation date (Hours after base year)
     601   2000         2000    ;Average CPU speed (Instructions per millisecond) *
     602      1         1       ;Maximum # installable logical units
     603     12         12      ;Number of physical data channels (DFTs) per port
     604  65740         65740   ;Location of Port Control Area
     605      1         1       ;Total number of active ports (TNAP)
                                ;  (May be increased by SIR)
     606 120000         120000  ;Special conditions flags **
     607  43200         MEPS    ;Location of end of processor storage
     610 177777         177777  ;Top word of core to be used
     611   1004         1004    ;Auxiliary buffer size (number of words)
     612      0         0       ;Maximum number of user discsubs (DISCSUBS.USER)
     613      4         4       ;Number of extra character queue nodes
     614     40         40      ;Minimum # of free nodes
     615     30         30      ;Number of signal buffer nodes
     616    200         200     ;Maximum number of system discsubs (DISCSUBS)
     617  24003         24003   ;Time slice parameters (Long time
                                ;  slice * 400 + short time slice)
     620 177777         177777  ;reserved
     621 177777         177777  ;reserved
     622     10         10      ;start with 10 pseudo dev.'s
     623      7         .BLK    SZICON+INFO-.;(Reserved)

;     *  For MARK 3, set to 1200
;        For NOVA 3 CPU, add 100000 for interrupt detour

;     ** Bit 15 = 1  ==>  No dirty buffers
;        Bit 14 = 1  ==>  Suppress BASIC error text
;        Bit 13 = 1  ==>  Intra user buffering
```

```
     PAGE 3
;                     << SI = R82CONFIGSC; BO = A.CONFIG.3162! >>

          1000         .LOC    1000      ;MEMORY RESIDENT DISCSUB LIST

;     Note:  The order of the discsubs in the following list is
;            the most to least important for a normal 64K word
;            system.  If it is necessary to remove memory resident
;            discsubs, remove or replace from the bottom of the list.


1000      67         AFSET
1001     100         LINKP
1002     101         LOADP
1003       3         FFILE
1004      15         ACNTLOOKUP
1005      22         OPEN&377
1006      26         CLOSE
1007      30         GETRR&377
1010      33         READITEM
1011       1         ALLOCATE
1012      40         CHARGE
1013      36         READCONTIG
1014      61         SEARCH&377
1015      62         SHUFFLE
1016      63         DEKEY
1017      27         CLEAR
1020      46         SPECIAL
1021      57         SIGPAUSE
1022      41         SYSCO
1023 177777          -1
```

```
;                    << SI = R82CONFIGSC; BO = A.CONFIG.31621 >>

       1400         .LOC   1400      ;DISC DRIVER TABLE

 1400       1 1     ;Real core address of LUFIX (Set by "SIR")
 1401   77777 77777 ;Virtual (listing) address of system disc driver
 1402   77777 77777 ;Virtual (listing) address of block zero utility driver
 1403       1 1     ;Actual Number of partitions for this driver
 1404      52 52    ;Device code of controller
 1405     500 500   ;Ratio for calculating MINB
 1406       0 0     ;(Reserved for future use)
 1407       0 0     ;(Reserved for future use)

               ;Partition 0.0 (IRIS system LU 0)

 1410       0 0     ;Real core address of LUVAR (set by "SIR")
 1411       0 0     ;NPTC - Number of physical tracks per cylinder
 1412       0 0     ;DFLG - Disc flag word
 1413       0 0     ;RESERVED
 1414       0 0     ;PHYU - Physical Unit selection
 1415       0 0     ;FCYL - First cylinder #
 1416       0 0     ;NCYL - Number of cylinders
 1417       0 0     ;NTRS - [# tracks] *100 + [# sectors]

 1420 177777 -1     ;Terminator for Drive Table for "UNIVERSAL" CONFIG base file

                    .END   ;R8.2 "UNIVERAL" CONFIG file base
```

BINDI	6115	BINMU	6116	BPI	16	BSACP	75	BUMPU	6117
C10	30	C100	51	C1000	67	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	52	C1777	70	C2	2	C20	42	C200	53
C2000	71	C205	54	C215	55	C240	56	C244	57
C260	60	C271	61	C3	3	C300	62	C334	63
C37	43	C377	64	C4	24	C40	44	C400	65
C4000	72	C5	25	C6	26	C600	100	C7	27
C77	50	C774C	22	C777	66	CALL	6101	CHANN	6106
CM400	23	DA	160	DAC	164	DAS	165	DATAP	6110
DAY	36	DB	166	DBA	41	DBC	172	DBS	173
DECIM	6120	DQUEU	6105	ERRP	76	ESCP	73	ETSP	74
FINDL	6123	FIX	6121	FLAGC	6102	FLOAT	6122	FREEN	6107
GETBY	6124	INBYT	6125	INSTB	6126	ISA2D	6127	ISA2L	6130
JFLTO	151	LOADD	6131	MONTH	4	OUTBY	6132	OUTTE	6133
PIB	4	PUTBY	6134	QCHAR	6103	QUEUE	6104	READB	6135
RELJM	6136	RTP	7	RUP	5	SBA	40	SDATE	72020
SPINP	6146	STINP	6140	STINT	6147	STORD	6137	STOUT	6141
TASKQ	15	TRAPP	6142	WRITB	6143	XGETB	6144	XPUTB	6145
YEAR	3677	.ABA	14	.BPS	77	.BRKP	150	.BSA	10
.DA	174	.DA3	175	.DB	176	.DB3	177	.FLTO	152
.HBA	11	.HXA	12	.INFO	100	.INTR	111	.LCM	114
.NRET	112	.SRET	113	.SSA	13				

COMMENT SHEET

MANUAL TITLE IRIS Installation and Configuration Manual

PUBLICATION NO. SM-030-0009 REVISION 12

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, CA 92714
(714) 863-1111