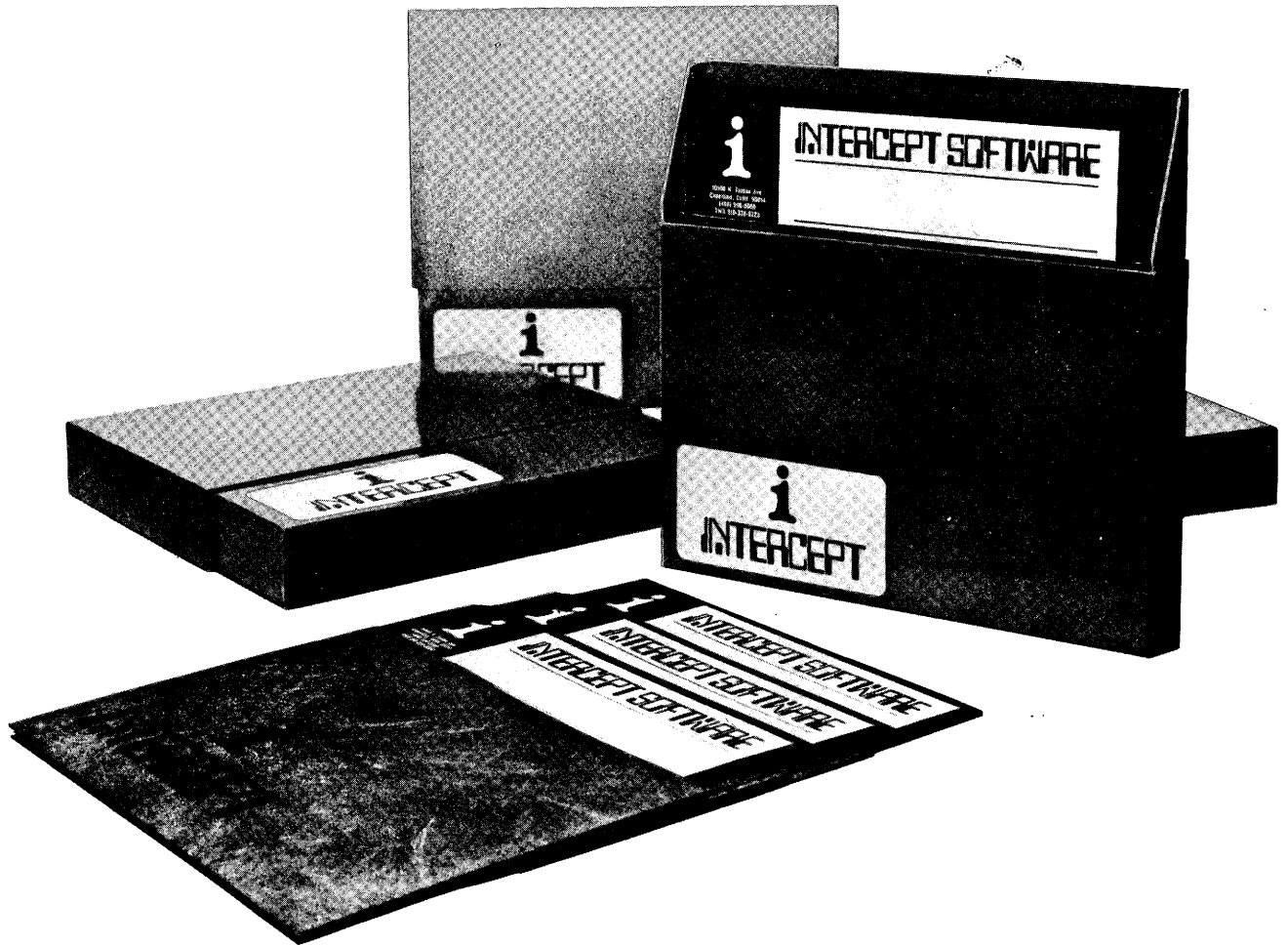


\$500

INTERCEPT FLOPPY DISK OPERATING SYSTEM FROM INTERSIL INTERFIL



IFDOS HANDBOOK



.

.



.

.



INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
TABLE OF CONTENTS

CHAPTER	CONTENTS	PAGE
1.	INTRODUCTION	
1.1	File system	1-1
1.2	KEYBOARD MONITOR	1-1
1.3	EDITOR	1-1
1.4	PAL	1-1
1.5	BIN	1-1
1.6	ODT	1-1
1.7	IFOCAL	1-2
1.8	Utility programs	1-2
2.	SYSTEM CONCEPTS & CONVENTIONS	
2.1	Logical devices	2-1
2.2	Files	2-1
2.3	Definitions of commonly used terms	2-2
2.4	Documentation conventions	2-2
3.	KEYBOARD MONITOR	
3.1	Introduction	3-1
3.2	KEYBOARD MONITOR commands	3-1
3.3	KEYBOARD MONITOR errors	3-4
4.	EDITOR	
4.1	Introduction	4-1
4.2	EDITOR commands	4-1
4.3	EDITOR command examples	4-4
4.4	Special EDITOR commands	4-4
5.	SYSTEM COMMANDS	
5.1	Introduction	5-1
5.2	SYSTEM COMMAND syntax	5-1
5.3	Available SYSTEM COMMANDS	5-2
6.	PAL ASSEMBLER	
6.1	Introduction	6-1
6.2	How to use PAL	6-1
6.3	PAL switches	6-2
6.4	PAL error messages	6-4
6.5	PAL language summary	6-7
7.	BIN LOADER	
7.1	How to use BIN	7-1
7.2	BIN switches	7-1
8.	OCTAL DEBUGGING TECHNIQUE	

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
TABLE OF CONTENTS

8.1	How to use ODT	8-1
8.2	ODT commands	8-1
9.	OTHER SYSTEM COMMANDS & UTILITIES	
9.1	BATCH	9-1
9.2	BSAVE	9-3
9.3	CHANGE	9-4
9.4	DATE	9-5
9.5	DUMP	9-6
9.6	FIND	9-9
9.7	GET	9-10
9.8	HILOAD	9-11
9.9	MEMORY	9-12
9.10	START	9-13
9.11	VARY	9-14
10.	EXTENDED SYSTEM COMMANDS & UTILITIES	
10.1	Introduction	10-1
10.2	SYSTYP	10-1
10.3	SYSCPYP	10-1
11.	IFDOS DETAILED SYSTEM INFORMATION	
11.1	System memory use	11-1
11.2	Resident system code	11-2
11.3	File formats	11-4
11.4	The system device	11-6
	APPENDICES	
A.	Bootstrapping procedure	A-1
B.	IFDOS ASCII table	B-1
C	Control panel PROM listing	C-1
D.	INTERCEPT SOFTWARE REPORTS	D-1
E.	Getting started with IFDOS	E-1

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 1--INTRODUCTION

The IFDOS monitor system is a complete program development system for the INTERCEPT prototyping system with at least 4096 words of random access memory, a direct access mass storage device, and an ASCII terminal. The system contains the following basic components:

1.1 IFDOS FILE SYSTEM

The file system maintains a catalogue of user files and performs basic file handling functions as specified by the user.

1.2 KEYBOARD MONITOR

The KEYBOARD MONITOR provides communication between the user and the monitor system. With simple keyboard commands the user may enter and delete files in the user catalogue, transfer files between memory and mass storage, print the user catalogue on the console terminal, and call system programs.

1.3 SYSTEM EDITOR

The system EDITOR is available to the user at the KEYBOARD MONITOR level, allowing the user to quickly and easily create and modify ASCII text at the console terminal.

1.4 PAL ASSEMBLER

The IFDOS PAL assembler accepts source programs created by the system EDITOR and produces binary output for subsequent loading and execution. It is the fastest known assembler of its kind and provides for great flexibility of assembler operations and listing formats.

1.5 BIN LOADER

The IFDOS binary loader loads and executes the binary output of the PAL assembler, or alternatively, input may be taken from a paper tape in either RIM or BIN format.

1.6 OCTAL DEBUGGING TECHNIQUE

IFDOS ODT allows the user to examine, modify, and control execution of programs contained in memory from the console terminal.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 1--INTRODUCTION

1.7 IFOCAL INTERPRETER

IFDOS IFOCAL is an interactive high-level language interpreter and on-line calculator. It also interacts with the file system for run time loading and saving of user programs and data. IFOCAL will be described in a separate manual.

1.8 SYSTEM UTILITY PROGRAMS

IFDOS provides facilities for absolute block copying and dumping of mass storage devices, printing of the system program catalogue, system date handling, and changing of system parameters through numerous utility programs.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 2--SYSTEM CONCEPTS & CONVENTIONS

2.1 LOGICAL DEVICES

The basic unit of mass storage within IFDOS is the logical device. A given physical device may, depending on size, contain up to 8 logical devices. Small devices, like floppy disks, generally contain one logical unit. However, a single large device (some hard disks, for example) may be divided into four or eight logical units. Logical devices are designated by the numbers 0-7.

IFDOS is supplied to the user on a flexible diskette, which, when mounted in physical drive 0 of the INTERCEPT D10 disk and bootstrapped, becomes the IFDOS system device. Any logical device may be the current system device. However, when the system is initially bootstrapped, the system device is always logical device 0.

The system device contains a complete copy of the system in memory image form, and also space for user information. All KEYBOARD MONITOR commands are directed to the current system device unless another logical device is explicitly specified in the KEYBOARD MONITOR command line. (the KEYBOARD MONITOR is discussed in CHAPTER 3.)

2.2 FILES

The IFDOS file system handles two general types of files: System files and user files.

SYSTEM FILES

System files contain memory images of the system programs (PAL, BIN, etc.). These programs are implemented as system commands--they reside near the beginning of the system device and are loaded by the system I/O routines.

USER FILES

User files may be of two kinds--source files and binary files. Source files are created by the EDITOR and are used as input to the PAL assembler. Binary files are produced by PAL and used as input to the BIN loader. User files are located above the system files and represented by entries in the user catalogue.

User files are of a fixed length--16 blocks of 128 words each. To facilitate editing, large source programs

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 2--SYSTEM CONCEPTS & CONVENTIONS

must be divided into multiple files. The assembler handles large programs by accepting multiple input files and producing one or more object files as output.

The fixed length file format provides many advantages in a small system environment. First, the user catalogue is so simple and concise that it may be conveniently contained in memory--thus eliminating much of the directory I/O normally required by an operating system. Secondly, if a single portion of a source program requires editing, only the file containing that portion of the program must be loaded into memory for the editing operation. When editing is completed, the updated file will always fit into its former position on mass storage, thus eliminating the buildup of empty file slots.

In addition to the user files, which may number up to 45, two special user files are located before the system files on the system device. These are scratch files, called "%" and "\$". For program development, it is convenient to use the \$ file for a source program and the % file for assembler binary output. The location of the scratch files greatly reduces their access time on slow mass storage devices.

2.3 DEFINITIONS OF SOME COMMONLY USED TERMS

BYTE	One half of a memory word, or 6 bits, or one IFDOS ASCII character
BLOCK	128 (200 octal) consecutive words of a mass storage device
PAGE	128 (200 octal) consecutive words of main memory
FILE	An area of contiguous blocks on mass storage with a corresponding catalogue entry

2.4 DOCUMENTATION CONVENTIONS

In some cases it is necessary to reference non-printing characters or key combinations on the console terminal. The following notations are used:

<CR> Carriage return

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 2--SYSTEM CONCEPTS & CONVENTIONS

<LF>	Line feed
<CR/LF>	Carriage return followed by a line feed
CTRL/x	A control character--x represents any alphabetic character (A-Z). Produced by depressing the "CTRL" key and another key simultaneously.
SHIFT/x	A shift character--x represents any alphanumeric character (A-Z,0-9). Produced by depressing the "SHIFT" key and another key simultaneously.
<ALT>	Altmode, labeled "ESCAPE" on some terminals
<RUB>	Rubout, labeled "DELETE" on some terminals

Under some circumstances, arguments or command parts are optional. Optional characters are enclosed in square brackets "[]".

Command examples in this manual are illustrated by simulated computer output. Most sample command lines begin with a "DOT". This dot is the KEYBOARD MONITOR prompting character and is not typed by the user.



INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 3--THE KEYBOARD MONITOR

3.1 INTRODUCTION

When the system has been successfully bootstrapped, the KEYBOARD MONITOR prints a "DOT" at the left margin indicating its readiness to accept user commands. (Bootstrapping information is given in APPENDIX A.) When the user types a KEYBOARD MONITOR command it is packed into the KEYBOARD MONITOR buffer. When the KEYBOARD MONITOR detects a <CR> or <ALT> command terminator, it echoes a <CR/LF> and begins to process the command in the buffer. If a mistake is made prior to terminating the command line, it may be corrected as follows:

<RUB> Echoes a backslash and the last character typed, and deletes a character (the last entered) from the KEYBOARD MONITOR buffer. Subsequent rubouts echo and delete additional characters from the keyboard monitor buffer.

SHIFT/O (backarrow) Same as rubout.

CTRL/U Echoes "ZAP !" followed by <CR/LF> and deletes the entire command line from the KEYBOARD MONITOR buffer. A "DOT" is printed at the left margin indicating that the keyboard monitor is awaiting further user input.

When entering a command, the user may verify the contents of the KEYBOARD MONITOR buffer as follows:

<LF> Prints the current command line on the terminal and awaits further user input.

A <CR> or <ALT> must be issued by the user to terminate the command line before it is processed by the system. Command lines may not exceed 80 characters. The KEYBOARD MONITOR recognizes three types of input lines: KEYBOARD MONITOR commands, EDITOR commands, and SYSTEM COMMANDS. EDITOR commands and SYSTEM COMMANDS will be covered later.

3.2 KEYBOARD MONITOR COMMANDS

KEYBOARD MONITOR commands consist of a two to six letter command followed by an argument. All KEYBOARD MONITOR commands may be abbreviated to the first two

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 3--THE KEYBOARD MONITOR

characters, and sometimes, arguments are optional or unnecessary. Arguments may be separated from commands by one or more spaces or tabs. An argument consists of a file name and/or a logical device designation. Most KEYBOARD MONITOR commands take the following general form:

.COMMAND FILNAM:U

"COMMAND" is a KEYBOARD MONITOR command from the list below.

"FILNAM" is a user filename, which may be from one to six alphanumeric (A-Z,0-9) characters or the special characters ([, \,], ^).

"U" is a logical device number (0-7) which refers to a device of the same type as the system device.

If no logical device number is specified in a keyboard monitor command, the number of the current system device is assumed. Whenever a logical device number is included in a KEYBOARD MONITOR command, that device will be the new system device when the KEYBOARD MONITOR again prompts the user with a "DOT".

The following is a list of KEYBOARD MONITOR commands and their functions:

CA[TALOG] [:U] Prints the user catalogue of device "U" on the console terminal. Empty file slots are marked "<FREE>".

DE[LETE] FILNAM [:U] Deletes the specified file entry from the user catalogue of device "U". DELETE is a catalogue operation only--it does not effect the contents of the file.

EC[HO] [N] Sets or resets the system echo switch. If N is even, the switch is turned off. If N is odd, the switch is turned on. If N is omitted, the state of the echo switch is complemented. With echo on, all keyboard input is echoed on the terminal. With echo off, keyboard echoing, many error messages, and the prompting "DOT" are inhibited.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 3--THE KEYBOARD MONITOR

EN[TER] FILNAM[:U] Enters "FILNAM" in the first free file slot of the specified device's user catalogue. ENTER, like DELETE, is a catalogue operation only. Therefore, a filename may be reentered after accidental deletion and the contents of the file will be unaltered.

HA[LT] Halts the computer at location 00600. The system is restarted by depressing "CONT" on the INTERCEPT frontpanel.

LO[AD] FILNAM[:U] Transfers the specified file from "U" to the EDITOR buffer. LOAD does not alter the contents of the file. Once the file is loaded the user may edit it and/or store it on another logical device. (see the "EDITOR" in CHAPTER 4 and the "SAVE" command below.)

MO[UNT] Bootstraps the monitor by executing a jump to location 07600.

PL[EASE] [STRING] Types the entire command line on the terminal. This command is useful for issuing instructions to the computer operator during a BATCH run. The line is typed, the system rings the bell, and waits for the operator to strike any key on the keyboard. The key is ignored and BATCH processing continues.

RA[DIX] [N] The argument "N", which may be 8 or 10, sets the radix for interpretation of the system's numeric input and output. If no argument is specified, the radix is changed from its current value to the other value. Initially, the radix is set to 10.

SA[VE] FILNAM[:U] Transfers the contents of the EDITOR buffer to the specified file on "U". If "FILNAM" is not an entry in the user catalogue, an ENTER will be done before the EDITOR buffer is transferred. Save may be used in conjunction with LOAD to copy a file from one logical device to another.

SY[STEM] [:U] Makes logical unit "U" the current system device. All further KEYBOARD MONITOR commands which do not explicitly state a

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 3--THE KEYBOARD MONITOR

logical device number will be performed on the new system device by default.

TA[BS] [N] Changes the system's interpretation of <TAB> on output. The argument N, if present, may take three values. If N=0, tabs are printed as single spaces. If N=3, tabs are printed as the number of spaces required to reach the next tab-stop. If N=5, tabs are printed as in N=3, except they are followed by a null character. If N is omitted, N=5 is assumed. Initially, N=3.

^^[STRING] Used to insert comments in a BATCH stream. "STRING", if given, will be typed on the terminal during execution of the BATCH file and will be ignored by the KEYBOARD MONITOR. BATCH processing continues with the next command.

3.3 KEYBOARD MONITOR ERROR MESSAGES

WHAT ? The command was not a KEYBOARD MONITOR, EDITOR, or SYSTEM COMMAND, or the command did not start with a legal line number. The command is ignored. The KEYBOARD MONITOR prints a "DOT" and awaits user input.

"NAME" NOT FOUND ! A non-existent filename was specified in a DELETE or LOAD command or a command was given that does not reside in the SYSTEM COMMAND catalogue.

CATALOG FULL ! the user catalog has no space for additional file names. The user must DELETE unwanted file names to make space for new entries.

NO SPACE LEFT ! The capacity of the EDITOR buffer has been reached. The buffer contents must be SAVED in a file and the buffer must be SCRATCHed before further text entry.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 4--THE SYSTEM EDITOR

4.1 INTRODUCTION

The system EDITOR is available to the user at the KEYBOARD MONITOR level. The user may use EDITOR commands in response to the prompting "DOT" of the KEYBOARD MONITOR without having to run a separate EDITOR program.

The EDITOR maintains a buffer in memory starting at location 03000. The length of the buffer is 2048 words, which corresponds exactly to one IFDOS user file. Input and output between the EDITOR buffer and mass storage is via the KEYBOARD MONITOR "LOAD" & "SAVE" commands.

The system EDITOR is line oriented, with commands to insert, modify, delete, and search for text on a line-by-line basis. Each text line in the EDITOR buffer has a unique line number associated with it. The user may edit a line by referring to its line number in an EDITOR command line.

4.2 EDITOR COMMANDS

EDITOR commands are of two general forms. A command may consist of a line number followed by text or it may be a command from the list in this chapter followed, in some cases, by one or more arguments.

The EDITOR treats these commands differently. If an EDITOR command line begins with a number, the EDITOR buffer is searched for another line with the same number. If one is found, the new line replaces it. If none is found, the new line is entered in the buffer. An existing line is deleted by typing its number followed by <CR>.

Text lines are stored in the EDITOR buffer sequentially by line numbers. Therefore, a line may be inserted between two existing lines by giving it a number between the existing numbers.

Line numbers must be between 0 and 4094 inclusive. Line numbers are interpreted as decimal or octal depending upon the current setting of the system radix. Using a number larger than 4094 (7776 octal) results in the "WHAT ?" error message.

Line numbers are stored in the EDITOR buffer starting at the end and extending toward the beginning of the buffer.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 4--THE SYSTEM EDITOR

They are ignored by programs such as the PAL assembler, which has its own line numbering scheme. Each line number occupies two words of EDITOR buffer storage, and each character in the source program occupies one byte.

Since file length is limited to 2048 words, and the line number imposes a two word overhead on each line in the EDITOR buffer, the maximum number of characters an ASCII file can contain ranges between 3000 and 3500.

It is very important to leave at least 100-200 free words in each ASCII file to allow flexibility for subsequent editing.

Since files are of a limited length, large source programs must be segmented into several files. After editing, the several files can be assembled collectively to create a single object program.

Like KEYBOARD MONITOR commands, EDITOR commands may be abbreviated to the first two letters. EDITOR commands listed below (except the EDIT command) take two general forms:

1. .COMMAND L-M
2. .COMMAND L I

In the first form, arguments L and M name the first and last lines, respectively, which the command affects. If L alone is specified, the command affects a single line. If L and M are both given, the command affects a range of lines, including lines L and M. If no arguments are present, the entire EDITOR buffer is affected. Specifying "L-" refers to everything from line L to the end of the buffer. "-M" refers to all lines from the beginning of the buffer up to and including line M.

The second form of EDITOR command only applies to SEQUENCE and RESEQUENCE. Arguments L and I name the sequence of numbers L, L+I, L+I+I, etc. If the user omits argument I, a default value of 10 (12 octal) is used. If both arguments are omitted, the values 100 and 10 (144 and 12 octal) are assigned to L and I respectively. The following is a list of EDITOR commands:

ED[IT] L [N] Selects line L for editing. The EDITOR
 types the number L on the terminal and waits
 for the user to re-type the line or modify it

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 4--THE SYSTEM EDITOR

using the special EDITOR commands (see special EDITOR commands later in this chapter). A line number L must appear or the "WHAT?" error occurs. When the line is terminated, it replaces the old line in the EDITOR buffer. If both arguments L and N appear, the contents of line L are moved to line N.

- LA[ST] Prints the number of the last line currently in the EDITOR buffer.
- LE[FT] Prints the number of words currently unused in the EDITOR buffer. Doubling this value gives the approximate number of characters which may be inserted in the buffer before overflow. The user should be careful to leave enough space in a text file to allow corrections and additions at a later time.
- LI[ST] [L-M] Lists the specified range of lines on the terminal. If no argument is given, the entire buffer is listed.
- PU[NCH] [L-M] Similar to list except a long leader precedes the text output, line numbers are omitted from the output, and a trailer is punched after the text output.
- RE[SEQUENCE] [L I] Renumbers the EDITOR buffer, starting with the number L and using I as the sequence index or incremental value for succeeding line numbers.
- SC[RATCH] [L-M] Deletes the specified range of lines from the EDITOR buffer. Giving L without M deletes only line L. Omitting both arguments erases the buffer.
- SE[QUENCE] [L I] Provides an automatic sequence of line numbers, starting with L, then L+I, L+I+I, etc. The EDITOR types a line number and waits for the user to enter text on that line. When the user terminates the line, the index is automatically added to the current line number and the EDITOR types the updated line number. This process continues until the user cancels sequence mode by typing CTRL/C or CTRL/D.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 4--THE SYSTEM EDITOR

4.3 EDITOR COMMAND EXAMPLES

^^EDITOR COMMAND EXAMPLE #1

```
.SE 10 10
10  START,  CLA
20          TAD      P      /INITIALIZE MESSAGE POINTER
30          DCA      PNTR
40  LOOP,  TAD I   PNTR  /GET A CHARACTER
50          SNA      /ALL DONE?
60          JMP I   K7600 /YES, BACK TO MONITOR
70          TLS      /NO, PRINT
80          TSF      /FLAG UP?
90          JMP      .-1  /NO, WAIT FOR IT
100         ISZ      PNTR /BUMP POINTER
110         JMP      LOOP /KEEP GOING
120  P,      LIST
130  PNTR,   0
140  K7600,  7600
150  LIST,   311;240;306;240;304;240;317
160         240;323;240;241;215;212;000
170  ^D
```

At this point the program has been entered and sequence mode has been canceled. The source program is still in the EDITOR buffer, and can be saved or assembled. The program has an error, however. The AC must be cleared after each character is typed. The correction follows:

```
.75          CLA
```

This places the "CLA" between lines 70 and 80. Now let us renumber the program.

```
.RE 10 10
170
```

Line 170 is now the last line in the buffer.

4.4 SPECIAL EDITOR COMMANDS

The system EDITOR provides several special commands to facilitate line editing. Prior to using these commands, a

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 4--THE SYSTEM EDITOR

single line must be selected for editing with the following command:

ED[IT] L Where L names the line to be edited.

When a line has been selected in this manner, the EDITOR prints the line number on the terminal and waits for user action. The following commands are available:

CTRL/Gx Control G followed by any single character x. The EDITOR searches for the first occurrence of the search character x on the present line, and when x is found, all characters up to and including x are printed. The EDITOR awaits text or command input.

CTRL/A Searches for the next occurrence on the present line of the search character specified in the CTRL/G command, and prints all characters up to and including the search character.

CTRL/S Prints the entire text line on the terminal and waits for the user to append text at the end of the line, or terminate the line with a <CR>.

CTRL/K Deletes that portion of the line which has been printed on the terminal, excluding the line number, and awaits text or command input.



INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 5--SYSTEM COMMANDS

5.1 INTRODUCTION

In addition to the user file catalogue, every logical device contains a system catalogue (or system command catalogue--the terms are used interchangeably). The system catalogue contains an entry for each system program which resides on the device, as well as information used by the system I/O routines to control loading and execution of the programs.

System programs may be called from the KEYBOARD MONITOR by typing the name of the program followed by any arguments the program requires to perform its function. Arguments may include filenames, device specifications, switch options and an octal number. When the KEYBOARD MONITOR determines that a system command has been given, it decodes all arguments in the command line and passes them to the system program. Arguments are interpreted differently by each system program.

5.2 SYSTEM COMMAND SYNTAX

Most system commands take the following format:

.COMMAND:UA OFILES:UB<IFILES:UC(SWITCHES)=N

"COMMAND" is a system command with an entry in the system command catalogue on the logical device designated by ":UA".

"OFILES" is one or more output file names, separated by commas or spaces, on device "UB".

"<" delimits the output file list.

"IFILES" is one or more input file names, separated by commas or spaces, on device "UC".

"(SWITCHES)" is one or more switch options. Alternatively, the parentheses may be omitted if each switch is preceded by a slash ("/").

"=N" is an octal number in the range 0-7777.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 5--SYSTEM COMMANDS

The following rules apply to system command syntax:

"COMMAND" must not be abbreviated, and must be separated from the first file specification by a space or comma.

No more than 17 filenames (input+output) may be given in a system command. If one output file is named, 16 input files may be named. If 8 output files are named, 9 input files may be named, etc. Since commands may not exceed 80 characters, it is best to use short filenames if many files are specified.

SYSTEM COMMANDS may be terminated by a <CR> or <ALT>. The KEYBOARD MONITOR does not distinguish between the two terminators.

5.3 AVAILABLE SYSTEM COMMANDS

TABLE 5-1 lists all available system commands. Each SYSTEM COMMAND is described separately elsewhere in this manual. Meanings of command arguments and detailed functions of the system programs are discussed there.

TABLE 5-1

<u>COMMAND</u>	<u>FUNCTION</u>
BATCH	executes command files
BIN	loads binary files
BSAVE	creates binary files from memory
CHANGE	changes EDITOR buffer text
DATE	sets and prints system date
DUMP	examines, changes, and transfers mass storage
FIND	searches EDITOR buffer for text
GET	produces memory images of binary files
IFOCAL	runs the IFOCAL interpreter program
MEMORY	gives information about memory size

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 5--SYSTEM COMMANDS

ODT runs the octal debugging technique program
PAL creates binary files from ASCII source files
START starts a program after loading with GET
VARY sets system parameters



INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 6--PAL ASSEMBLER

6.1 INTRODUCTION

IFDOS PAL is a 4K, two pass assembler which accepts standard IFDOS source files as input. Pass 1 reads the input file and generates a symbol table. Pass 2 reads the input file again, and using the symbol table from pass 1, produces an absolute binary object file. Octal and symbolic program and symbol table listings are produced during pass 2 if requested.

IFDOS PAL recognizes a number of switch options to control output format and chain to the BIN loader. Loader switch options and a starting address may be specified on the PAL command line, therefore a user may assemble, list, load, and execute a program with a single PAL command.

Although PAL is designed to operate efficiently in a 4K environment, it is able to take advantage of larger configurations. The primary advantage of a larger memory configuration (as far as PAL is concerned) is that larger programs with more symbols may be assembled. Each additional 4K of memory above field 0 allows PAL to handle an additional 1024 symbols. In a 4K environment, PAL accepts approximately 400 user defined symbols.

6.2 HOW TO USE PAL

The PAL assembler is called along with file specifications and switch options as follows:

.PAL OFILES<IFILES(SWITCHES)=N

"OFILES" are output file names separated by commas or spaces.

"IFILES" are input file names separated by commas or spaces.

"SWITCHES" are switch options from table 6-1.

PAL accepts up to 17 files. This allows large programs to be segmented by the user into several files for editing purposes and combined into one assembly. One output file must be specified for approximately each 2K words of memory used by the program.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 6--PAL ASSEMBLER

For example, assume the user has entered a large source program into files F1 through F10. The program uses approximately 8K of memory, therefore 4 output files must be specified. The program is assembled as follows:

```
.PAL 01,02,03,04<F1,F2,F3,F4,F5,F6,F7,F8,F9,F10(BLG)=200
```

The ten input files are assembled into four output files, a listing is generated, and when assembly is done PAL chains to BIN, passing certain switch options.

TABLE 6-1 gives the meaning of all PAL switch options and indicates the switches PAL passes to BIN.

6.3 PAL SWITCHES

The following table lists the switch options recognized by PAL:

TABLE 6-1

/A	all symbols, including permanent symbols, are typed in the symbol table listing at the end of pass 2. /A implies /S.
/B*	force binary output, use unit 0 % scratch file for object output if no output file specified.
/C	maximize configuration. Use up to 4K of memory above the software memory size for symbol table storage, if that memory exists.
/D*	force binary, use unit 0 % and \$ scratch files for object output if no output files specified.
/F**	passed to BIN.
/G**	passed to BIN.
/H**	passed to BIN
/I**	passed to BIN.
/J	do not generate zero fill-words for TEXT & SIXBIT psuedo-ops.
/K	force two pass assembly to find all errors, but don't

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 6--PAL ASSEMBLER

produce binary output unless output files were specified. With /L/N, the first line of each new input file becomes the listing title.

- /L produce listing during pass 2. Other options specify the device which gets the listing.
 - /N produce neat terminal output. /N does not imply /L, but if used with /L, titles, dates, pagination, etc., are included in the listing during pass 2. This option removes one page (32 symbols) from PAL symbol table space.
 - /O enables off-page references (links). Also see /Q.
 - /P assume 132 column output for symbol table. (up to 8 columns of symbols).
 - /Q used with /O. Enables literals and suppresses "LG" warning errors. If /O is used without /Q, links are generated correctly but they are flagged with the "LG" warning message. If /Q is used without /O, literals are generated but links are not. If both /O and /Q are used, both links and literals are generated and the "LG" message is suppressed. This switch removes 5 pages (160 symbols) from PAL symbol table space.
 - /R** passed to BIN.
 - /S produce symbol table listing at end of pass 2.
 - /U* implies /B, uses unit 1 % scratch file for output. If used with /D, unit 1 % and \$ scratch files will be used for output.
 - /V** passed to BIN.
 - /W** passed to BIN.
 - /Z** passed to BIN.
 - /1-/7* passed to BIN.
 - =n* passed to BIN.
- ** switches marked with two asterisks ("**") cause a chain to BIN at the end of assembly. These switches have no

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 6--PAL ASSEMBLER

other effect on PAL, but are maintained during the chain and processed by BIN. * switches marked with one asterisk have no effect whatever on PAL, but are processed by BIN if BIN is chained to. See table 7-1 for a description of BIN switches.

6.4 PAL ERROR MESSAGES

PAL error messages consist of a two character error code followed by the octal location in the user's program where the error occurred. In some cases additional information is supplied which may help the user determine the cause of the error. Errors are typed on the terminal as they are found by the assembler. At the end of assembly, a summary is printed listing the number of errors, links, and symbols in the assembly, and the amount of memory used in the assembly.

TABLE 6-2 lists the PAL error diagnostic codes.

TABLE 6-2

<u>ERROR CODE</u>	<u>MEANING</u>
BO	Binary overflow. The object output of the assembly is too lengthy to fit in the output files allotted by the user. Use more output files (approximately one output file is required for each 2K of memory used by the program). Fatal.
CO	Conditional assembly error. Indicates a syntax error in or the misuse of a conditional assembly statement (a statement using IFZERO, IFNZRO, etc). Fatal.
DT	Duplicate tags. An attempt was made to redefine an existing symbol as a tag. Non-fatal--the symbol is not redefined.
ER	User error. The assembler has detected an error which it cannot precisely categorize.
FI	Fixmri error. Error in equating a memory reference type symbol. Non-fatal--the symbol is not defined.
IC	Illegal character. The 6-bit ASCII code of the offending character is included in the error message. Non-fatal--the line containing the illegal character

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 6--PAL ASSEMBLER

is ignored.

- IE Illegal expression in an equate. A symbol has been equated to an expression containing an element not previously defined. Non-fatal--the equated symbol remains undefined.
- II Illegal indirect. An indirect reference was made to an off-page operand. A link cannot be generated for an instruction already using indirect addressing. This message is only generated when both /O and /Q are in effect. Non-fatal--the offending instruction is ignored.
- IM Insufficient memory. The source program is too large for assembly in available memory. Fatal.
- IR Illegal reference. An attempt was made to reference a location not on the current page or page zero and /O is not in effect. Non-fatal--the offending instruction is ignored.
- LG Link generated. /O is in effect without /Q and a link has been generated. Non-fatal.
- NE Null expression in a literal. Non-fatal--the literal is given a value of zero.
- NU Numerical error. An "8" or "9" was found when the radix was octal. Non-fatal--the expression is ignored.
- PA A pause psuedo-op has been reached. The remainder of the input file is ignored and the next input file is begun. This message requires no action by the user. If any expression followed the pause psuedo-op, its value is included in the PA message.
- PE Page exceeded. An attempt was made to overlay a link or literal with an instruction. Reduce the number of links, literals, or executable instructions on the current page. The message is only generated if /O and/or /Q are in effect. Fatal.
- PH Phase error. A conditional assembly was in effect at the end of the assembly. Check for mis-matched conditional assembly brackets ("<" and ">"). Fatal.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 6--PAL ASSEMBLER

- PO Pushdown overflow. An expression is too complex. Break the expression into several simpler expressions. Fatal.
- ST Symbol table overflow. Too many symbols for available memory. The symbol which caused the overflow is included in the message. Fatal.
- SY Syntax error. The assembler has detected illegal syntax in the source program. Non-fatal--the offending line is ignored.
- UF Undefined field. The argument of a FIELD psuedo-op contains an undefined value or a value greater than 7 octal.
- UO Undefined origin. An origin statement contains an undefined symbol. Non-fatal--the location counter remains unchanged.
- US Undefined symbol. A symbol has been encountered that is not defined in the current assembly. The octal address of the symbol in the source program appears in the error message. Non-fatal--the statement containing the undefined symbol is not assembled.
- ZE Page zero exceeded. Similar to PE except with reference to page zero.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 6--PAL ASSEMBLER

6.5 PAL LANGUAGE SUMMARY

IFDOS PAL is a superset of OS/8 PAL8. Statement syntax, character set, arithmetic and logical operators, numbers and symbols are exactly the same in both assemblers. IFDOS PAL accepts several pseudo-operators not recognized by PAL8, as well as offering more symbol table space and greater assembly speed in less memory.

Characters

IFDOS PAL recognizes the alphanumeric characters (A-Z,0-9) and the special characters in TABLE 6-4. All other characters (except when used in comments) are considered illegal and generate an "IC" error message. The characters "8" and "9" are only legal if the current assembler radix is decimal (see DECIMAL psuedo-op below).

Symbols

A symbol is one or more alphanumeric characters, beginning with a letter, representing a number or memory address between 0-4095 (0-7777 octal) inclusive. A symbol must either reside in the assembler's permanent symbol table or be defined within the user program as a tag or in a direct assignment statement. A symbol not defined in this manner is flagged with a "US" error message each time the assembler encounters it. A symbol may contain any number of characters, but it is identified by its first six characters only. Symbols with the same first six characters are considered identical.

Expressions

An expression is any sensible mathematical combination of defined symbols, numeric values and arithmetic and logical operators from TABLE 6-3.

TABLE 6-3

OPERATOR	OPERATION
+	Two's compliment addition
-	Two's compliment subtraction
*	Unsigned 12-bit multiplication
/	Unsigned 12-bit division
!	Boolean inclusive OR

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 6--PAL ASSEMBLER

& Boolean AND

The assembler performs an implied Boolean inclusive OR between symbols separated by spaces or tabs except if the leftmost symbol is a memory reference instruction.

(EXAMPLE: CLA OSR = LAS or 7200 7404 = 7604)

Statements

IFDOS PAL source programs are prepared with the system EDITOR on the console terminal. A program is a series of statements which take the following form:

TAG, INSTRUCTION OPERAND /COMMENT

A tag is a symbolic name which represents a memory address. A tag, if present, must begin with an alphabetic character, contain only alphanumeric characters, appear first in the statement, and be terminated by a comma.

An instruction is a defined symbol designating an operation to be performed when the program is executed, or a defined symbol designating an operation to be performed by the assembler during assembly. The latter is referred to as a psuedo-operation (abbreviated "PSUEDO-OP"). Instructions may be PAL mnemonics or psuedo-ops from TABLE 6-5, or they may be defined by the user in a direct assignment statement. Instructions may be separated from tags by one or more spaces or tabs.

An operand is an expression or defined symbol representing an address to be operated on by an instruction. Operands must be separated from instructions by one or more spaces or tabs.

Comments appear at the end of a statement. Any text may be used in a comment, but the comment must begin with a slash ("/"). Comments may be separated from operands by one or more spaces or tabs.

Statements must be terminated with a <CR> or ";". Any part of a statement may be omitted except the operand in a memory reference instruction. In fact, a source line may consist of just a <CR>, in which case the assembler skips a

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 6--PAL ASSEMBLER

line in the listing at that point.

TABLE 6-4 lists the special characters accepted by PAL, and their functions.

TABLE 6-4

CHARACTER	FUNCTION
=	Defines a symbol. (example: SYM=123)
,	Delimits a tag. (example: TAG, 123)
*	Sets the location counter. (example: *200)
.	Equal to the current value of the location counter. (example: JMP .-1)
"	Finds the ASCII code for the following character. (example: BEE="B")
()	Encloses a literal operand. PAL finds a location for the literal operand on the current page and makes the instruction reference that location. The right member of the enclosure may be excluded. (example: TAD (456)
[]	Similar to (), except PAL puts the literal on page 0.
/	Denotes the beginning of a comment.
;	Terminates a statement so two or more statements may appear on one line.
<>	Encloses a conditional assembly section. The right member must be present.
\$	Indicates the end of a source program. If \$ is omitted at the end of program, PAL inserts it in the assembly listing.

TABLE 6-5 lists the IFDOS PAL permanent symbol table. All psuedo-ops, which are preceded by asterisks, are described at the end of this chapter. Instructions are described in the documentation for the IM6100 microprocessor. Due to the changing nature of assemblers, some entries in the table below may not be correct. The actual permanent symbol table for any given version of IFDOS PAL may be obtained via the /A switch option.

TABLE 6-5

ACL	7701	GTF	6004	SQL	7421	SNA	7450
-----	------	-----	------	-----	------	-----	------

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 6--PAL ASSEMBLER

AND	0000	HLT	7402	NOP	7000	SNL	7420
BSW	7002	*I	1200	*NOPUNC	1143	SPA	7510
CAF	6007	IAC	7001	*OCTAL	0277	SRQ	6003
CAM	7621	*IFDEF	1601	OSR	7404	STA	7240
CDF	6201	*IFDOS	1144	*PAGE	2354	STL	7120
CIA	7041	*IFNDEF	1600	*PAUSE	2552	SWP	7521
CIF	6202	*IFNZRO	1615	RAL	7004	SZA	7440
CLA	7200	*IFZERO	1616	RAR	7010	SZL	7430
CLL	7100	IOF	6002	RDF	6214	TAD	1000
CMA	7040	ION	6001	*RELOC	3071	TCF	6042
CML	7020	ISZ	2000	RIB	6234	*TEXT	1301
*CONSOL	6372	JMP	5000	RIF	6224	*TEXTNZ	1300
*DATE	6373	JMS	4000	RMF	6244	TFL	6040
DCA	3000	KCC	6032	RTF	6005	*TITLE	3123
*DECIMA	0304	KCF	6030	RTL	7006	TLS	6046
*EJECT	3122	KIE	6035	RTR	7012	TPC	6044
*ENPUNC	1142	KRB	6036	SGT	6006	TSF	6041
*EXPUNG	1144	KRS	6034	*SKIP	3135	TSK	6045
*FIELD	0600	KSF	6031	SKON	6000	*XLIST	0364
*FIXMRI	0552	LAS	7604	SKP	7410	*Z	1144
*FIXTAB	1144	MOA	7501	SMA	7500	*ZBLOCK	1205
GLK	7204						

IFDOS PAL recognizes the following assembler directives (or "psuedo-ops"):

- CONSOL** Represents the binary value of the INTERCEPT switch register.
- DATE** Represents the value of the current system date word. The format of this word is described in CHAPTER 11.
- DECIMA** Changes the IFDOS PAL radix to decimal. All numerals in the source program are interpreted as decimal until the next OCTAL directive.
- DEVICE** Included for OS/8 PAL8 compatibility.
- EJECT** Causes PAL to begin a new page of the assembly listing. EJECT has no effect unless /N is in effect.
- ENPUNC** Reenables binary output after it has been disabled by NOPUNC.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 6--PAL ASSEMBLER

- FIELD** Advances the location counter to the next sequential memory field. FIELD may be followed by a number N, in which case the location counter is set to field N.
- FILENA** Included for OS/8 PAL8 compatibility.
- FIXMRI** Instructs PAL that the equate following the directive defines a memory reference instruction. For example:
FIXMRI TAD=1000
- FIXTAB** Instructs PAL that all previously equated symbols are permanent symbols for the current assembly. These symbols do not appear in the symbol table listing unless /A is in effect.
- I** Denotes an indirect memory reference. For example:
JMP I 0 means jump indirectly through 0.
- IFDEF** Begins a conditional assembly section. IFDEF must be followed by a symbol and a section of code enclosed by "<" and ">". If the symbol is defined in the current source program, the enclosed code is assembled. If not, the code is ignored. Syntax example:
IFDEF SYMBOL <JMP I 0>
- IFDOS** Used with IFDEF and IFNDEF, so that parts of source programs can be recognized by IFDOS PAL and not by other assemblers. Since IFDOS is a permanent symbol, the statement "IFDEF IFDOS" is always true under IFDOS PAL.
- IFNDEF** Similar to IFDEF, except the enclosed code is assembled only if the symbol is not defined.
- IFNZRO** Similar to IFDEF, except the enclosed code is only assembled if the symbol following IFNZRO is defined as not equal to zero.
- IFZERO** Similar to IFDEF, except the enclosed code is only assembled if the symbol

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 6--PAL ASSEMBLER

following IFZERO is defined as equal to zero.

- NOPUNC Inhibits binary output to the object file until the next ENPUNC directive.
- OCTAL Changes the IFDOS PAL radix to octal until the next DECIMAL directive.
- PAGE Advances the location counter to the next sequential memory page. PAGE may be followed by a number N, in which case the location counter is set to page N.
- PAUSE Causes PAL to ignore the remainder of the current input file and begin the next. PAUSE may be followed by an expression, in which case the value of the expression is typed on the terminal during assembly.
- RELOC If followed by an expression, RELOC begins a relocation section by establishing a virtual location counter with the value of the expression. The code in the relocation section is loaded into the area of memory defined by the actual location counter, but it is assembled to execute in the memory area defined by the virtual location counter. Generally, the code must be moved at run time for correct execution. RELOC without an expression following it ends the relocation section. Relocation sections appear in the program listing with asterisks after the location counter. For example:

```
          *200  
          TAD       TAG  
          RELOC 300  
TAG.       1234  
          RELOC
```

"TAG" loads into location 201, but references to TAG reference location 300. The relocation section must be moved to 300 before it can be referenced correctly from other parts of the program.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 6--PAL ASSEMBLER

SIXBIT The delimited ASCII text string following the directive is converted to a special form of six bit ASCII code. An octal 40 is subtracted from each 7-bit character and the resulting codes are packed as in the TEXT directive below.

SKIP Followed by a number N. PAL skips N lines in the assembly listing.

TEXT The delimited ASCII text string following the directive is converted to six-bit ASCII and packed two characters per word by PAL. The string is ended by a byte or word of zeroes (unless /J is in effect). For example:

```
*200  
TEXT /ABC/
```

```
0200 0102  
0201 0300
```

TITLE When /N is in effect, the text following the directive becomes the new title of the program and appears at the top of each page of the listing.

XLIST Inhibits assembly listing if the expression following the directive is equal to zero. If the expression does not equal zero, assembly listing is enabled.

Z Has no meaning to IFDOS PAL, but is included for compatibility with older assemblers.

ZBLOCK Creates a block of zeros in the object file equal in length to the value of the expression following the directive.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 7--BIN LOADER

The BIN loader accepts binary object files produced by the PAL assembler, loads these files, and starts them at the user specified starting address. Alternatively, by switch option, input and output may be via the high or low speed paper tape reader and punch.

When BIN has loaded each of the input files, it transfers control to the user supplied starting address. If no starting address is supplied, BIN halts and further action must be taken at the computer front panel. Normally, BIN does not load a program into locations 07600-07777.

7.1 HOW TO USE BIN

To run the BIN program, type a command of the following form to the KEYBOARD MONITOR:

```
.BIN OFILES<IFILES (SWITCHES)=SA
```

"OFILES" are binary files produced by BIN if input was from the paper tape reader.

"IFILES" are the binary object files of the PAL assembler.

"(SWITCHES)" are BIN switch options from TABLE 7-1.

"=SA" is an octal starting address in the range 0-7777.

7.2 BIN SWITCHES

TABLE 7-1

- /B* Take input from % scratch file (% is the default input file for BIN).
- /D* Use % and \$ for input.
- /F* Used with /P and /W. Designates high speed reader and punch for paper tape input and output.
- /G* If used with /P, chain back to loader for additional paper tapes. Has no meaning if used separately.
- /H* Pre-load memory with halts (7402).
- /I* Use fast loader--reload system I/O routines after loading user files. This option limits file speci-

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 7--BIN LOADER

fications to 16 (rather than 17).

- /P Load RIM and BIN paper tape loaders into memory, and take input from paper tape reader (high speed if used with /F). Field for loading may be specified by /1-/7. If used with output files, paper tapes are transferred to output files. Field settings in these files may be specified with /1-/7.
- /R* Take paper tape input in RIM format. Used with /P.
- /U* Use unit 1 for input and output.
- /V* Load virtually. /V loading is more time consuming than /I loading. /V recognizes binary files and does not load source or memory image files.
- /W* Produce paper tape output in BIN format.
- /Z* Pre-load memory with zeros.
- /1-/7* Field setting for loading. Field 0 is default.
- =n Octal address at which to start. Must be specified or BIN halts upon completion of loading.

* switches marked by asterisks may be passed to BIN from the PAL assembler.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 8--OCTAL DEBUGGING TECHNIQUE

8.1 HOW TO USE ODT

The octal debugging technique allows the user to run a program, examine and modify the contents of memory, and control program execution from the terminal keyboard. ODT is virtually invisible to a user program, since it requires only locations 07600-07777 (normally reserved for system use) and locations 4 and 7756-7777 in any field where an active breakpoint resides.

When ODT is used to debug a program, the program must have been loaded with the GET command before ODT is called. For example:

```
.GET PROG,PROG2  
.ODT
```

ODT generates a <CR/LF> and awaits user action. At this point field 0 of the user's program resides in the % and \$ scratch files (the GET command reads binary files and produces memory images in the scratch area). ODT accepts commands from the keyboard and performs these operations as if the program was in memory. When the user starts the program, ODT first swaps the program into memory, then begins execution. If any breakpoints were set prior to starting the program, execution stops at the breakpoint, the user's program is swapped out, and ODT resumes control and accepts keyboard commands.

When the user sets a breakpoint at a memory location, ODT traps the instruction residing there so control returns to CDT just prior to executing the instruction. The trapped instruction is the first to be executed when the user program is resumed. While at a breakpoint, the user may obtain intermediate results, diagnose problems, and devise temporary corrections before continuing the program.

8.2 ODT COMMANDS

ODT commands fall into five categories. These are: Location examination, internal register examination, program control, breakpoint, and search.

Whenever a memory location or internal register is examined, it is said to be open. The user may modify and close the location, or close it without modification. Only

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 8--OCTAL DEBUGGING TECHNIQUE

one location may be open at a time. Opening a new location automatically closes the currently open location, if any.

When ODT encounters an illegal command or a command which does not make sense, a "?" is typed and ODT ignores the command. Typing <RUB> causes ODT to print a "?", ignore the command preceding the <RUB>, and await further commands.

The following is a list of ODT commands (all command arguments are in octal):

LOCATION EXAMINATION COMMANDS

nnnnn/ Open location nnnnn, where nnnnn is between 0-77777 inclusive. The contents of location nnnnn are printed. If the user types a number on the same line and closes the location, the new number replaces the previous contents of location nnnnn. ODT does not reference non-existent or non-contiguous memory.

<CR> Close the currently open location.

<LF> Close the current location and open the next location.

SHIFT/N (uparrow) Close the current location and use its contents as a memory reference instruction. Open the referenced location. If the original location was modified by the user, its new contents are used by SHIFT/N.

SHIFT/O (backarrow) Similar to SHIFT/N, except the contents of the current location (after it is closed) are used as the 12-bit address of the next location to open.

nnnn+ Close current location, use nnnn as positive offset from current location address and open that location. nnnn must be between 0-7777 inclusive.

nnnn- Similar to nnnn+ command, except nnnn is negative offset.

; Similar to <LF>, except that the contents of each succeeding location are not printed. This command provides a quick way of modifying several sequential locations on one command line. If N semicolons are typed without intervening arguments, N-1 locations are skipped and left unmodified.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 8--OCTAL DEBUGGING TECHNIQUE

INTERNAL REGISTER EXAMINATION COMMANDS

ODT maintains several internal registers and tables accessible to the user with the following commands. These commands open locations just as the "/" command, but they reference specific locations and therefore take no arguments.

- A Open the register containing the user program's accumulator during a breakpoint.
- D Open the register containing the user program's data field during a breakpoint.
- F Open the register containing the field used by ODT for the ^ and _ location examination commands, and for the W and X searches (see search commands below).
- L Open the register containing the user program's link during a breakpoint.
- M Open the first word of the search parameter table. This three entry table helps the user define a particular value during the W and X searches (see the W search command for a description of the search parameter table).
- N Open the breakpoint identification register (or N register). Since ODT has multiple breakpoints, the contents of the N register determine which breakpoint is affected by the breakpoint commands.

PROGRAM CONTROL COMMANDS

The commands in this section initiate user programs under ODT control, or resume user program execution after a breakpoint. Once a user program is started, ODT will not accept further commands until a breakpoint is reached.

nnnnG Clear AC and link and begin program execution at location nnnn.

nnnnS Similar to G with the following differences:

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 8--OCTAL DEBUGGING TECHNIQUE

Breakpoints in the user program are not recognized. Any file specifications and switch options specified on the ODT command line (when ODT was called from the KEYBOARD MONITOR) reside in parsed form in the system tables (system tables are discussed in CHAPTER 11). The S command is useful in debugging programs which perform file manipulation.

- kC Continue execution from the current breakpoint and set the iteration counter for that breakpoint to k. The instruction trapped by the breakpoint must be executed k-1 times before ODT again comes to a breakpoint. If k is omitted, k=1 is assumed.
- nnnnR Resume execution from location nnnn. If nnnn is omitted, the address of the current breakpoint is assumed.

BREAKPOINT COMMANDS

ODT supports numerous commands for setting and deleting up to 15 breakpoints, and performing associated operations.

Each breakpoint has a unique number between 1-17 (octal) inclusive and a four word entry in ODT's breakpoint table. Breakpoint commands may reference any of the 15 breakpoints. ODT's breakpoint identification register (or N register) may be used to select a breakpoint for a given operation. Also, some of the commands below take an argument which consists of a breakpoint number (1-17) followed by a comma. When an argument of this form is given, the N register is changed to the value preceding the comma. The breakpoint whose number is in the N register is called the default breakpoint. When ODT is first called, the default breakpoint is one. This means that if no breakpoint number is explicitly designated in one of the commands below, the default breakpoint is used. When a breakpoint occurs during program execution, that breakpoint becomes the default breakpoint.

- nnnnB Set the default breakpoint at location nnnn.
- b,nnnnB Set breakpoint b at location nnnn.
- B Remove default breakpoint.
- b,B Remove breakpoint b.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 8--OCTAL DEBUGGING TECHNIQUE

- kK Set iteration counter of the default breakpoint to k. This command is similar to the C command except program execution does not continue.
- b,kK Set iteration counter of breakpoint b to k.
- b,kC Similar to b,kK , except that program execution continues from the current breakpoint.
- K Open the iteration counter of the current breakpoint.
- b,K Open the iteration counter of breakpoint b.
- I Add one to the N register, thereby changing the default breakpoint.

SEARCH COMMANDS

ODT may be used to search through all or part of memory for specific words or bit patterns. A three entry search parameter table is used to control searching operations.

Word one of the table contains the search mask. During a search, sequential memory locations are bitwise "ANDED" with the search mask. The result of the logical AND determines whether the memory word matches the word being searched for. Word two of the search parameter table holds the lower search limit. Searching begins at the lower search limit and progresses to the upper search limit. Word three of the table contains the upper search limit.

The M command opens the search mask for modification. With the search mask open, a line feed opens the lower search limit and another line feed opens the upper search limit. When ODT is called from the KEYBOARD MONITOR, the search parameter table is as follows:

```
search mask=7777
lower search limit=0000
upper search limit=7577 (below resident system code).
```

Word searching is always done in the field specified by the "F REGISTER" (which may be accessed via the F command).

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 8--OCTAL DEBUGGING TECHNIQUE

ODT has two search commands:

nnnnW Search for and type every location within the search limits which, when logically ANDED with the search mask, equals nnnn.

nnnnX Similar to the W search, except all words not matching nnnn are typed.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

BATCH

IFDOS BATCH accepts input files of KEYBOARD MONITOR, EDITOR, and SYSTEM COMMANDS and executes the commands in these files as if they were typed separately. The EDITOR is used to prepare BATCH command files. Long or frequent command procedures are typed at the EDITOR, saved in a user file, and run with a single command. BATCH jobs often need no operator intervention and may be left unattended.

As the BATCH job executes, each command in the command file is typed on the terminal and executed. Error messages, if any, are typed on the terminal. If an error condition develops which prohibits BATCH from completing a command, the next command in the BATCH input file is executed.

BATCH files are prepared with the system EDITOR, just as PAL source programs. BATCH, like PAL, ignores EDITOR line numbers. BATCH accepts any command which the KEYBOARD MONITOR recognizes. This includes KEYBOARD MONITOR commands, EDITOR commands, SYSTEM COMMANDS, and extended SYSTEM COMMANDS (discussed in CHAPTER 10). BATCH command syntax is exactly the same as the usual command syntax, except for the presence of an EDITOR line number on each command line. Filenames (up to 17) and switch options may be included in BATCH commands.

BATCH allows the user to designate a special character to be used as an "END OF SEQUENCE" character during BATCH execution. When the special character (if none is designated, backslash ("\") is assumed) is encountered by the EDITOR during a BATCH job, sequence mode is cancelled. CTRL/C and CTRL/D cannot be used to cancel sequence mode under BATCH because they have an immediate effect and will cancel sequence while the BATCH file is being edited.

BATCH is called from the KEYBOARD MONITOR as follows:

.BATCH INPUT FILES/X=N

"FILES" are BATCH input files prepared with the EDITOR.
"/X" is a switch option from TABLE 9-1.

TABLE 9-1

/A Use the contents of the EDITOR buffer as the input to BATCH. No input file specifications are needed

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

with /A.

- /E Run "subset" BATCH. Monitor overlays, such as the "FIND", and "CHANGE" SYSTEM COMMANDS can only be run from subset BATCH.
- /N Do not scratch the EDITOR buffer before starting BATCH. /N insures that the previous contents of the EDITOR buffer are available under BATCH control.
- /R Allow "R" and "RUN" commands in the BATCH stream.
- /X Use the number following the "=" as the six-bit ASCII code of the special end of sequence character.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

BSAVE

The BSAVE command creates binary files from virtual memory images. When a binary file is loaded with BIN or GET, a virtual memory image is created in the system scratch files. BSAVE produces a binary file using the scratch file contents and arguments specified by the user. These arguments designate the areas of memory to be saved. BSAVE accepts no input files.

The BSAVE command is useful after an ODT session when the user has modified a program and wishes to save the modified program as a binary file.

To run BSAVE, type:

```
.BSAVE OFILES<A,B,C-D,...
```

"OFILES" are output files, separated by commas, used by BSAVE for binary output.

"A,B,C-D,..." is the form of the arguments designating memory areas to be saved. For example, 0,77,200-300,500 saves locations 0, 77, 200 to 300, and location 500.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

CHANGE

The CHANGE command searches for and modifies ASCII text in the EDITOR buffer. The command takes the following form:

.CHANGE *OLDTEXT*NEWTEXT

"*" represents a delimiter character, which may be any printing six-bit ASCII character not used in the text string arguments.

"OLDTEXT" is the string of characters to replace.

"NEWTEXT" is the new string to be inserted in the buffer.

The command searches for the first occurrence of "OLDTEXT" and replaces it with "NEWTEXT". CHANGE accepts one switch option, /A, which means replace all occurrences of "OLDTEXT" with "NEWTEXT". Also, an "=N" parameter may be specified, where N is the line number at which to start making changes. For example:

.CHANGE /A=100"OLD"NEW

Changes all occurrences of "OLD" to "NEW" from line 100 to the end of the buffer.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

DATE

The DATE command sets the system date and the message of the day. The PAL assembler uses the system date in the header of a program listing. DATE is run as follows:

.DATE MM/DD/YY+MESSAGE

Typing the DATE command without arguments causes the system to type the current date followed by the message of the day, if any. To remove the message of the day, type:

.DATE /R

Certain holidays have messages associated with them. These messages are stored in a table on mass storage. They supercede user messages, and cannot be removed with /R. When the user enters a message it becomes a part of the table and remains until /R is given.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

DUMP

The DUMP program provides facilities for examination, modification, and transfer of absolute blocks on mass storage. DUMP has three modes: DUMP, ZAP, and TRANSFER. When DUMP is called, it asks which mode is to be used. User response may be abbreviated to the two letters "DU", "ZA", or "TR". DUMP treats each mode differently after the "MODE?" question.

DUMP MODE

DUMP mode is used to list the octal and ASCII contents of any block or group of blocks on mass storage. DUMP mode is used as follows:

```
.DUMP          (DUMP is called from the KEYBOARD
                MONITOR)
MODE? DU       (DUMP mode is selected)
BLOCK =        (device and first block to DUMP are
                selected)
NUMBER=        (number of blocks to be dumped
                is typed)
```

DUMP begins typing the contents of the specified block. Striking a key causes DUMP to skip to the next block, if more than one block is being dumped. Striking CTRL/G causes DUMP to return to the "MODE?" question. Switch register switches (5-11) control the relative location within each block where dumping begins.

EXAMPLE 9-1

```
.DUMP
MODE? DU
BLOCK =1:400
NUMBER=10
```

Blocks 400-410 on logical device 1 are typed on the terminal.

ZAP MODE

The ZAP mode of DUMP is used to examine and modify mass storage on a block by block basis. ZAP mode is invoked as

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

follows:

```
.DUMP          (DUMP is called from the KEYBOARD
               MONITOR)
MODE? ZA      (ZAP mode is selected)
BLOCK =       (device and block to be modified are
               selected)
```

Dumping of the specified block begins with the relative location in switch register bits (5-11). Striking a key causes DUMP to print:

```
NUMBER=
```

Type the location to be changed, followed by <CR>, and the new contents of the location. Succeeding locations may be changed by typing commas between values. The <CR> following the location and the commas between values are echoed as spaces by DUMP. A line is terminated with a second <CR>, and DUMP types "NUMBER =" again. Other locations may be changed or CTRL/G may be typed to make the changes specified and return to the "MODE?" question.

EXAMPLE 9-2

```
.DUMP
MODE? ZA
BLOCK = 1:400
```

(block 400 is dumped until a key is struck.)

```
NUMBER= 0<CR>10,11,12,13,14,15,16,17
        (the <CR> and commas echo as spaces)
        ^G
```

```
MODE?
```

Block 400 on device 1 is selected for modification. The block is dumped until a key is hit, and locations 0-7 of block are changed to 10-17. CTRL/G is typed to return to the "MODE?" question.

TRANSFER MODE

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

TRANSFER mode copies absolute blocks. Copying may be from one device to another of the same type, from one device back to itself, or from a device back to another on the same drive. If the latter is being done, switch mass storage media when dump types "SWITCH". TRANSFER mode is invoked as follows:

```
.DUMP          (DUMP is called from the KEYBOARD
                MONITOR)
MODE? TR      (TRANSFER mode is selected)
BLOCK =       (input device and starting block
                selected)
NUMBER=       (number of blocks to transfer
                (up to 34))
(Here the input is made, DUMP types "SWITCH", and
types:)
BLOCK =       (select the output unit and block)
```

EXAMPLE 9-3

```
.DUMP
MODE? TR
BLOCK =1:400
NUMBER= 20 SWITCH
BLOCK = 0:400
```

Blocks 400-420 on device 1 are transferred to blocks 400-420 on device 0.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

FIND

The FIND command searches the EDITOR buffer for ASCII text strings and types lines containing the specified text on the terminal. An ASCII file must have been loaded into the EDITOR buffer before FIND can be used. FIND accepts no input files. FIND is used as follows:

```
.FIND *TEXTSTRING<CR>
```

"*" is a text delimiter which may be any printing six-bit ASCII character not found in the text.

"TEXTSTRING" is the text to search for.

This command finds the first occurrence of "TEXTSTRING" in the buffer and types the line containing it on the terminal. If /A is specified to FIND, all occurrences of the text string are typed. If the "=N" parameter is included, searching begins at line N.

EXAMPLE 9-4

```
.FIND /A=100 "TEXT<CR>
```

All lines containing "TEXT" from line 100 to the end of the EDITOR buffer are typed on the terminal.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

GET

The GET command is a special entry point of the BIN loader, which accepts most loader switches, loads a file virtually into the system scratch files, and returns to the KEYBOARD MONITOR. GET is most often used to load a program in preparation for an ODT session. To use the GET command, type:

.GET IFILES(SWITCHES)

"IFILES" are binary input files

"(SWITCHES)" are binary loader switch options from TABLE 7-1.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

HILOAD

The location of the IFDOS system I/O routines makes it impossible for the BIN program to load memory locations 07600-07777. However, users simulating ROM based IM6100 systems with the INTERCEPT often need locations 07600-07777 for their application program. In a ROM based system locations 00000-00377, if they exist, are generally used for RAM and therefore must be initialized by the ROM program before being used by that program. The HILOAD program uses locations 00000-00377 to load locations 00400-07777 of the user's program.

HILOAD assumes that a binary file has previously been loaded (virtually) into the % and \$ scratch areas with the GET command.

To use HILOAD, first GET the binary program as described earlier in this chapter. When IFDOS returns to the KEYBOARD MONITOR, type:

.HILOAD=N

"=N" specifies the starting address of the user program in field zero. If "N" is not specified, HILOAD halts and further action must be taken at the INTERCEPT front panel.

This command loads the contents of the % and \$ scratch areas into memory field zero and transfers control to the specified starting address, if any. HILOAD normally loads from the unit zero scratch files unless the switch option "/1" is specified. HILOAD accepts no other switch options or file specifications.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

MEMORY

IFDOS provides a means of software memory protection via the MEMORY command. Any memory above field 0 may be protected by setting the software memory size to the highest usable field. Some programs, such as the PAL assembler, check the software memory size and do not use memory above the software memory size. To set the software memory size, type:

.MEMORY N

where N is the highest memory field to be used. If N is omitted, the system types:

AK ACTUAL/BK AVAILABLE/CK USABLE

A, B, and C are replaced by appropriate values. Actual memory is the memory which physically exists on the system. Available memory is the upper limit of the software memory size, determined at system generation. Usually, available memory is equal to actual memory--see CHAPTER 11 for a more precise definition of available memory. Usable memory is set by the MEMORY command.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

START

The START command is used to start a program after loading it with GET. An "=N" parameter must be included with START, where n is the starting address of the program. START is a special BIN entry point, and accepts no switch options. To start a program, type:

.START =N

"N" is the desired starting address in the range 0-7777.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

VARY

Some system parameters, such as system radix, tab handling, and keyboard echoing may be modified with KEYBOARD MONITOR commands. When the system is rebootsrapped, however, these parameters revert to their default values. The VARY program allows the user to set new default values for the parameters above and many others.

The system radix, for example, may be set to octal or decimal by the "RADIX" KEYBOARD MONITOR command. By default, the radix is decimal, and it returns to decimal when the system is rebootsrapped. However, if VARY is used to change the radix to octal, the default radix becomes octal. The RADIX keyboard monitor command may still be used to change the radix, but rebootsrapping returns the radix to octal.

A list of parameters which may be changed with VARY is typed on the terminal when the /A switch is used as follows:

```
.VARY /A
```

Table 9-2 also lists VARY options. The left column lists the keywords of the parameters which may be changed. To the right of each keyword is a list of options available. To vary a parameter, type a command of the form:

```
.VARY KEYWORD OPTION
```

"KEYWORD" is a parameter keyword from the /A list.
"OPTION" is an available option for the given keyword.

For example, to make the radix parameter octal by default, type:

```
.VARY RADIX 8 or .VARY RADIX OCTAL
```

Options marked by asterisks are in effect when IFDOS is delivered and remain in effect until changed by the user. A password is required to change some VARY options. Initially, the password is IFDOS. See below for proper syntax.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

TABLE 9-2

KEYWORD	OPTIONS			
LF	ECHO	*VIEW	END	IGNORE
DOLLAR	LOAD	*NOLOAD	SAVE	*NOSAVE
LEADER	*LONG	SHORT	*ON	OFF
RADIX	8	OCTAL	*10	*DECIMA
TTY	*33	35	BLANK	TAB NOFILL
	FILL	SPACE	SPACES	*SIMULA
PUNCH	CODE	33	35	BLANK TAB
	*NOFILL	FILL	SPACE	SPACES SIMULA
ASSUME	*ON	OFF		
READER	CODE			
ALTMOD	CODE			
UNITS	*ON	OFF		
ECHO	*ON	OFF		
PASS	WORD			

Most of the options are self-explanatory. A "CODE" option allows the user to change a device or character code, whichever the case may be.

For example, to change the reader device code from 601x to 651x, type:

```
.VARY READER CODE=6510
```

Some keywords and options require a password. "UNITS" is one such keyword. For example, to restrict a user to a single device, type:

```
.VARY UNITS OFF:IFDOS
```

The password must be specified, preceded by a colon. To change the password, type a command of the form:

```
.VARY PASS WORD:OLDPAS:NEWPAS
```

A list of the less obvious keyword-option combinations and their meanings follows. Keywords are underlined, and followed by options and brief descriptions.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 9--OTHER SYSTEM COMMANDS AND UTILITIES

TABLE 9-3

<u>LF</u>	
ECHO	echo <LF>
VIEW	view KEYBOARD MONITOR buffer contents
END	<LF> treated as <CR>
IGNORE	<LF> ignored, for reading paper tapes
<u>DOLLAR</u>	
LOAD	automatically loads \$ file on bootstrap
SAVE	automatically saves \$ when program runs
<u>LEADER</u>	
LONG	two foot leader/trailer
SHORT	one inch leader/trailer
<u>TTY</u>	
33	tabs simulated by spaces
35	tabs followed by appropriate number of nulls
BLANK	tabs replaced by one space
TAB	tabs output as tabs
NOFILL	same as TAB
FILL	same as 35
SPACE	same as BLANK
SPACES	same as BLANK
SIMULA	same as 33
<u>ASSUME</u>	
ON	remember last file name for LOAD, SAVE, ENTER, DELETE
<u>UNITS</u>	
OFF	do not allow colons in keyboard monitor commands

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 10--EXTENDED SYSTEM COMMANDS & UTILITIES

Extended SYSTEM COMMANDS and utilities are memory image programs which come after the SYSTEM COMMANDS on mass storage. Extended SYSTEM COMMANDS are usually seldom used or user written programs. They perform in exactly the same way as SYSTEM COMMANDS, but the extended SYSTEM COMMAND catalogue is separate from the SYSTEM COMMAND catalogue and extended SYSTEM COMMANDS are invoked differently. The general form of an extended SYSTEM COMMAND is:

.R PROGRM

"PROGRM" is an extended SYSTEM COMMAND with an entry in the extended SYSTEM COMMAND catalogue. See CHAPTER 11 for the location of this catalogue and information on implementing user extended SYSTEM COMMANDS.

IFDOS currently supports three extended SYSTEM COMMANDS: SYSTYP; SYSCPYP; and FORMAT.

SYSTYP

The SYSTYP program types the contents of the system command catalogue, and optionally, the extended SYSTEM COMMAND catalogue on the terminal.

SYSTYP is run as follows:

.R SYSTYP

Setting SR(0) equal to 1 causes SYSTYP to print only the SYSTEM COMMAND catalogue. Setting SR(0)=0 causes SYSTYP to print both the SYSTEM COMMAND catalogue and the extended SYSTEM COMMAND catalogue.

SYSCPYP

The SYSCPYP program copies absolute blocks from one logical unit to another, and optionally verifies the result of the copying operation. To run SYSCPYP, TYPE:

.R SYSCPYP(SWITCHES)

"(SWITCHES)" are SYSCPYP switch options, as follows:

/A Modifies the dialogue below so the message
"LAST INPUT BLOCK" becomes "NUMBER OF BLOCKS".

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 10--EXTENDED SYSTEM COMMANDS & UTILITIES

/C Allows SYSCPY to violate the software memory size and use all available memory for greatest throughput.

When the command is terminated, SYSCPY replies:

INPUT
OUTPUT
LAST INPUT BLOCK
VERIFY ?

Input and output devices and block numbers are specified as "U:B" where "U" is the logical device, followed by ":", and "B" is the starting block number of the transfer. The last input block is an octal number between 0 and the limits of the mass storage device. When the user types "Y" or "N" on the last question the copy operation begins. All responses except the last must be terminated with "<CR>".

The user may repeat a copying operation any number of times without having to respecify the blocks being copied. When SYSCPY types "DONE !", hitting "CTRL/S" repeats the exact operation. The user may change either or both mass storage media before hitting "CTRL/S". This feature is useful when multiple copying operations are done.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 11--IFDOS DETAILED SYSTEM INFORMATION

This chapter explains the IFDOS monitor system in detail, including system memory use, system device layout, file formats, system I/O calls, etc. Before reading this chapter, a thorough understanding of the preceding chapters is very important.

SYSTEM MEMORY USE

The operation of IFDOS is largely invisible to the user. When the system is bootstrapped, it occupies all of field 0. However, the permanently resident portion of the system occupies only locations 07600-07777. User programs may load into locations 0-07577, and when the user program is finished, the system may be rebootstrapped by starting at location 07600. Of course, if a user program alters any locations between 07600-07777, a "cold-start" bootstrap must be used to bring in the system. (see APPENDIX A.)

TABLE 11-1 shows the memory areas used by various system components. Some parts of the system are implemented as overlays to the basic structure as shown in TABLE 11-1. These actions are rather complex, and are not discussed here to avoid confusing the reader.

TABLE 11-1

LOCATIONS	CONTENTS
00000-02527	KEYBOARD MONITOR & EDITOR
02530-02577	KEYBOARD MONITOR buffer
02600-02777	EDIT command & special EDITOR commands
03000-06777	EDITOR text buffer
07000-07377	current user catalogue
07600-07777	resident system code

Locations 0-06777 contain the components of the system discussed in the early chapters of this manual. (i.e., KEYBOARD MONITOR, EDITOR, etc.) These components should be understood from a user's point of view. Even the sophisticated user need not know the intricacies of their operation.

Locations 07000-07377 contain the user catalogue of the current system device. Each user file on the system device is represented in this catalogue by a six-character (3 word) entry. The last word in the user catalogue is the block

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 11--IFDOS DETAILED SYSTEM INFORMATION

number at which user files begin. Empty spaces in the user catalogue are filled with zeros. Since all user files are 20 octal blocks long, the position of a file on mass storage is easily determined by its location in the catalogue.

RESIDENT SYSTEM CODE--SYSTEM TABLES

TABLE 11-2 outlines the memory resident portion of IFDOS. The system I/O routines and system tables, in locations 07600-07777, greatly simplify logical device I/O and provide other important information.

TABLE 11-2

LOCATIONS	CONTENTS
07600-07603	system rebootstrap code
07604	A B C D E F G H I J K L
07605	M N O P Q R S T U V W X
07606	Y Z 0 1 2 3 4 5 6 7 8 9
07607	output file count
07610	system date word
07611	software memory size
07612-07755	system I/O routines*
07756	"=" parameter value
07757-07777	starting blocks for all files specified

*Part of the INTERCEPT D10 system I/O routines reside in control panel memory.

When the user types a SYSTEM COMMAND (or extended SYSTEM COMMAND) all file specifications and switch options included with the command are decoded by IFDOS and stored in the system tables. The system looks up each filename in the order in which it was specified, and stores its starting block number in the table beginning at location 07757. The total number of output files specified is stored at 07607. All switch options typed are represented by set bits in locations 07604-07607. The value of the "=" parameter, if any, is stored in location 07756. The system sets all unused table entries to zero, except 07756, which is set to 7777 by default.

SYSTEM COMMANDS (and extended SYSTEM COMMANDS) are the only commands which accept file specifications and switch options as arguments. Therefore, user programs which perform file manipulation should eventually be implemented

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 11--IFDOS DETAILED SYSTEM INFORMATION

as extended SYSTEM COMMANDS (this procedure is outlined later in this chapter). During the debugging stage, however, file specifications and switch options may be passed to programs via ODT.

After the program is assembled and the binary program resides in a user file, the following commands are used:

```
.GET PROG  
.ODT OFILES<IFILES(SWITCHES)  
nnnnS
```

"PROG" is a binary file containing the user program.
"OFILES<IFILES (SWITCHES)" are file specification and switch options to be interpreted by the user program.
"nnnnS" is the ODT command to start the program at location nnnn, with the system tables intact and no breakpoints.

Location 07611 contains the software memory size. This is the number of the highest usable memory field (as set by the "MEMORY" SYSTEM COMMAND) multiplied by 10 octal. For example, if field 2 is the highest usable field, the software memory size is 20.

Location 07610 contains the system date word. Dividing this value by 564 octal gives the years past 1976. The remainder of the first division is divided by 14 octal to obtain the month, and the remainder of this division gives the year. For example:

```
7/4/77 = 674 octal  
674 / 564 = year past 1976 + remainder = 1 + 110  
110 / 14 = month + remainder = 7 + 4  
4 = day
```

RESIDENT SYSTEM CODE--SYSTEM I/O ROUTINES

The main entry point of the system I/O routines is at 07640. The system I/O routines perform input/output between memory and mass storage on a block by block basis. A call to the system I/O routines must include three arguments, as follows:

```
CDF n           /DATA FIELD = BUFFER FIELD  
CIF 00         /ALWAYS FIELD ZERO  
JMS I   K7640  /ENTRY POINT = 7640
```

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 11--IFDOS DETAILED SYSTEM INFORMATION

	/AC DOESN'T MATTER
ADDR	/STARTING ADDRESS OF TRANSFER
FUNCT	/FUNCTION WORD (SEE BELOW)
BLOCK	/STARTING BLOCK OF TRANSFER

Arguments one and three are self-explanatory. Argument two, the function word, is broken down as follows:

<BIT 0>	<BITS 1-5>	<BITS 6-8>	<BITS 9-11>
0 = INPUT	# OF BLOCKS	FIELD OF	LOGICAL DEVICE
1 = OUTPUT	0-37 (0=40)	TRANSFER (1-7)	NUMBER (1-7)

Bit 0 of the function word controls the direction of the I/O transfer with respect to main memory. Bits 1-5 specify the number of blocks to transfer. If these bits are zero, a full field (40 pages or blocks) is transferred. Bits 6-8 of the function word name the memory field, and bits 9-11 name the logical device for the transfer.

If an error condition occurs during a transfer, the system halts. Pressing continue on the computer front panel causes a retry of the I/O operation. Attempting I/O to a nonexistent device, or attempting output to a write protected device results in an error. When an I/O transfer is complete, control returns to the user program at the instruction immediately following argument three.

FILE FORMATS

IFDOS uses three basic file formats. User files are either ASCII or binary format, and system files are in memory image format. ASCII files contain two characters per word. With two exceptions, characters are represented in standard six-bit ASCII. The <CR> character has octal code 00 and the <TAB> character has octal code 37. Consequently, neither the SHIFT/P ("@") nor the SHIFT/O ("_") may be represented in an ASCII file. Characters are left justified. ASCII files created by the EDITOR contain line numbers at the end of the file. Each text line in the file is represented by a two-word entry. The first (working from the end of the file to the beginning) is the octal line number and the second is the pointer to the line in the text buffer.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 11--IFDOS DETAILED SYSTEM INFORMATION

Binary files contain eighteen 7-word data frames, plus two extra words per block. The first word of each data frame identifies the type of data in the six successive words. This frame control word is divided into six bit-pairs. Each bit-pair characterizes a word in the data frame. The bit-pairs are defined as follows:

BINARY	MEANING
00	word of code--load & advance location counter
01	end of file--don't load this word
10	origin setting--set location counter
11	field setting--execute this CDF

Memory image files are self explanatory. The first word in the file corresponds to the first memory location used by the program, and successive words in the file correspond to successive memory locations. Memory image files are the most efficient means of program storage.

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 11-IFDOS DETAILED SYSTEM INFORMATION

THE SYSTEM DEVICE

The IFDOS system device contains a complete copy of the monitor system (KEYBOARD MONITOR, resident system code, etc.), and system programs, as well as scratch files and space for user files. Generally, blocks 0-14 and 72-100 are for system use only. Any modification of these blocks by the user may impair the proper operation of the system. The basic layout of the system device is described in TABLE 11-3.

TABLE 11-3

BLOCKS	CONTENTS
0	Resident system code (memory locations 07600-07777)
1-14	KEYBOARD MONITOR and EDITOR
15-16	User file catalogue
17	Reserved
20-37	% scratch file
40-57	\$ scratch file
60	SYSTEM COMMAND catalogue
61	The R command
62	The RUN command
63-65	Extended user's catalogue
66-71	Extended SYSTEM COMMAND catalogue
72-100	Used by BATCH
101-377	SYSTEM COMMANDS, extended SYSTEM COMMANDS
400-...	User files

SYSTEM COMMAND CATALOGUE

The SYSTEM COMMAND catalogue, in block 60, contains one six-word entry for each SYSTEM COMMAND. Words 1-3 contain the command name used to call the program. Word 4 contains the first block the program occupies on mass storage. Word 5 contains the memory load address (bits 0-5) and the length (bits 6-11) in blocks or memory pages. Word 6 contains the starting address of the program when it is loaded into memory. This address always refers to field 0.

A five-page program named PROG, at block 300, which loads into locations 00000-01177 and starts at 00200 would be represented in the catalogue as follows:

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
CHAPTER 11-IFDOS DETAILED SYSTEM INFORMATION

2022 2307 0000 0300 0005 0200

EXTENDED SYSTEM COMMAND CATALOGUE

The extended SYSTEM COMMAND catalogue is located in mass storage blocks 66-71. The format of each entry is exactly the same as the format for SYSTEM COMMANDS. In fact, a copy of the SYSTEM COMMAND catalogue is included in the extended SYSTEM COMMAND catalogue, starting at block 66. The last word of block 72 contains the first free mass storage block for extended SYSTEM COMMAND use. This entry must be updated by the user whenever an extended SYSTEM COMMAND is added to the system. Extended SYSTEM COMMANDS must never overlap the beginning of the user files. The user files may be moved to accomodate more system programs by changing the last three words of block 16 with DUMP (ZAP mode).

User programs may be made into extended SYSTEM COMMANDS with the following procedure:

1. Use DUMP (ZAP mode) to make the proper entries in the extended SYSTEM COMMAND catalogue.
2. Use GET to produce a virtual memory image of a binary program, in the scratch files.
3. Use DUMP (TRANSFER mode) to transfer the memory image from the scratch files to the space where the program will reside.



.

.



.

.



INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL

APPENDIX A--BOOTSTRAPPING PROCEDURE

The IFDOS bootstrap has been implemented in the INTERCEPT control panel PROMs. This makes bootstrapping a simple four step operation, as follows:

- 1.) Press "RESET" on the INTERCEPT front panel.
- 2.) With all SWITCH REGISTER switches in the "0" position (down), press "DEP FLAGS" on the INTERCEPT front panel.
- 3.) With SWITCH REGISTER switch 11 in the "1" position, and all other switches in the "0" position, press "BIN BOOT" on the INTERCEPT front panel.
- 4.) Press "CONT" on the INTERCEPT front panel as soon as the INTERCEPT halts at location 07600.

It is, of course, necessary that a diskette be properly inserted in drive "0" of the D10 before bootstrapping (see Appendix E, Step 8).

IFDOS responds with a "DOT" and rings the terminal bell when it has been successfully bootstrapped. It is now ready to accept user input.

IFDOS may be bootstrapped with the WRITE ENABLE/DISABLE switch in the DISABLE position. However, during normal use, this switch must be in the ENABLE position whenever output to the disk is made, or a disk error occurs. Recovery from this error or any other disk error requires that the error condition be corrected and that the 30 Hz switch on the INTERCEPT front panel is up.



4

2



4

4



INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
APPENDIX B--IFDOS ASCII TABLE

CHARACTER CODES

8-bit ASCII CODE	IFDOS ASCII CODE	CHARACTER REPRESENTATION	REMARKS
240	40		space (non-printing)
241	41	!	exclamation point
242	42	"	quotation marks
243	43	#	number sign
244	44	\$	dollar sign
245	45	%	percent
246	46	&	ampersand
247	47	'	apostrophe
250	50	(opening parenthesis
251	51)	closing parenthesis
252	52	*	asterisk
253	53	+	plus
254	54	,	comma
255	55	-	minus sign or hyphen
256	56	.	period, decimal point
257	57	/	slash
260	60	0	
261	61	1	
262	62	2	
263	63	3	
264	64	4	
265	65	5	
266	66	6	
267	67	7	
270	70	8	
271	71	9	
272	72	:	colon
273	73	;	semicolon
274	74	<	less than
275	75	=	equals
276	76	>	greater than
277	77	?	question mark
300	00	@	at sign (1)
301	01	A	
302	02	B	
303	03	C	

INTERCEPT FLOPPY DISK OPERATING SYSTEM USER'S MANUAL
APPENDIX B--IFDOS ASCII TABLE

304	04	D	
305	05	E	
306	06	F	
307	07	G	
310	10	H	
311	11	I	
312	12	J	
313	13	K	
314	14	L	
315	15	M	
316	16	N	
317	17	O	
320	20	P	
321	21	Q	
322	22	R	
323	23	S	
324	24	T	
325	25	U	
326	26	V	
327	27	W	
330	30	X	
331	31	Y	
332	32	Z	
333	33	[
334	34	\	opening bracket, SHIFT/K
335	35]	backslash, SHIFT/L
336	36	^	closing bracket, SHIFT/M
337	37	_	up arrow, circumflex
			back arrow, underline (2)

NOTES:

- (1) In IFDOS ASCII, 00 octal represents CARRIAGE RETURN.
- (2) In IFDOS ASCII, 37 octal represents TAB.

The INTERCEPT SOFTWARE REPORT is a cumulative report sent occasionally to all IFDOS users to inform them of:

- A. Software problems and solutions
- B. Software documentation errors and corrections
- C. General system information

INTERCEPT SOFTWARE REPORTS are numbered with the version of IFDOS which they refer to, and should be appended to the documentation package sent with each IFDOS System. Old reports should be discarded when new versions of IFDOS are received.

Any comments or questions related to INTERCEPT software, software documentation, or software reports should be directed to:

INTERSIL INC.
Department 2977 - INTERCEPT SOFTWARE
10900 N. Tantau
Cupertino, CA 95014



/IFDOS 3C, 3D, 3E CONTROL ROUTINE
 /FOR THE INTERCEPT FLOPPY DISK OPERATING SYSTEM
 /PROM LOCATIONS 7400-7777. THE PROM ADDRESS IS
 /COMPLIMENTED--SEE LOGIC DIAGRAM

/RAM LOCATIONS 0000-0017
 /PC DISPLAY IN 0020
 /DISPLAY IN 0040
 /FUNCTION SWITCHES IN 0060
 /EXAM-DX0 DEP PC-DX1 DEP FLAGS-DX2
 /DEP MEM-DX3 BINBOOT-DX4 USER-DX5
 /ROTARY SWITCH IN 0100
 /MD-DX8 AC-DX9 MQ-DX10 FLAGS-DX11

/IMPLEMENT USER FUNCTION. CURRENTLY
 /THE "USER FN " DECREMENTS PC BY 1 TO
 /RESTORE PC FOR A "EXAM AND MODIFY"
 /MEMORY FUNCTION.

/ THE TIME REQUIRED TO SERVICE A 30HZ
 /REQUEST IS 200 MICROSECONDS AT 4MHZ.
 /FOR CP ROUTINES TO FUNCTION PROPERLY
 /WHEN THE HLT SW IS DOWN, AN EVEN NUMBER
 /OF INSTRUCTIONS MUST BE EXECUTED IN
 /THE CONTROL PANEL PROGRAM. "NOP" IS
 /USED IN CERTAIN ROUTINES TO ENSURE THIS.

/ THE PROGRAM DEBOUNCES A FN SW CLOSURE AND
 /RELEASE BY 63.5 MS AT 4 MHZ

/BIN BOOT SWITCH SETTINGS:
 / 4000 = LOW SPEED PAPER TAPE READER
 / 0001 = INTERCEPT FLOPPY DISK OPERATING SYS.

0020	PCLEDS=0020
0040	DISLEDS=0040
0060	FN SW=0060
0100	ROTSW=0100

*0000	*0000
	/RAM LOCATIONS

00000	0000	PC,	0000	/PC SAVED HERE BY CP REQ GRANT
00001	0000	AC,	0000	/SAVED AC
00002	0000	BEGSW,	0	/BINARY LOADER RETURN SWITCH

			IFNZRO	.-3	<BAD ASSEMBLY>
00003	0000	EXEC,	0000	/SUBROUTINE ENTRY. ALSO TEMPORARY	
00004	0000		0000	/INIT TO CDF OR CIF	
00005	0000		0000	/INIT TO JMP I EXEC	

```

00006 0000 FNTR, 0000 /FLOPPY
00007 0000 SECT, 0000 /BOOT LOCATIONS
00010 0000 0000 /UNUSED
00011 0000 0000 /LOCATIONS
00012 0000 0000 /
00013 0000 CHAR, 0000 /BINBOOT SAVE LOCATIONS
00014 0000 WORD1, 0000
00015 0000 WORD2, 0000

IFNZRO --16 <BAD ASSEMBLY>
00016 0000 FLAGS, 0000 /L,0,IREQ,IIF,IENFF,0,<IF>,<DF>
00017 0000 CHKSUM, 0000 /BIN CHECK SUM ACCUMULATED
    
```

/FROM PROGRAM

```

*7777 *7777
07777 5776 JMP I 7776 /CPREQ ENTRY
*7776 *7776
07776 7532 START /CP ROUTINE ENTRY
*7400 *7400
    
```

```

07400 1000 HZ30, TAD PC /EVENTUAL 30HZ PATH
07401 3020 DCA PCLEDS /DISPLAY PC
07402 1100 TAD ROTSW /GET ROTARY SWITCH FUNCTION
07403 7012 RTR
07404 7500 SMA
07405 5264 JMP FLDIS
    
```

```

07406 7420 SNL
07407 5267 JMP MQDIS
    
```

```

07410 7012 RTR
07411 7700 K7700, SMA CLA
07412 5271 JMP ACDIS
    
```

```

K0070,
07413 0070 MDDIS, 70 /FOR EVEN INSTRUCTION SYNC
07414 7240 CLA CMA /DISPLAY MEMORY DATA
07415 1000 TAD PC
07416 3003 DCA EXEC
07417 1403 TAD I EXEC
    
```

```

07420 3040 EXIT, DCA DISLEDS /INSTR THAT WAS JUST EXECUTED
/OR DATA THAT WAS JUST DEPOSITED
    
```


07421	1000		TAD	PC	
07422	1225		TAD	K23	/CALLED FROM 7755 ?
07423	7640		SZA	CLA	/SKIP IF YES
07424	5247		JMP	NOERRS	/ELSE NO ERRORS
07425	0023	K23,		23	
07426	6755		SIN		/POSSIBILITY OF DISK ERRORS ?
07427	5247		JMP	NOERRS	/NO, DO NORMAL 30 HZ RETURN
		/	XDR		/YES, GET IMPLICIT STATUS
		/	AND	K3	/JUST PAR, CRC
		/K7640,	SZA	CLA	/ERRORS ?
		/	JMP	BAD	/YES, RETRY I/O OPERATION
07430	1342		TAD	K16	/READ ERROR REGISTER BITS
07431	6751		LCD		/LOAD COMMAND
07432	6755		SIN		/DONE YET ?
07433	5232		JMP	--1	/NO, WAIT
07434	6752		XDR		/READ ERROR REGISTER
07435	0301		AND	K377	/MASK OUT UNUSED BITS
07436	7650		SVA	CLA	/ERRORS ?
07437	5245		JMP	DONOP	/NO, DO NORMAL 30 HZ RETURN
		/K3,		3	/"NOP", ALSO A CONSTANT
		BAD,	INIT		/CLEAN UP DISK ERRORS
07440	6757		TAD	K7774	
07441	1331		TAD	I K7640	/BACK UP SYSIO ENTRY
07442	1720		DCA	PC	
07443	3000		JMP	NOERRS	
07444	5247		TAD	KNOP	/ISSUE NOP TO RAISE DONE
07445	1362	DONOP,	LCD		
07446	6751				
07447	1016	NOERRS,	TAD	FLAGS	/RESTORE DATA FIELD
07450	7004		RAL		
07451	7006		RTL		
07452	0213		AND	K0070	
07453	1351		TAD	K6201	
07454	3004		DCA	EXEC+1	
07455	4003		JMS	EXEC	/EXECUTE CDF TO RESTORE DF
07456	1016		TAD	FLAGS	
07457	7004		RAL		/RESTORE LINK
07460	7651	DEPPC,	XDEPPC		/GROUP 3 CLA
07461	1001		TAD	AC	/RESTORE AC
07462	6001		ION		/RESET CP INT FF
07463	5400		JMP	I 0	/RETURN TO MAIN MEMORY

07464	7300	FLDIS,	CLA CLL	/DISPLAY FLAGS
07465	1016		TAD FLAGS	
07466	5220		JMP EXIT	
07467	7701	MQDIS,	CLA MQA	/DISPLAY MQ
07470	5220		JMP EXIT	
07471	1001	ACDIS,	TAD AC	/DISPLAY AC
07472	5220		JMP EXIT	
07473	7300	EXAM,	CLA CLL	/EXAMINE ROUTINE
07474	1100		TAD ROTSW	
07475	7012		RTR	
07476	7012		RTR	
07477	7620		SNL CLA	/SKP IF NOT MD
07500	2000	ISZPC,	ISZ PC	/MD EXAM, INCREMENT PC
07501	0377	K377,	377	/NOP, ALSO A CONSTANT
07502	5316		JMP DEB	
07503	7604	DEPMEM,	CLA OSR	/DEPOSIT MEMORY ROUTINE
07504	3400		DCA I PC	
07505	5300		JMP ISZPC	

/RTF CANNOT BE USED TO RESTORE FLAGS
 /SINCE IT WILL CAUSE CPINTFF TO BE RESET

		XBIN00,		
07506	7604	DEPFLA,	CLA OSR	/DEPOSIT MEMORY ROUTINE
07507	3016		DCA FLAGS	
07510	1016		TAD FLAGS	
07511	0213		AND K0070	
07512	1327		TAD K6202	
07513	3004		DCA EXEC+1	
07514	4003		JMS EXEC	/EXECUTE CIF

/THE EMA LOGIC INHIBITS
 /IB TO IF TRANSFER IN CP MEM. THIS
 /MUST BE DONE BY EXECUTING LIF (6254)
 /WHICH LOADS IT DIRECTLY FROM IB

	6254		LIF=6254	
07515	6254		LIF	

07516	1060	DEB,	TAD FNSW	/DEBOUNCE FUNCTION SWITCH
07517	0211		AND K7700	
07520	7640	K7640,	SZA CLA	/STAY IN LOOP UNTIL SW
07521	5316		JMP --3	/IS RELEASED
07522	3003		DCA EXEC	
07523	2003		ISZ EXEC	

07644	6752	XDR		/GET A WORD
07645	3406	DCA I	FNTR	/STORE IT AWAY
07646	2006	ISZ	FNTR	/BUMP FNTR/CNTR
07647	5242	JMP	WAITTR	/DO MORE
07650	5767	JMP I	XDEB	/ALL DONE, START SYSTEM

7201	NL0001=CLA	IAC		
7326	NL0002=CLA	CLL	CML	RTL
7325	NL0003=CLA	CLL	CML	IAC RAL
7327	NL0006=CLA	CLL	CML	IAC RTL
7346	NL7775=CLA	CLL	CMA	RTL

6751	LCD=6751
6752	XDR=6752
6753	STR=6753
6754	SER=6754
6755	SIN=6755
6757	INIT=6757

07651	7604	XDEPPC,	CLA	OSR	/DEPOSIT PC ROUTINE
07652	3000		DCA	PC	
07653	5767		JMP I	XDEB	

/INTERSIL BINARY PAPER TAPE LOADER
 /PROGRAM COMPATIBLE WITH DEC "BIN" PAPER TAPE LOADER
 /IF THE USER TAPE IS PROPERLY LOADED CPU WILL STOP
 /WITH AC=0000. IF THE USER TAPE IS STARTED BEFORE
 /THE LEADER, THE PROGRAM WILL STOP AT THE LEADER
 /WITH AC=0000 OR 7600.

/NOTE!!! SWITCH REGISTER MUST EQUAL 4000 FOR
 /PAPER TAPE BINARY LOAD.

/DEC PDP-8/E COMPATIBLE TTY MNEMONICS
 /KCC -SET TTY RDR RUN
 /KSF -SKP IF TTY CHAR RDY
 /KRB -AC(4-11) GETS TTY CHAR
 /& SETS TTY RDR RUN

		BEGIN,		
07654	7700	X7700,	7700	/THIS SKIPS AND CLEARS AC
07655	6201	X6201,	6201	/ALWAYS SKIPPED
07656	6032		KCC	/INIT TTY READER
07657	6224		RIF	/GET IF
07660	1255		TAD X6201	
07661	3004		DCA EXEC+1	/FORM CDF INSTR. DEFAULT /IS THE CURRENT IF

/THE "BEGG" ROUTINE MAY BE ENTERED
 /FROM "BEGIN" OR "GO" LOOP. BEGSW=7777
 /IF FROM BEGIN AND =0000 IF FROM GO.

07662	7040		CMA
07663	3002		DCA BEGSW

/TTY RDR ENTRY

07664	6031	BEGG,	KSF
07665	5264		JMP .-1
07666	6036		KRB

/SAVE CHARACTER

07667	3013	SAV,	DCA CHAR
07670	1013		TAD CHAR

07671	0371		AND X0300	/CH 7&8
07672	1341		TAD M200	/AC<0 IF DATA OR ORIGIN
07673	7510		SPA	/SKIP IF L/T OR FIELD
07674	5307		JMP DAORG	
07675	7750	MASKX,	SPA SVA CLA	/SKP IF FIELD. IF L/T AC=0000
07676	5304		JMP LT	

/FIELD ENTRY POINT

07677	1013		TAD CHAR	
07700	0202		AND X0070	
07701	1255		TAD X6201	
07702	3003		DCA EXEC	/UPDATE EXEC SUBROINE
07703	5264		JMP BEGG	/FETCH NEXT CHARACTER

/LT EXIT

07704	2002	LT,	ISZ BEGSW	
07705	5352		JMP END	/BEG ENTERED FROM GO AND HENCE /TRAILER. IBIN EXIT
07706	5262		JMP BEGG-2	/BEGG ENTERED FROM BEGIN AND /HENCE LEADER. GO FETCH NEXT CHAR

/DATA OR ORIGIN EXIT

		DAORG,		
07707	7600	X7600,	7600	/CLA
07710	2002		ISZ BEGSW	
07711	5326		JMP GO+14	/ENTERED FROM GO ; RETURN

/CONTINUE BEGIN ENTRY

07712	3017	GO,	DCA CHKSUM	/CHECK SUM CLEARED IF INITIAL /ENTRY.
07713	1013		TAD CHAR	
07714	3014		DCA WORD1	/SAVE CHAR IN WORD1
07715	6031		KSF	/TTY FLAG UP ?
07716	5315		JMP --1	/NO, WAIT
07717	6036		KRB	/YES, GET A CHARACTER
07720	3013		DCA CHAR	/SAVE IT
07721	1013		TAD CHAR	/GET IT BACK
07722	3015		DCA WORD2	
07723	3002		DCA BEGSW	/SET UP BEGG SW FOR /ENTRY FROM GO
07724	4003		JMS EXEC	/EXECUTE CDF
07725	5264		JMP BEGG	/PSEUDO JMS TO BEGG /CHARACTER LOOK AHEAD RETURN /FROM BEGG IF NEXT CHAR IS NOT /TRAILER.
07726	1014		TAD WORD1	
07727	7106		CLL RTL	
07730	7006		RTL	
07731	7006		RTL	
07732	1015		TAD WORD2	
07733	7430		SZL	/L=0 IF DATAJ=1 FOR ORIGIN
07734	5343		JMP ORIGIN	
				/DATA ENTRY
07735	3400		DCA I 0000	/DEPOSIT DATA IN MEM
				/DISPLAY CODE
07736	1400		TAD I 0000	
07737	3040		DCA DISLEDS	/DISPLAY DATA THAT /WAS JUST DEPOSITED
07740	2000		ISZ 0000	/UPDATE POINTER
07741	7600	M200,	7600	/GROUP 2 CLA ALSO -200 CONSTANT
07742	7410		SKP	

```

                                /ORIGIN ENTRY
07743  3000  ORIGIN, DCA 0000    /UPDATE ORIGIN- NEW

                                /DISPLAY CODE
07744  1000          TAD 0000
07745  3020          DCA PCLEDS    /DISPLAY PC

                                /CHECKSUM CALCULATION
07746  1014  CHEX,  TAD WORD1
07747  1015          TAD WORD2
07750  1017          TAD CHKSUM
07751  5312          JMP GO    /UPDATE CHECK SUM AND CONTINUE
                                /ON GO LOOP

                                /RETURN FROM BEGG IF NEXT CHARACTER
                                /ON LOOK AHEAD IS TRAILER
07752  1014  END,   TAD WORD1
07753  7002          BSW
07754  0275          AND MASKX    /SOMETIMES PAL PUNCHES CH 7
                                /FOR PARITY . MASK OUT BIT 11
                                /AFTER BSW
07755  1015          TAD WORD2
07756  7041          CIA          /FETCH AND NEGATE CHECKSUM
                                /FROM TAPE
07757  1017          TAD CHKSUM   /AND ADD TO CALCULATED
                                /CHECKSUM. AC=0000 IF OK

                                /BINBOOT EXIT
07760  3001          DCA AC

                                /CHANGE IF TO DF
07761  6214          RDF
07762  1370          TAD X6202
07763  3004          DCA EXEC+1
07764  4003          JMS EXEC
07765  6254          LIF          /IB TO IF TRANSFER
07766  5767          JMP I .+1
07767  7516  XDEB,  DEB

07770  6202  X6202,  6202
07771  0300  X0300,  300

```

```

/SEPT 10 1976 THTH
/LAST EDIT: 1/14/77 DRK

```

S

AC	0001	FNTR	0006
ACDIS	7471	RET	7530
BAD	7440	ROTSW	0100
BEGG	7664	SAV	7667
BEGIN	7654	SDN	6755
BEGSW	0002	SECT	0007
BINBOO	7604	SER	6754
CHAR	0013	START	7532
CHEX	7746	STR	6753
CHKSUM	0017	USER	7576
DAORG	7707	WAITN	7615
DEB	7516	WAITTR	7642
DEPFLA	7506	WORD1	0014
DEPMEM	7503	WORD2	0015
DEPPC	7460	XBINBO	7506
DI SLED	0040	XDEB	7767
DONOP	7445	XDEPPC	7651
END	7752	XDR	6752
EXAM	7473	X0070	7602
EXEC	0003	X0300	7771
EXIT	7420	X6201	7655
FLAGS	0016	X6202	7770
FLDIS	7464	X7600	7707
FNSW	0060	X7700	7654
GO	7712		
HZ30	7400		
IFDOSX	7607		
INIT	6757		
ISZPC	7500		
KNOP	7562		
K0070	7413		
K16	7542		
K23	7425		
K377	7501		
K6201	7551		
K6202	7527		
K7640	7520		
K7700	7411		
K7774	7531		
LCD	6751		
LIF	6254		
LT	7704		
MASKX	7675		
MDDIS	7413		
MQDIS	7467		
M200	7741		
NL0001	7201		
NL0002	7326		
NL0003	7325		
NL0006	7327		
NL7775	7346		
NOERRS	7447		
ORIGIN	7743		
PC	0000		
PCLDS	0020		

NO ERRORS DETECTED

NO LINKS GENERATED

79 SYMEOLS

4K MEMORY UTILIZED



APPENDIX E

GETTING STARTED WITH IFDOS

VERSION 1B

The IFDOS package consists of the following:

HARDWARE:

INTERCEPT D10 Floppy Disk System
6908-FDI INTERCEPT to D10 interface board

SOFTWARE:

Two (2) IFDOS system diskettes
Three (3) blank, pre-formatted flexible diskettes

FIRMWARE:

Three (3) control panel PROMs, labeled IFDOS-3C, 3D and 3E.

DOCUMENTATION:

Intercept D10 Hardware Manual, IFDOS Handbook and D10
Engineering and Diagnostic Software Documentation.

Before any attempt is made to start IFDOS, the above documentation should be read carefully.

To prepare the D10 and start IFDOS, the following steps must be followed:

1. Unpack D10 floppy disk system and check for shipping damage. Notify INTERSIL immediately if any damage is found.
2. Remove CPUTTY board from INTERCEPT prototyping system. Break the trace which runs to pin 8 on the CPUTTY edge connector. Place a jumper between pin 6 on the CPUTTY edge connector and the trace which formerly ran to pin 8 (XTB).
3. Insert 6908-FDI PC card into any available slot of the INTERCEPT bus. Be certain to insert the 6908 correctly (i.e., component side up on left hand side of INTERCEPT; component side down on right hand side).
4. Remove 6903-CONTRL PC card from INTERCEPT. Remove the three control panel PROMs from their sockets and replace them with the new PROMs supplied. Observe the markings "3C", "3D" and "3E" on the PC card and on the PROMs. Replace the 6903-CONTRL card.

5. Connect ribbon cable between 6908-FDI and D10.
6. Remove foam-rubber inserts from both D10 floppy disk drives. Remove foreign material from the drives, if any.
7. Power up the INTERCEPT and the D10. (Caution: The D10 should never be powered up or powered down with a diskette inserted in either drive when the doors are closed).
8. Insert one IFDOS system diskette in drive 0 (the lefthand drive). Close the door and place the "WRITE ENABLE/DISABLE" switch in the "DISABLE" position. The diskette label must be facing left, and the head access slot in the diskette jacket must be horizontal.
9. Press "RESET" on the INTERCEPT front panel.
10. With all switch register switches in the 0 (down) position, press "DEP FLAGS" on the INTERCEPT front panel.
11. With switch register bit 11 in the "1" position (up), and all other switches in the "0" position, press "BIN BOOT" on the INTERCEPT front panel.

NOTE: Switch register bit 0 must be in the "1" position and all other switches must be in the "0" position for loading binary paper tapes.

12. When the INTERCEPT halts at location 07600, press "CONT" on the INTERCEPT front panel. IFDOS will respond by ringing the bell and typing a dot on the terminal. IFDOS is now ready to accept keyboard commands.

NOTE: If IFDOS does not respond by ringing the terminal bell and typing a dot, check the following, verify that steps 1-8 have been done correctly, and proceed to step 9. Otherwise, proceed to step 13.

- A. Ensure that INTERCEPT and D10 are supplied with power.
- B. Ensure that ASCII terminal is "ON LINE".

13. Insert the empty diskette in drive 1 (the righthand drive) and close the door. Place the "WRITE ENABLE/DISABLE" switch (for drive 1 only) in the "ENABLE" position.
14. Copy the IFDOS system diskette in drive 0 to the empty diskette in drive 1 as follows. (Underlined characters are typed by IFDOS).

```

._ R SYSCP Y
INPUT 0:0
OUTPUT 1:0
LAST INPUT BLOCK 377
VERIFY? YES!

```

15. When copying is completed, type "CONTROL/C". When IFDOS types a dot, remove the IFDOS system diskette from drive 0, return it to the envelope provided, and store it (along with the backup system diskette provided) in a safe place.
16. Remove the diskette from drive 1, label it, place in drive 0 and bootstrap the system by repeating steps 9, 10 and 11.
17. Normal operation of IFDOS requires the "WRITE ENABLE/DISABLE" switch to be in the "ENABLE" position.





INTERSIL

10710 N. Tantau Avenue
Cupertino, CA 95014
Tel: (408) 996-5000
TWX: 910-338-0171