CARNEGIE-MELLON UNIVERSITY

HYBRID COMPUTATION LABORATORY

USER MANUAL

Series 69-1

January, 1969

Contents

The following three sheets are an update to the INTRODUCTION (white) section of the hybrid lab user's manual.

TABLE OF CONTENTS

board outside the laboratory. The computers are available for student and faculty use from 9:30 a.m. to 5 p.m., Monday through Saturday; and from 7 p.m. to midnight, Monday through Friday. Other hours may be scheduled by experienced users by appointment. Funded research projects will be given first choice of the scheduled hours. In all cases, scheduled time is subject to pre-emption for necessary hardware and/or software maintenance.

All necessary supplies with the exception of personal copies of the documentation, personal DECtapes, and private analog patch boards are provided cost free by the hybrid lab. Documentation and DECtapes may be purchased from the CMU bookstore. Private analog patch boards will be ordered (at cost) from EAI. Funded research projects are encouraged to purchase their own DECtape and analog patch boards if the projected computer usage warrants. Plans for long and short term leasing of analog patching equipment are under consideration.

1.4 DOCUMENTATION

This user manual constitutes the primary documentation facility for the CMU hybrid lab. It is organized in a loose leaf structure to facilitate corrections and additions. Corrections and additions, as they become available, will be announced in the data flag and may be picked up in the hybrid lab (at no cost).

In addition, the following equipment manufacturer publications will be very useful for hybrid lab users and are strongly recommended:

EAI 680 Reference Handbook, EAI publication #00800.2048-1, July, 1967.

Handbook of Analog Computation, EAI publication #00800.0001-3, July, 1967.

Fortran IV Manual, DEC-9A-AF4B-D, Digital Equipment Corporation, 1968.

Utility Program Manual, DEC-9A-GUAB-D, Digital Equipment Corporation, 1968.

The following are recommended as supplementary manuals:

Basics of Parallel Hybrid Computers, EAI publication #00800.3039-0, June, 1968.

Keyboard Monitor Guide, DEC-9A-NGBA-D, Digital Equipment Corporation, 1968.

PDP-9 User Handbook, F-95, Digital Equipment Corporation, January, 1968.

The following manuals are necessary for assembly language users:

MACRO-9 Manual, DEC-9A-AM9A-D, Digital Equipment Corporation, 1967.

Keyboard Monitor Manual, DEC-9A-MABO-D, Digital Equipment Corporation, 1968.

All of the above-listed manuals and the hybrid lab user's manual are for sale in the CMU bookstore and available for reference in the hybrid lab.

The following three sheets are an update to the SYSTEM DESCRIPTION (white) section of the hybrid lab user's manual.

TABLE OF CONTENTS

2.0 HARDWARE

2.1 PDP-9 (DIGITAL)

The CMU hybrid computation laboratory PDP-9 digital computer configuration currently consists of the following equipment:

a.) PDP-9 digital computer with 1 μsec memory cycle time, real-time (60 cps) clock, 16K of 18-bit word core memory, and an extended arithmetic element.

b.) API option: 8 levels of automatic priority interrupts (4 hardware and 4 software levels).

c.) KSR35 teletype (console typewriter): 10 characters/second.

d.) High-speed paper tape I/O: 300 characters/second input and 50 characters/second output.

e.) Bulk storage I/O: 1 DECtape controller with 4 DECtape units--fixed position reading and writing, 576 blocks of 256 18-bit words each per tape, 200 μsec/18-bit word transfer rate, 53 msec/block transfer rate (2.6 million bits/tape).

f.) Dataphone: full duplex, variable speed.

Detailed information concerning PDP-9 hardware is contained in DEC publication F-95, PDP-9 User's Handbook, January, 1968.

2.2 EAI 680 (ANALOG)

The EAI 680 is a 10 volt analog computer capable of computation in real time and compressed time up to 10,000 times real time. The 680 also has digital components for logic operations and control inputs to the analog components, and peripheral equipment for a wide range of output display.

3.2 HYBRID COMPUTATION LABORATORY

The software described below was developed and is maintained by the hybrid lab. Complete documentation of this software is contained in this user's manual.

a.) Hybrid Static Analog Test Program (HYSAT): provides the user with a standard facility for initial problem setup, static analog checkout, and potentiometer value save.

b.) linkage routines: a package of FORTRAN-callable functions and subroutines which provides the user access to all of the capabilities of the EAI 693 hybrid interface, as well as PDP-9 priority interrupt manipulations, console facilities, and the real-time clock.

c.) Continuous Systems Modeling Program (CSMP): provides a digital assist in the accurate simulation of dynamic systems modeled by ordinary differential and difference equations.

d.) function generation package: a collection of routines for hybrid function generation and digitally-computed optimization of analog function generators.

e.) DCT-2000 I/O handlers: input/output device handlers which enable the use of a DCT-2000 (UNIVAC Data Communications Terminal) as a standard PDP-9 peripheral device (via the dataphone or direct cable).

The following five sheets are an update to the PDP-9 OPERATION (yellow DIGITAL) section of the hybrid lab user's manual. The DCT-2000 may now be used as a standard peripheral device of the PDP-9 via the PDP-9's dataphone interface.

TABLE OF CONTENTS

Do not attempt to handle or clean the read/write heads. They are kept sufficiently clean by hybrid lab personnel during periodic preventive maintenance checks.

3.2 DCT-2000

The UNIVAC DCT-2000 (Data Communication Terminal) may be utilized by the PDP-9 as a standard I/O peripheral device. This section describes operating procedures for the DCT-2000 in the PDP-9 Keyboard Monitor environment. The DCT-2000 can read or punch up to 120 cards/minute and print up to 250 lines/minute when communicating with the PDP-9. Only one DCT-2000 card facility can be used (card punch or card reader) at any given time. Also, data transmission may only proceed in one direction at a time.

3.2.1 PREPARATION

Before any PDP-9/DCT-2000 transmission is initiated, the user should perform the following steps at the DCT-2000 console.

A. Push POWER ON indicator-button.
B. If the ON LINE/LIST indicator-button indicates LIST, push it once so that it indicates ON LINE.
C. Maintenance panel (behind the UNIVAC DCT-2000 nameplate):
    1. SHORT BLOCK ON.
    2. PARITY CHECK ON.
    3. TEST MODE to OFF.
    4. Others are irrelevant.
D. Console:
    1. SYSTEM CONTROL RUN/STOP to RUN.
    2. Others will be set as required.

3.2.2 LINE PRINTER

Before use of the line printer is initiated, perform the following steps at the DCT-2000 console:

1. Depress GENERAL CLEAR once.
2. Set RECEIVE CONTROL BLOCK LENGTH to 128.
3. Set RECEIVE CONTROL SELECT RECEIVE UNIT to PRINTER.

The PDP-9 will then automatically initiate, control, and terminate

transmissions to the DCT-2000 line printer without further user intervention.

3.2.3 CARD PUNCH

Before use of the card punch is initiated, perform the following steps at the DCT-2000 console:

1. Depress GENERAL CLEAR once.
2. Set RECEIVE CONTROL BLOCK LENGTH to 80.
3. Set RECEIVE CONTROL SELECT RECEIVE UNIT to PUNCH.
4. Set READ/PUNCH to READ.
5. Clear out the input hoppers.
6. Depress CARD FEED once (to clear out the internal card stations).
7. Clear out the output hopper.
8. Load blank cards in the input hopper (9 edge face down).
9. Set PUNCH CHECK to ON (verifies that cards are punched correctly).
10. Set READ/PUNCH to PUNCH.
11. Depress CARD FEED once (to advance a blank card to the punch station).
12. Turn RECEIVE CONTROL PRINT MONITOR ON if it is desired to print as well as punch.

The PDP-9 will then automatically initiate, control, and terminate transmissions to the DCT-2000 card punch without further user intervention.

3.2.4 CARD READER

Before use of the card reader is initiated, perform the following steps at the DCT-2000 console:

1. Depress GENERAL CLEAR once.
2. Set TRANSMIT CONTROL BLOCK LENGTH to 80.
3. Set TRANSMIT CONTROL SELECT TRANSMIT UNIT to READER.
4. Set READ/PUNCH to READ.
5. Clear out the input hopper.
6. Depress CARD FEED once (to clear out the internal stations).
7. Clear out the output hopper.
8. Load card deck in the input hopper (9 edge face down).
9. Depress CARD FEED once (to advance a card to the read station).
10. Turn TRANSMIT CONTROL PRINT MONITOR ON if it is desired to print the cards as they are transmitted.

After the card reader handler is initialized by a system or user program, the PDP-9 will be waiting for user input. Depress the TRANSMIT

CONTROL TRANSMIT switch once (TRANSMIT CONTROL TRANSMIT indicator will light). The PDP-9 will then begin reading cards from the DCT-2000.

Should the PDP-9 require card records more slowly than the DCT-2000 reads them, the DCT-2000 alarm will eventually sound (3 NAK status indicator lights). If this happens, depress ALARM CLEAR (on the DCT-2000 console) once to clear the alarm, or set the ALARM CLEAR switch to OFF (up position) to disable the alarm completely.

Sometimes it is necessary to physically indicate to the PDP-9 (i.e., when transmitting cards to PIP) that there are no more cards to be read. This is accomplished at the DCT-2000 by depressing and releasing the TRANSMIT CONTROL SEND EOT (end-of transmission) switch once (and only once) after the last card has been transmitted and the TRANSMIT CONTROL TRANSMIT indicator goes out.

3.2.5 INTERRUPTING A COMMUNICATION

If it is necessary to interrupt a PDP-9/DCT-2000 communication (e.g., in order to add more blanks cards to the input hopper during a PUNCH operation), perform the following steps at the DCT-2000 console:

1. Set the RUN/STOP switch to STOP.
2. When the DCT-2000 comes to a halt, perform the desired function (but do not depress GENERAL CLEAR).
3. Set RUN/STOP to RUN.
4. If the DCT-2000 was transmitting to the PDP-9 from the card reader, depress the TRANSMIT CONTROL TRANSMIT switch once (TRANSMIT CONTROL TRANSMIT indicator will light).

This procedure insures that the communication will resume after the interruption without any data loss in the meantime.

3.2.6 TERMINATING A COMMUNICATION

When communicating with user programs and (less frequently) system programs, the DCT-2000 will not clear completely when the communication is terminated, especially if the communication was aborted.

The user may depress GENERAL CLEAR in this instance to insure that the communication is properly terminated and that the DCT-2000 will be ready for the next-initiated communication.

3.3 OTHER

If any other I/O equipment needs servicing or supplies (malfunction, teletype paper or ribbon, paper tape for the punch), see the user consultant on duty for assistance.

The FEED switch to the right of the punched paper tape receptacle will feed blank tape (sprocket holes only) for leader or trailer purposes for as long as it is depressed. The POWER switch controls power to the paper-tape punch mechanism. It should be left permanently on (indicator lit) except when the PDP-9 is turned off.

The ON/OFF switch on the teletype keyboard controls power to the teletype. It should be left permanently in the ON position. The BREAK indicator beneath the ON/OFF switch is inoperative. The red BRK-RLS button is also inoperative. The red LOC CR button will return the teletype carriage when depressed. The red LOC LF button will cause rapid paper unspacing as long as it is depressed. The red REPT button will cause the last character typed to be repeated as long as the REPT button is held down.

This write-up replaces the KEYBOARD MONITOR (yellow DIGITAL) section of the hybrid lab user manual.

USER MANUAL

KEYBOARD MONITOR

An Operating Guide
for the
PDP-9 Monitor System

Carnegie-Mellon University
Hybrid Computation Laboratory
January, 1969

TABLE OF CONTENTS

INDEX OF TABLES

INDEX OF FIGURES

1.0 INTRODUCTION

The Keyboard Monitor (KM9) system for the PDP-9 is an executive program which provides for automatic storing, calling, loading, and executing of system and user programs; executes all I/O operations; provides error detection, error messages, and error recovery; and provides for program pause and saving of the status of all relevant registers. This write-up describes the input-output conventions and operation of the Keyboard Monitor.

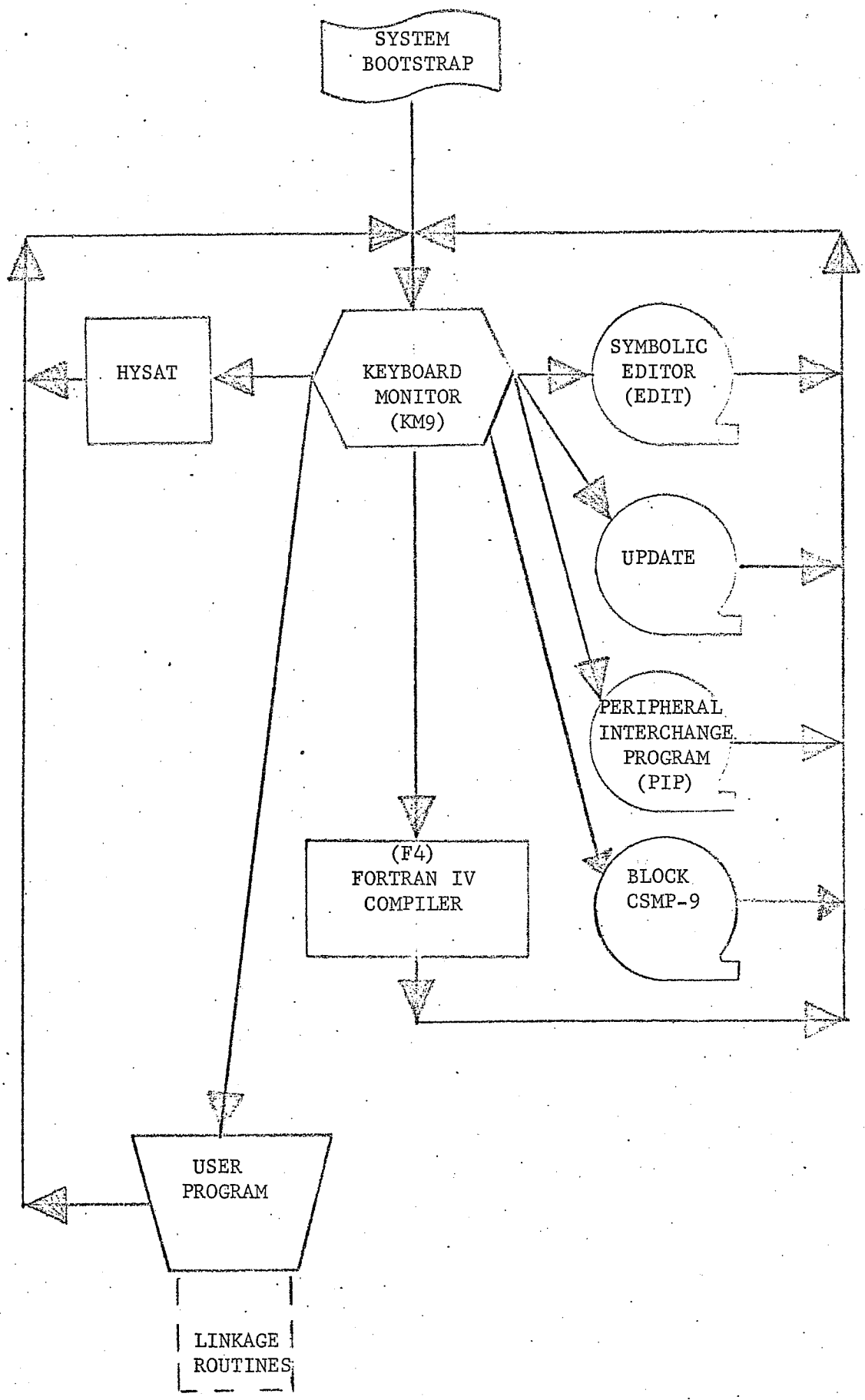
1.1 PROGRAM FLOW

Figure 1 illustrates the normal sequence of operations of the PDP-9. If the Keyboard Monitor is not already loaded into core memory, the PDP-9 must be initialized with the System Bootstrap (see section 3.0). The Keyboard Monitor accepts commands to load any system or user program (see sections 4.6 and 5.0). The control of all system programs is of a conversational, on-line nature. When execution of any system or user program is terminated, control is automatically returned to the Keyboard Monitor.

1.2 SYSTEM TAPE

The Keyboard Monitor System Tape is a file-structured DECtape permanently mounted on DECtape unit 0(8), WRITE LOCK. This system tape contains all system programs (including the Keyboard Monitor itself) as well as a system library file which contains all necessary system subroutines. The system library file is so ordered that only one pass through the system library loads all necessary subroutines, the subroutines that they might require, and all necessary I/O device handlers.

FIGURE 1. PDP-9 Program Flow



2.0 I/O CONVENTIONS

Input-output operations in the PDP-9 Keyboard Monitor system are file-oriented and device-independent. This section describes the I/O conventions of the KM-9 I/O system.

2.1 DEVICE ASSIGNMENT

Device-independent I/O operations in system or user programs are related to a specific I/O device by referencing a slot in the device assignment table (.DAT). Each slot in the .DAT contains the name of the specific I/O device handler to be associated with this .DAT slot. The standard .DAT associations are listed in Tables 1 and 2. Note that all .DAT system references are to negative slots and all .DAT user references are to positive slots. Note also that .DAT slot numbers are in octal radix. Altering .DAT device associations is covered in section 4.5.

2.2 DECTAPE

DECtapes may be either file-structured or non file-structured; all other peripheral devices are by definition non file-structured. Non file-structured DECtapes are analogous in use to standard magnetic tapes; they may be manipulated in user Fortran IV programs with the REWIND, END FILE, and BACKSPACE commands.*

The term "file-structured" means simply that a directory exists on the DECtape to identify by name and location the files which are recorded on it. A directory listing of any DECtape so recorded is available via the (L)ist command in PIP on the D(IRECT) command in KM-9. A directory may be cleared via the S or N switch in PIP or the N(EWDIR) command in KM-9 (use of the PIP S switch is recommended in order to assure regeneration of

*DTD is the only DECtape I/O handler which will process magnetic tape type commands.

TABLE 1. System .DAT Associations.

.DAT Slot	I/O Device	Unit	Handler	Use
-15	DECtape	3	DTA	Output (EDIT, UPDATE)
-14	DECtape	2	DTA	Input (EDIT, UPDATE)
-13	DECtape	3	DTA	Output (FORTRAN IV)
-12	DCT-2000 lineprinter		LPZ	Listing (FORTRAN IV, UPDATE).
-11	DECtape	2	DTA	Input (FORTRAN IV)
-10	paper-tape reader		PRA	Secondary Input (EDIT, UPDATE)
- 7	DECtape	0	DTC	*System Device (system loader)
- 6	none			Reserved (dataphone)
- 5	DECtape	0	DTC	User Library Tape
- 4	DECtape	3	DTC	Input (Linking Loader)
- 3	teleprinter		TTA	*Output } All system programs
- 2	keyboard		TTA	*Input } All system programs
- 1	DECtape	0	DTC	System Library

* These device assignments cannot be altered.

TABLE 2. User .DAT Associations.

.DAT Slot	I/O Device	Unit	Handler	Use
1	DECtape	1	DTA	Input and Output
2	DECtape	2	DTA	Input and Output
3	DECtape	3	DTA	Input and Output
4	teletype		TTA	Input and Output
5	paper-tape reader		PRA	Input
6	paper-tape punch		PPA	Output
7	DCT-2000 card reader*		CDZ	Input
10	DCT-2000 line printer*		LPZ	Output

*N.B.: .DAT slots 7 (CDZ) and 10 (LPZ) cannot both be in active use at the same time.

system blocks on the tape.) File-structured DECtapes may be manipulated in user Fortran IV programs by use of the SEEK, ENTER, and CLOSE subroutines.

Standard directory information is recorded on DECtape blocks 71-100 and provides for up to 56 files per DECtape. A file is identified by a 6-character file name* and an optional 3 character file-name extension,* or 9 characters in all. Short file-names and short (or non-existent) file-name extensions are left-justified and zero-filled. System programs use predetermined file-name extensions in their operation. For example, if the FORTRAN IV compiler wishes to seek program ABCDEF as source input, it searches for ABCDEF SRC (ABCDEF, Source). The binary output produced would be named ABCDEF BIN (ABCDEF, Relocatable Binary), while the listing produced would be name ABCDEF LST (ABCDEF, Listing). The Linking Loader, if told to load ABCDEF, would seek ABCDEF BIN.

2.3 TELETYPE

Whenever the teletype keyboard is used for input to the monitor, system programs or user programs, the following error correction characters may be used:

RUBOUT(RO) - delete the previous character typed and echo a reverse slash (\).

↑U (CTRL U) - delete the entire line typed so far and echo a commercial at sign (@).

*any printing characters in the ASCII set may be used with the exception of :;,(and).

3.0 SYSTEM BOOTSTRAP

The only function of the System Bootstrap is to initial load and then transfer control to the Keyboard Monitor.

To load the System Bootstrap:

A. Position the System Bootstrap paper tape* in the high-speed reader.

B. Mount the System Tape on DECtape unit 0, WRITE LOCK.

C. Position REPT SPEED control at position 3, 4, or 5.

D. Set the ADDRESS switches to 37637.

E. Depress the I/O RESET switch.

F. Depress the READIN switch.

The Keyboard Monitor will then be automatically loaded into core memory.

If the System Bootstrap has not been inadvertently destroyed, it may be restarted as follows:

A. Set the ADDRESS switches to 37646.

B. Depress the I/O RESET switch.

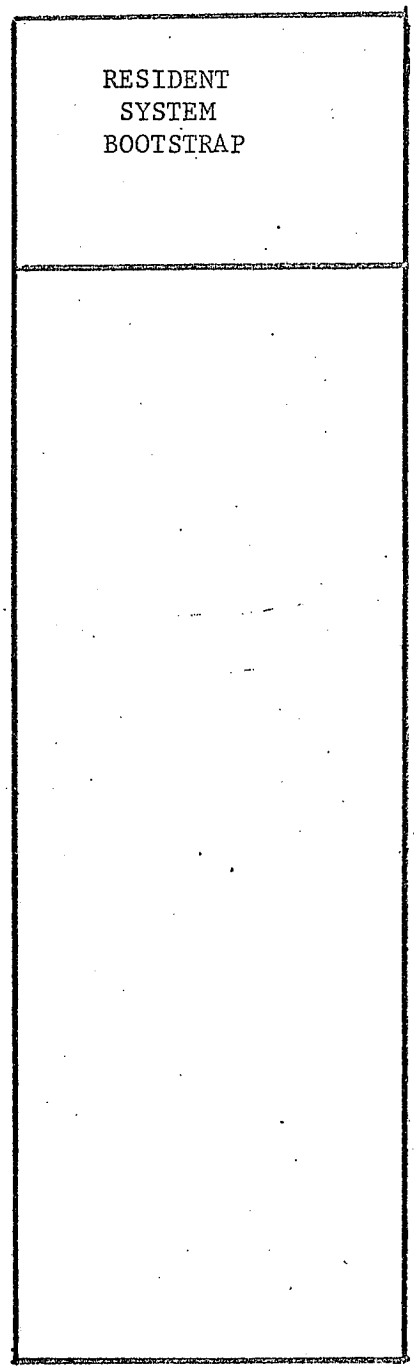
C. Depress START.

See figures 2 and 3 for memory maps of the System Bootstrap and Keyboard Monitor.

---

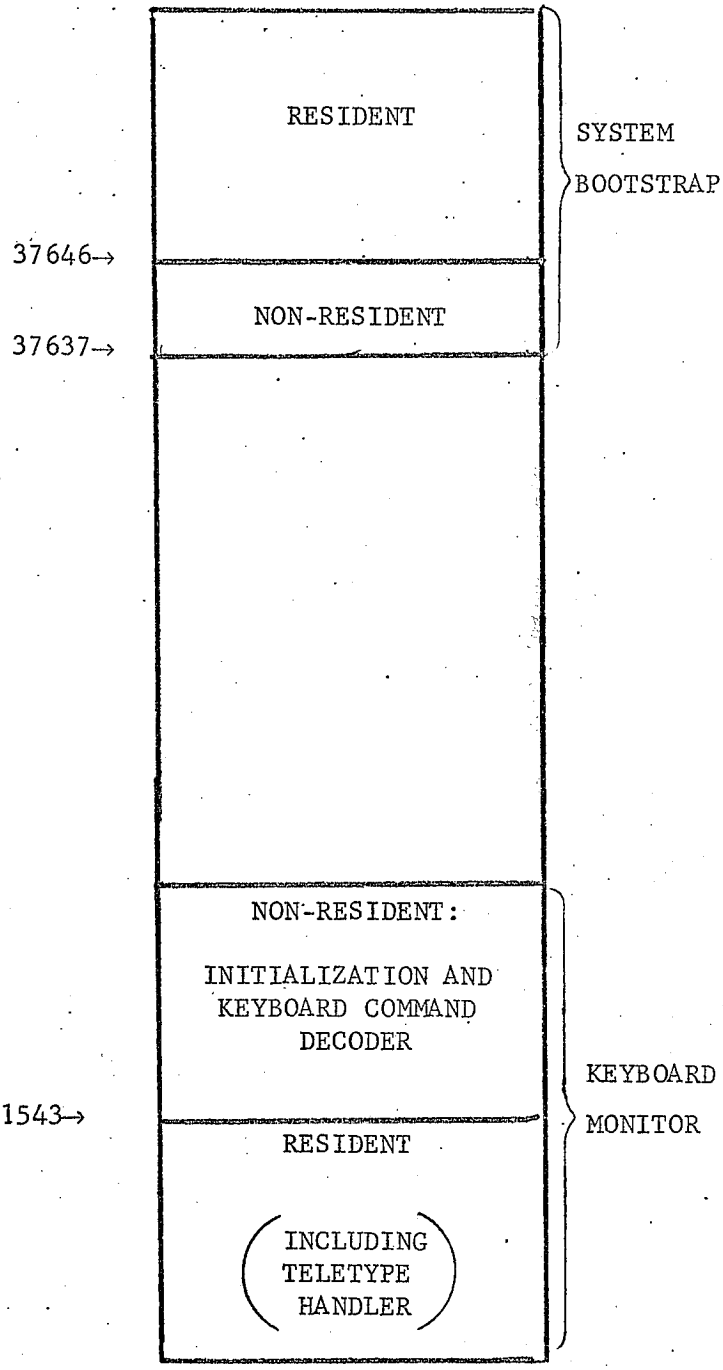
*located just above the paper tape reader.

FIGURE 2 Memory Map: System Bootstrap



(Addresses are in octal radix.)

FIGURE 3 Memory Map: Keyboard Monitor



(Addresses are an octal radix.)

4.0 KEYBOARD MONITOR COMMANDS

When the Keyboard Monitor initially gets control it outputs

```
MONITOR V4B
$
```

to the teleprinter to indicate readiness to accept a keyboard command. Subsequently, it outputs "$" to indicate readiness. In both cases, keyboard commands should be typed on the same line as the dollar sign ($). The first letter of the command is sufficient for the first six commands described in this section. V4B is the Keyboard Monitor version identification number.

4.1 LOG

The LOG command is used to make hard copy records of the user's comments on the teletype. Upon encountering the LOG command, the Keyboard Monitor ignores all typing up to and including the next ALT MODE.

EXAMPLE 1: $LOG THIS IS A SAMPLE LOG COMMAND.(ALT MODE)

4.2 DIRECT

The DIRECT command causes a printout of the directory of any file-structured DECtape. The command takes the following form:

```
DIRECT n
```

where n is the unit number Ø-7, with Ø as the default assumption.

```
EXAMPLE 2:  $DIRECT
            DIRECTORY LISTING
            .LOAD  BIN  31
            .LIBR  BIN  52
             EDIT  SYS 217
             72 FREE BLOCKS
```

The third column in the printout is the number of the first DECtape block occupied by the file; all numbers in the printout are in octal radix.

4.3 NEWDIR

The command NEWDIR n will clear the directory blocks on DECtape unit n, n=Ø being illegal. (Caution is urged in the use of this command; make absolutely certain that n is correct before typing carriage-return.) Note that this command does not reserve a ↑Q area on DECtape n.

4.4 REQUEST

The REQUEST command allows examination of selected .DAT dot associations. The command takes the form:

REQUEST XXXXXX

where XXXXXX is the system program name (EDIT, F4, UPDATE, PIP, CHAIN, or LOAD), USER for all positive .DAT slots, or blank for the entire .DAT table.

EXAMPLE 3: $REQUEST F4

.DAT	DEVICE	USE
-13	DTA3	OUTPUT
-12	LPZ	LISTING
-11	DTA2	INPUT
-3	TTA	CONTROL AND ERROR MESSAGES
-2	TTA	COMMAND STRING

4.5 ASSIGN

The ASSIGN command may be used to alter .DAT I/O device handler associations whenever it is necessary to use I/O device assignments other than the standard assignments (see Tables 1 and 2). The change of assignment is only effective for the current job, since the standard assignments are restored whenever control is returned to the Keyboard Monitor. The

command takes the following form:

ASSIGN DEVm a,b,.../DEVn x,y,...

where DEV is the device handler name (see Table 3); m,n,... are unit numbers if DEV refers to DECtape; and a,b,...x,y,... are .DAT slot numbers. DEVm can be replaced by NONE (or NON) to clear .DAT slots. Handler A is assumed if a two-letter handler name is typed, and unit 0 is assumed if no unit number (m,n,...) is typed. .DAT slots -2, -3, and -7 cannot be modified. .DAT slot -1 should not be modified. One and only one I/O handler for a device should be in use at any given time since there is no communication between redundant handlers.

CDZ (DCT-2000 card reader) and LPZ (DCT-2000 line printer) cannot both be active (performing I/O operations) at the same time. The ASSIGN command must be executed immediately before the system or user program which is to reference the newly assigned .DAT slots is loaded.

EXAMPLE 4: $ASSIGN DTA1 -11/DTA2 -12/DT -13

result: DECtape I/O handler A is assigned to DECtape unit 1 on .DAT slot -11, DECtape unit 2 on .DAT slot -12, and DECtape unit 0 on .DAT slot -13.

EXAMPLE 5: $ASSIGN NONE 4, 5, -12

result: .DAT slots 4, 5, and -12 are cleared.

EXAMPLE 6: $ASSIGN DTC1 -11/PPA -12, -13

result: DECtape I/O handler C is assigned to DECtape unit 1 on .DAT slot -11, paper-tape punch I/O handler A is assigned to .DAT slots -12 and -13.

EXAMPLE 7: $ASSIGN CDZ -11/LPZ -12

result: The DCT-2000 card reader handler is assigned to .DAT slot -11 and the line printer handler to .DAT slot -12. This command would result in havoc (if the FORTRAN IV compiler is subsequently used) because the card reader (source input device) and the line printer (listing device) would then be active simultaneously.

TABLE 3. Available I/O Device Handlers

I/O Device	Handler Name	Handler Size*	Use
teletype	TTA	(in Keyboard Monitor)	input & output
paper-tape reader	PRA	444	input, all data modes
	PRB	294	input, symbolic ASCII tapes only
paper-tape punch	PPA	377	output, all data modes
	PPB	270	output, all data modes except ASCII
	PPC	210	output, binary tapes only
DECtape	DTA	2321	3 files, input & output, all data modes
	DTB	1552	2 files, input & output, symbolic ASCII or binary files only
	DTC	680	1 file, input only, symbolic ASCII or binary files only
	DTD	1569	1 file, input & output, all data modes, magnetic tape functions
dataphone/DCT-2000	CDZ	535	card-reader input, symbolic ASCII mode only
	LPZ	427	line printer (or card punch) output, symbolic ASCII mode only

*in decimal radix

EXAMPLE 8: $ASSIGN DTC1 -11/DTA2 -12/DTB3 -13

result: DECtape I/O handler C is assigned to DECtape unit 1 on .DAT slot -11, DECtape I/O handler A is assigned to DECtape unit 2 on .DAT slot -12, DECtape I/O handler B is assigned to DECtape unit 3 on .DAT slot -13. This command would result in havoc (if the FORTRAN IV compiler is subsequently used) because three different DECtape I/O handlers would then be in use simultaneously.

4.6 EXECUTE

EXECUTE is a (DEC) system program which loads an XCT type file (XCT files are constructed by the CHAIN system program), allocates blank COMMON, and controls transfer between chains of an XCT file. The command to EXECUTE consists of the file name of the XCT type file to be run. The XCT file is searched for on the I/O device associated with .DAT slot -4.

EXAMPLE 9: MONITOR V4B
$EXECUTE MYPROG

result: The user program MYPROG XCT on DECtape unit 3 (normal assignment of .DAT slot -4) is loaded and run.

4.6.1 ERROR MESSAGES

If EXECUTE detects a fatal error condition during execution of the specified XCT-type program file, an appropriate error message (see Table 4) will be output to the teleprinter and control is returned to the Keyboard Monitor. If the blank COMMON size in a new chain differs from that of the previous chain during execution, EXECUTE will output the following warning message on the teleprinter:

WARNING - COMMON SIZE DIFFERS

TABLE 4. EXECUTE Error Messages.

Message	Meaning
SEG Ø1	A chain attempted to call itself.
SEG Ø2	End-of-file reached without finding the requested chain.
SEG Ø3	End-of-medium reached without finding the requested chain.
SEG Ø4	Read error on device associated with .DAT slot -4.
SEG Ø5	Blank COMMON overlaps the requested chain.

4.6.2 LOADING (HYBRID LAB) SYSTEM PROGRAMS

Hybrid lab maintained system programs are stored on the system tape as single-chain XCT-type files. They may be loaded via the following commands:

```
MONITOR V4B
$ASSIGN DTCØ -4
$EXECUTE FILENM
```

where FILENM is any of the following:

CSMP - Continuous Systems Modeling Program
DYNAM - Dynamic Optimal Breakpoint Selection Program
HYSAT - Hybrid Static Analog Test Program

4.7 LOADING (DEC) SYSTEM PROGRAMS

The commands available to the user for loading DEC system programs via the Keyboard Monitor are:

$LOAD	Linking Loader
$GLOAD	Linking Loader and Go
$EDIT	Symbolic Editor
$PIP	Peripheral Interchange Program
$F4	FORTRAN IV Compiler
$UPDATE	Library File Utility Program
$CHAIN	FORTRAN IV chain builder

All of these commands should be terminated by a carriage-return or ALT MODE. When the requested system program has been loaded and is waiting for keyboard input, an indication is given on the teleprinter with an appropriate message, such as

```
PIP V7A
>
```

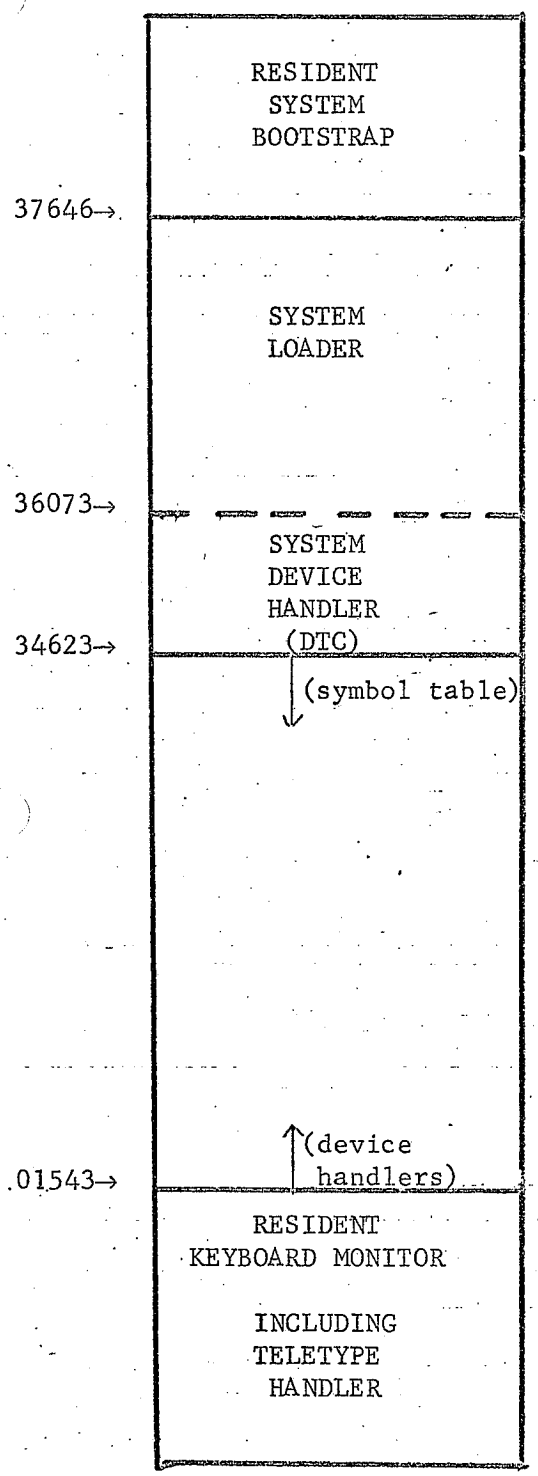
or

```
LOADER V3A
>
```

etc. V3A, V7A, etc. are system program version identification numbers.

See Figure 4 and 5 for memory maps of system program loading and operation. See Table 5 for a list of system program loader (.SYSLD n) error messages.

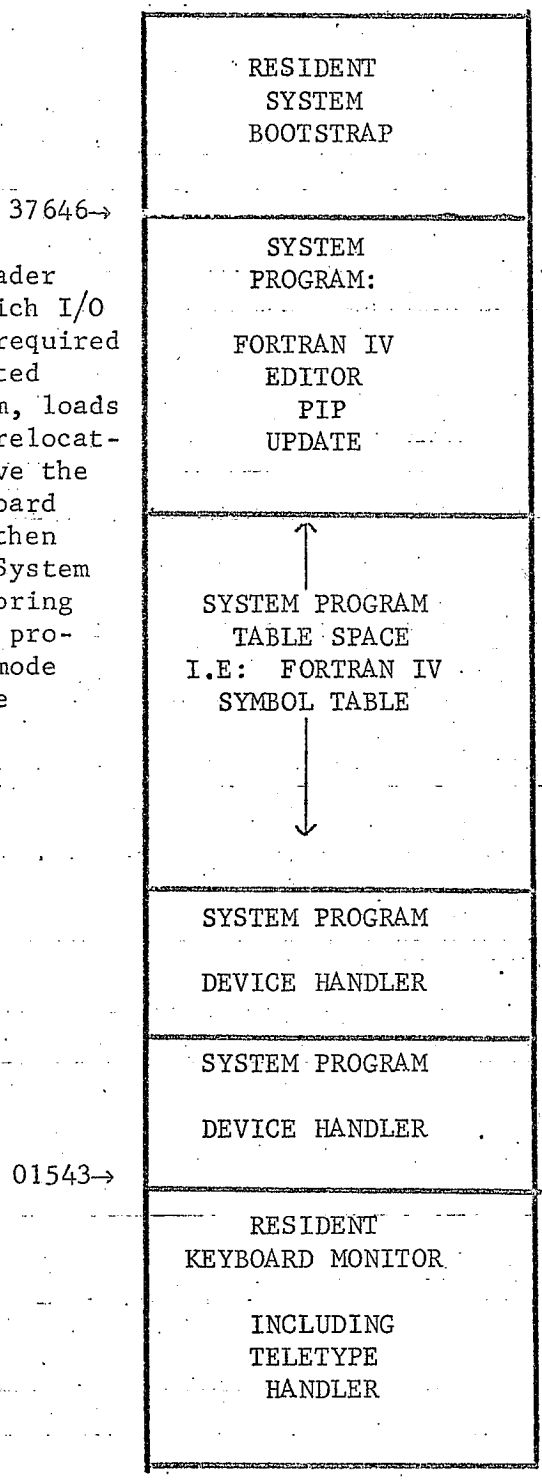
FIGURE 4 Memory Map: System Loader



(Addressed are in octal radix.)

The System Loader determines which I/O handlers are required by the requested system program, loads the handlers relocatably just above the resident Keyboard Monitor, and then modifies the System Bootstrap to bring in the system program in dump mode just below the Bootstrap.

FIGURE 5 Memory Map: System Programs



(Addresses are in octal radix.)

4.8 ↑C

Depressing CTRL and C simultaneously on the teletype keyboard (Monitor echoes ↑C) at any time forces re-initialization of the Keyboard Monitor which types

MONITOR V4B

$

to indicate that it is waiting for a keyboard command.

4.9 ↑P

Depressing CTRL and P simultaneously on the teletype keyboard (Monitor echoes ↑P) is used for the following functions: 1) to reinitialize or restart system programs, 2) to signal the paper-tape reader I/O handlers that the next tape in a series of tapes to be loaded is now ready in the high-speed reader, or 3) to signify termination of a Fortran PAUSE.

4.10 ↑S

Depressing CTRL and S simultaneously on the teletype keyboard (Monitor echoes ↑S) will start a user program after the Linking Loader has brought it into core via a $LOAD command.

4.11 ↑R

When the Keyboard Monitor detects a not-ready condition on a requested I/O device, it types .IOPS 4 on the teleprinter and halts. The user may then ready the device and continue by depressing CTRL and R simultaneously on the teletype keyboard (Monitor echoes ↑R).

4.12 ↑Q

A user may, at any time, pause and save the status of any running program by depressing CTRL and Q simultaneously on the teletype keyboard. After the Keyboard Monitor echoes ↑Q, type a DECtape unit number n, $0 \leq n \leq 7$. The current job will then be dumped, in core-image, onto blocks 101-200 of DECtape unit n (be sure to WRITE ENABLE DECtape unit n before requesting the dump). No error checking is performed on the unit number n. (It is important that the S switch in PIP be used to reserve blocks 101-200 on the file-structured DECtape prior to employing ↑Q. Previously saved information may be lost if this area is not reserved). Control returns to the Keyboard Monitor after the core-image dump is completed.

4.13 GET

The GET command retrieves the core-image dump stored on blocks 101-200 of DECtape unit n by a previous ↑Q command, and restores it to operation. The GET command has three forms: GET n will cause the job stored on DECtape unit n to be restarted where it was terminated; GET n XXXXX will cause the job stored on DECtape unit n to be restarted from location XXXXX; GET n HALT will cause the PDP-9 to halt after the job stored on DECtape unit n is restored to memory.

Users are urged to use ↑Q and GET to save their most frequently used and/or modified programs. The program may be prepared for a ↑Q core-image dump in either of two ways:

EXAMPLE 10:

```
$LOAD

LOADER V3A
>MYPROG
MYPROG
----
----
↑S↑Q3

$MONITOR V4B

$GET 3

↑S
```

or

EXAMPLE 11:

```
$GLOAD

LOADER V3A
>MYPROG
MYPROG
----
----
PAUSE ØØØØ ↑Q3

$MONITOR V4B

$GET 3

↑P
```

(User-typed commands are underlined above). In example 10, the user has written a normal FORTRAN IV program (MYPROG), loaded it, and responded to the ↑S by typing ↑Q3 to save MYPROG on DECtape unit 3 in core-image format. In example 11, the user has inserted a PAUSE statement as the first executable statement of MYPROG, requested a load and go (GLOAD), and responded to the PAUSE message by typing ↑Q3 to save MYPROG on DECtape unit 3 in core-image format. In both cases, the user may retrieve MYPROG via the Keyboard Monitor command GET 3. When DECtape 3 stops spinning, MYPROG may be started by typing ↑S (example 10) or ↑P (example 11).

5.0 LOADING USER PROGRAMS

All user programs are initially loaded via the Linking Loader. When the Linking Loader is loaded into core and is ready to accept an input command string from the keyboard, it will type:

```
LOADER
>
```

on the teleprinter. At this point, the input devices should be set up; if the paper-tape reader is being used, the tape-feed control button should be depressed momentarily, after the paper tape is loaded, to clear the reader-out-of-tape flag.

5.1 SPECIFYING USER PROGRAMS

If the user's binary program(s) are on a file-structured DECtape, the input command string to the Linking Loader should consist of the list of all file-names of all programs to be unconditionally loaded from the DECtape associated with .DAT slot -4. The main program must be requested first, followed by any desired subprograms. The file-names specified to the Linking Loader should agree with the names originally used in the compilation of the programs. The file-names should be one to six characters in length, with any characters over six being ignored. Only the file-names should be specified; the Linking Loader will automatically assume that the file-name extension is BIN (relocatable binary) and will search on both file-name and extension. A file-name, in the input command string, is terminated by a comma, a carriage-return, or an ALT MODE. ALT MODE terminates the input command string.

EXAMPLE 12: LOADER V3A
>MAIN
>SUB1
>SUB2 (ALT MODE)

EXAMPLE 13: LOADER V3A
>MAIN,SUB1,SUB2 (ALT MODE)

When the input device is not file-structured, n commas, followed by the ALT MODE character, prepares the Linking Loader to load n+1 programs from the input device.

EXAMPLE 14: (three programs to be loaded)
LOADER V3A
>,,(ALT MODE)

In either case, the subprograms should be loaded in order of decreasing size in order to take advantage of the Linking Loader's ability to fit programs into available memory space.

5.2 PROGRAM LOADING

After detection of an ALT MODE character, the LINKING LOADER will begin loading the user-specified program(s). If the input device is the paper-tape reader and the Linking Loader detects an end-of-tape condition, it will type ↑P on the teleprinter. Additional input should be placed in the paper-tape reader and the tape feed control button depressed momentarily to clear the reader-out-of-tape flag. Then, to continue, the user should type ↑P (CTRL P) on the keyboard.

After loading all programs requested in the keyboard input command string, the Linking Loader will attempt to resolve all unsatisfied subroutine requests by scanning the external user library (.DAT slot -5) and system library (.DAT slot -1), in that order. The external user library, if required, must be named .LIBR BIN; it can be constructed using the UPDATE system program.

When all requested programs are loaded and all library requests (explicit and implicit) are satisfied, the Linking Loader will either:

a. (If the Linking loader was called via LOAD) output ↑S and wait for the user to type ↑S (CTRL S) on the keyboard, then transfer control to the starting address of the user's main program, or

b. (If the Linking Loader was called via GLOAD) immediately transfer control to the starting address of the user's main program.

See figures 6 and 7 for memory maps of the Linking Loader and user programs.

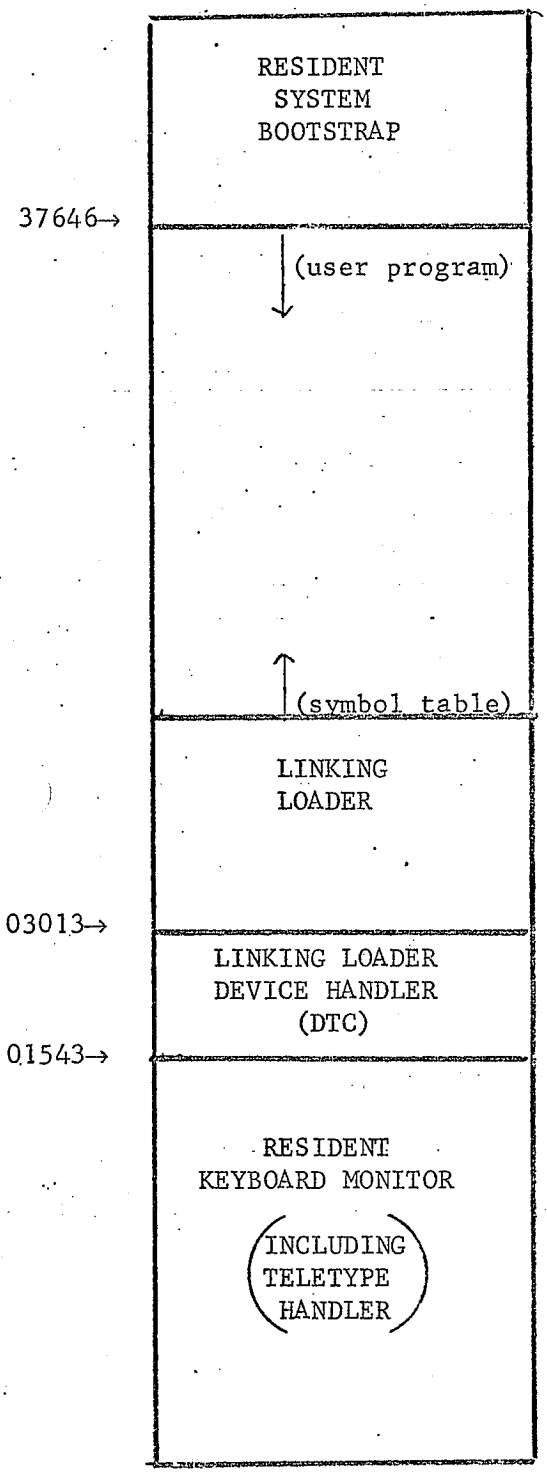
5.3 MEMORY MAP

The Linking Loader prints a memory map on the teleprinter during loading which consists of a list of names of programs loaded, a list of library subroutines loaded, and their starting load addresses (in octal radix). Any programs or subroutines required but not found, whether called explicitly or implicitly, are indicated with an address of 00000.

EXAMPLE 15:
```
LOADER V2A
>MAIN,SUB1,SUB2
MAIN    36421
SUB1    3635Ø
SUB2    35112
BCDIO   32114
STOP    321Ø3
FIOPS   31347
OTSER   31253
INTEGE  31123
```

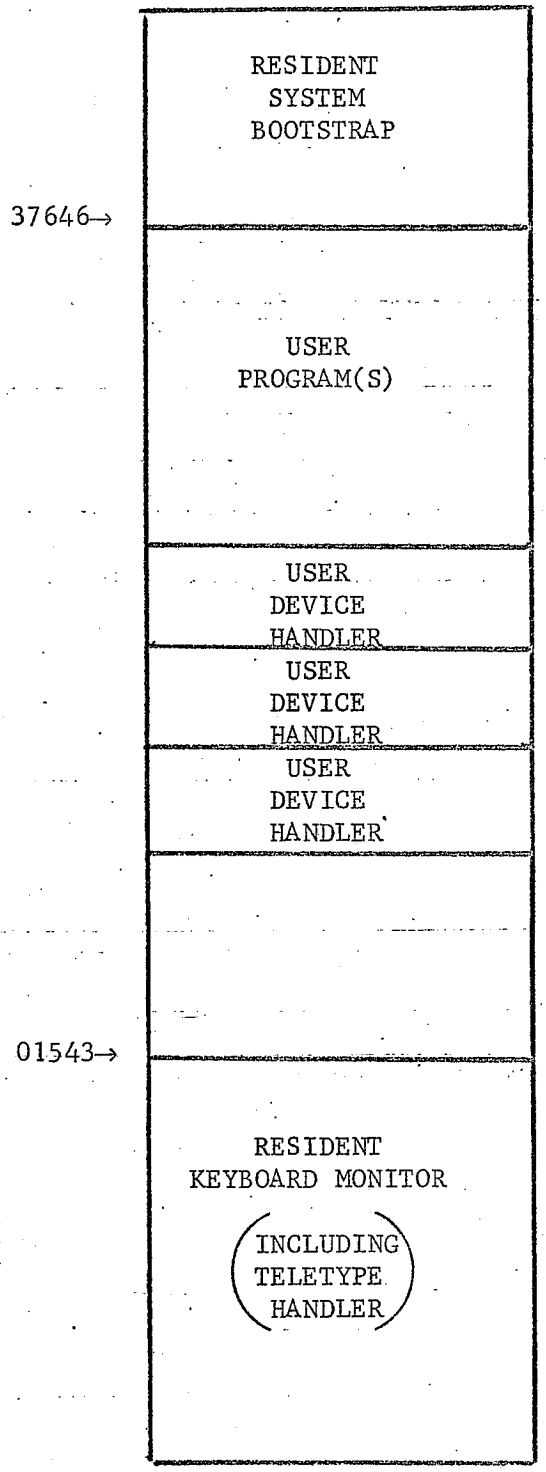
If any of the entries in the memory map have a 00000 address, loading was not successful; the cause of the trouble should be remedied and the loading procedure repeated.

FIGURE 6 Memory Map: Linking Loader



(Addresses are in octal radix.)

FIGURE 7 Memory Map: User Programs



(Addresses are in octal radix.)

5.4 ERROR MESSAGES

If a loading error occurs, the Linking Loader types:

.LOAD n

on the teleprinter and returns control to the Keyboard Monitor. The type of error is indicated by n (see Table 4).

TABLE 5. Linking Loader (or System Program Loader) Error Messages.

Error Code	Meaning	Explanation
1	memory overflow	The Linking Loader's symbol table and the user's program have overlapped. At this point the Linking Loader memory map will show the addresses of all programs loaded successfully before the overflow. Increased use of COMMON storage may allow the program to be loaded as COMMON storage can overlay the Linking Loader and its symbol table, since COMMON is not loaded into until run time.
2	input data error	parity error, checksum error, illegal data codes, or buffer overflow
3	unsatisfied global symbol	missing program
4	illegal .DAT slot request	A .DAT slot requested for use by a user program referenced an illegal .DAT slot number.

6.0 ERROR DETECTION

When the Keyboard Monitor or an I/O device handlers detects an illegal or improper situation, the Monitor Error Diagnostic routine assumes control, outputs an appropriate error message on the teleprinter, and waits (in a tight loop) for the user to specify the recovery procedure (↑P to restart a system program, ↑Q to save the current core-image, or ↑C to reload the Keyboard Monitor).

Unrecoverable error messages are output in the following format:

.IOPS nn XXXXXX

where nn is the error code (see Table 6) and XXXXXX is additional related system information.

If the Keyboard Monitor detects a "not ready" condition on an I/O device for which I/O has been requested, it will type:

.IOPS 4

on the teleprinter and halt. The user should make the device ready and type ↑R (CTRL R) on the keyboard to continue.

TABLE 6. Keyboard Monitor Error Codes.

Error Code	Meaning
00	Illegal Keyboard Monitor call
01	Illegal Keyboard Monitor call
02	.DAT slot error
03	Illegal interrupt
(04	Device not ready)
05	Illegal Keyboard Monitor call
06	Illegal I/O handler function requested
07	Illegal data mode requested
10	Old file still active when new file requested on same .DAT slot
11	SEEK/ENTER not executed prior to READ/WRITE on file-structured DECtape
12	Unrecoverable DECtape error
13	File not found during SEEK
14	Directory full on ENTER
15	DECtape full on WRITE
16	Output buffer overflow (too large for device)
17	Too many files active for this handler
23	Illegal word pair count (illegal I/O buffer size requested)
30	API software level error
70	Illegal dataphone transmit interrupt
71	Illegal dataphone receive interrupt
72	Dataphone transmit logic is inactive when it should be active
74	Clock queue overflow - too many clock requests active
75	A user LTSH or LTCH HANDLER took longer than 16.7 msec to execute
76	Clock is off when it should be on
77	A nonpositive clock interval was requested

The following ten sheets are an update to the PDP-9 LINKAGE (yellow DIGITAL) section of the hybrid lab user's manual. Real-time clock software has been completely revised.

TABLE OF CONTENTS

2.3 PROGRAM: Initialize (PDP-9 only)

NAME: INIT9

PURPOSE: 1. To provide the necessary control and linkage for timer and software level interrupt processing.

AUTHOR: C. L. Cross

CALLING SEQUENCE: CALL INIT9

ARGUMENTS: none

EXECUTION TIME: 22 μsec

STORAGE REQUIREMENTS: 81 locations

COMMENTS: 1. INIT9 is provided as an abbreviated version of INIT (see hybrid interface linkage routines write-up) for those FORTRAN IV programs which use the linkage routines in this section only. In such a situation, either INIT or INIT9 (but not both) may be used.

2. A call INIT9 statement is required before execution of ISLI.

3. Havoc results if a user attempts to load both INIT and INIT9 from one FORTRAN IV program.

4. INIT9 may be called more than once: each time it is called, the timer will be disabled and the API system will be re-initialized.

2.5 PROGRAM: Suspend API

NAME: SAPI

PURPOSE: 1. Disable the Automatic Priority Interrupt (API) system.

AUTHOR: C. L. Cross

CALLING SEQUENCE: CALL SAPI

ARGUMENTS: none

EXECUTION TIME: 7 μsec

STORAGE REQUIREMENTS: 4 locations

COMMENTS: 1. If the API system is already disabled, execution of SAPI will have no effect.

3.0 REAL-TIME CLOCK

The real-time clock generates a clock pulse every 16.7 msec* (60cps) to increment a time counter stored in system memory. The counter initiates a program interrupt on API level 1 when a programmed preset time interval is completed. The clock can be enabled or disabled under program control. The console CLK switch must be in the down position (or the entire console LOCKed) to permit programmed control of the facility. The clock remains disabled with the switch in the up position. Depressing the I/O RESET console key also disables the facility.

System software for the real-time clock is based on a 4-deep interval request queue. The real-time clock is enabled by the first program request for a time interval, and disabled only if all program segments which have requested time intervals have subsequently requested the clock be disabled. Whenever the real-time clock is enabled, the system software will increment a "master system clock" every time the real-time clock generates a clock pulse. When a program segment requests notification after a specified time interval, the system software calculates and stores in a 4-register queue the value which the master system clock will have when the requested time interval has elapsed. Each time the master system clock is incremented, its value is compared against all queue registers and program segments are notified accordingly if one or more matches are found.

All queue registers are automatically freed whenever the clock is disabled. They are not (normally) released when the user interrupts program execution via a ↑P. It is therefore possible for the queue to overflow after several successive ↑P program restarts (Monitor Error Diagnostic IOPS 74).

*NOTE: The first clock pulse after enabling the clock facility occurs anywhere from 0 to 17 msec later.

Four linkage routines are provided for control of the clock facility. The TIME routine provides the user with an elapsed-time clock. LTSH provides a single time interval for applications such as time delays. LTCH provides periodic time interval markers for applications such as regular data transferrals. DTIM requests that the real-time clock be disabled.

The DCT-2000 line printer handler is the only system routine which requests time intervals. Thus, if the line printer, TIME, LTSH, and LTCH are all actively requesting time intervals, then the clock queue will be full; i.e., four queue registers are sufficient unless the user creates an abnormal condition by using the ↑P abort too liberally.

System software for the real-time clock occupies 178 locations. It is loaded into core if the user program includes the DCT-2000 line printer handler, TIME, LTSH, LTCH, or DTIM.

3.1 PROGRAM: Elapsed Time Clock

NAME: TIME

PURPOSE: 1. To provide an elapsed-time clock which can be referenced from a FORTRAN IV program.

AUTHOR: C. L. Cross

CALLING SEQUENCE: CALL TIME (ICYCLE, ISEC1Ø, ISECOND, IMINUTE, SWITCH)

ARGUMENTS: ICYCLE specifies the FORTRAN IV variable to be assigned the 60 cycle-per-second elapsed time count.

ISEC1Ø specifies the FORTRAN IV variable to be assigned the tenths-of-a-second part of the elapsed time clock.

ISECOND specifies the FORTRAN IV variable to be assigned the seconds part of the elapsed time clock.

IMINUTE specifies the FORTRAN IV variable to be assigned the minutes part of the elapsed time clock.

SWITCH specifies when TIME is to be disabled: the cycles-per-second count and the elapsed time clock are frozen at their last values at the first real-time clock pulse after SWITCH is set to .FALSE.

EXECUTION TIME: at least 129 μsec

STORAGE REQUIREMENTS: 92 locations

COMMENTS: 1. ICYCLE, ISEC1Ø, ISECOND, and IMINUTE are all cleared to zero when TIME is called.

2. At any time thereafter, ICYCLE equals the number of 1/60-second intervals which have elapsed since TIME was called, and IMINUTE: ISECOND.ISEC1Ø equals the elapsed time since TIME was called. The following relation is always true after calling TIME: ICYCLE = 36Ø * IMINUTE + 6Ø * ISECOND + 1Ø * ISEC1Ø.

3. TIME automatically requests that the real-time clock be enabled each time it is called, unless the previous call has not been disabled by setting SWITCH = .FALSE.

4. TIME automatically requests that the real-time clock be disabled whenever it detects that SWITCH = .FALSE.

5. Subsequent calls of TIME without setting SWITCH = .FALSE.

in the meantime will have the effect of resetting (to zero) the cycles-per-second count and the elapsed-time clock, and (optionally) changing the associated FORTRAN IV variables.

```
EXAMPLE 2:  Z = .TRUE.
            CALL TIME (IC,IS1Ø,IS,IM,Z)
          A{      }
            Z = .FALSE.
            WRITE (4,1Ø) IC,IM,IS,IS1Ø
         1Ø FORMAT (1H ,7HCYCLES=,I6,14H ELAPSED TIME=,I2,1H:,I2,1H.,I2)
```

result: The time taken to execute the code at A will be output to the teleprinter as follows:
CYCLES= 1216 ELAPSED TIME= 3: 2. 1
i.e., 1216 1/60-second intervals, or 3 minutes, 2.1 seconds.

3.2 PROGRAM: Load Timer Once and Set Handler

NAME: LTSH

PURPOSE: 1. To cause an interrupt to occur after the specified time interval.

AUTHOR: C. L. Cross

CALLING SEQUENCE: CALL LTSH (ITIME, HANDLER, IERROR)

ARGUMENTS: ITIME specifies the time interval before the interrupt as follows: t = (ITIME)*(16-2/3 msec), where ITIME > 0.

HANDLER specifies the parameter-less FORTRAN IV user subroutine which is to handle the clock overflow interrupt when it occurs.

IERROR = 0 no error
= 1 illegal ITIME requested (ITIME ≤ 0)

EXECUTION TIME: 81 to 96 μsec

STORAGE REQUIREMENTS: 41 locations

COMMENTS: 1. A second call of LTSH will cause unpredictable results if it is executed before the previously-requested LTSH time interval is finished and the associated HANDLER has been initiated.

2. Calling LTSH, from subroutine HANDLER will result in unpredictable errors if subroutine HANDLER does not RETURN within one (new) time interval after calling LTSH.

3. LTSH automatically requests that the real-time clock be enabled each time the user requests an LTSH time interval.

4. LTSH automatically requests that the real-time clock be disabled every time a requested time interval has elapsed and before initiating the appropriate HANDLER.

5. The PDP-9 will hang up if any I/O is attempted from HANDLER.

PROGRAMMING HINTS: HANDLER must be declared EXTERNAL in any FORTRAN IV program which contains a CALL LTSH statement

3.3 PROGRAM: Load Timer Continuously and Set Handler

NAME: LTCH

PURPOSE: 1. To cause the occurrence of variable-frequency real-time interrupts whose period is independent of interrupt processing time.

AUTHOR: C. L. Cross

CALLING SEQUENCE: CALL LTCH (ITIME, HANDLER, IERROR)

ARGUMENTS: ITIME specifies the time interval between interrupts as follows: t = (ITIME)*(16-2/3 msec), where ITIME > 0.

HANDLER specified the parameter-less FORTRAN IV user subroutine which is to handle the clock overflow interrupts when they occur.

IERROR indicates errors as follows:
IERROR = 0 no error
= 1 illegal ITIME request (ITIME ≤ 0)

EXECUTION TIME: 81 to 96 μsec

STORAGE REQUIREMENTS: 42 locations

COMMENTS: 1. The interrupts will occur with a constant period of t, regardless of the execution time of HANDLER. Thus, results are unpredictable if HANDLER has an execution time greater than t.

2. A second call of LTCH will cause unpredictable results if it is executed from anywhere other than the HANDLER of the previously-requested LTCH.

3. Calling LTCH from subroutine HANDLER will result in unpredictable errors if subroutine HANDLER does not RETURN within one (new) time interval after calling LTCH.

4. LTCH automatically requests that the real-time clock be enabled each time the user requests LTCH time intervals.

5. LTCH never requests that the real-time clock be disabled.

6. The PDP-9 will hang up if any I/O is attempted from HANDLER.

PROGRAMMING HINTS: HANDLER must be declared EXTERNAL in any FORTRAN IV program which contains a CALL LTCH statement.

3.4 PROGRAM: Disable Timer (Request)

NAME: DTIM

PURPOSE: 1. To request that the real-time clock be disabled.

AUTHOR: C. L. Cross

CALLING SEQUENCE: CALL DTIM

ARGUMENTS: none

EXECUTION TIME: 21 (real-time clock stays on) or 25 (real-time clock is turned off) μsec

STORAGE REQUIREMENTS: 4 locations

COMMENTS:

1. DTIM only requests that the real-time clock be disabled. The real-time clock actually goes off only if all program segments which have requested gueue registers have subsequently requested that the real-time clock be disabled.

2. If DTIM is called more times than the sum of LTCH calls and TIME calls which have not been disabled by setting SWITCH.= FALSE., results are unpredictable.

3. If DTIM does not actually cause the real-time clock to turned off (due to other active queue requests), then LTCH interrupts will continue to occur and/or the cycles-per-second count and elapsed-time clock will continue to increment. In the former case, the user should alter the LTCH HANDLER (or provide a new one) which will ignore the interrupts (do-nothing). In the latter case, the user should freeze the cycles-per-second count and elapsed-time-clock by setting SWITCH = FALSE. (which also generates an additional real-time clock disabling request).

The following two sheets are an update to the 680 OPERATION (blue ANALOG) section of the hybrid lab user's manual. The 680 DVM now converts on a 60-cycle clock regardless of whether the 680 is operating stand-alone or hybrid.

TABLE OF CONTENTS

4.0 CAUTIONS AND HINTS

4.1 CAUTIONS

1) Patching amplifier output to another amplifier output will cause overload and result in amplifier damage.
2) Patch from cold to hot: i.e., input to output and input to reference.
3) Patching computer reference directly into a function relay is dangerous since an inadvertant short will weld the relay contacts together.
4) Prolonged overloads reduce the life of amplifiers and should thus be avoided.
5) A potentiometer that fails to set properly will drive continuously or until the clear (CL) button is pushed. If allowed to drive damage to the servo motor and/or servo power supply will result.

4.2 HINTS

1) Overloading of all amplifiers indicates patch panel is not engaged.
2) The 6 handset pots, prefixed by a Q, are not clearly marked on the patch panel and may inadvertantly be used as a servo pot.
3) A D/A switch must be patched to a summing junction of an amplifier.
4) Patching a pot into a multiplier or function generator causes errors due to their non-linear input impedance.
5) The indicator to the left of and below the DVM lights whenever the 680 console is selected by the PDP-9.

This write-up replaces the HYSAT (cherry red HYBRID) section of the hybrid lab user manual.

USER MANUAL

HYSAT

Hybrid Static Analog Test Program

Carnegie-Mellon University
Hybrid Computation Laboratory
January, 1969

TABLE OF CONTENTS

INDEX OF TABLES

1.0 INTRODUCTION

The primary purpose of HYSAT (Hybrid Static Analog Test Program) is to determine the operational status of the EAI 680 analog computer and the EAI 693 hybrid interface as part of the standard preventive maintenance procedures of the hybrid laboratory. In addition, HYSAT may be used to perform the initial setup (potentiometers, modes, time constants, function relays, output control lines, and digital-to-analog converters) and static checkout of any analog/hybrid problem.

2.0 OPERATING PROCEDURES

2.1 INITIALIZATION

2.1.1 Loading

HYSAT may be loaded via the following Keyboard Monitor commands:

MONITOR V4B		MONITOR V4B
$ASSIGN DTCØ -4	or	$A DTCØ -4
$EXECUTE HYSAT		$E HYSAT

2.1.2 Console Initialization

After HYSAT is loaded and the user types CTRL S, HYSAT will execute INIT to select the 680 console and prepare the computers for hybrid operation.

2.1.3 Program Initialization

After a successful INIT, HYSAT will then initialize program switches and parameters; i.e., status readout mode is turned OFF, error messages are allowed (YES), and the error tolerance is set to 5 millivolts (ER5).

2.2 SPECIFYING INPUT UNIT

When HYSAT is properly initialized and ready to accept user instructions, it types:

```
HYSAT
TYPE INPUT DIRECTIVE- 1-TTY,2-PAPER TAPE,3-DECTAPE,4-CARD READER
```

on the teleprinter. Type the desired directive* in column 1 and carriage-return. HYSAT will record the desired input unit (IU).

2.2.1 DECtape Input

If DECtape is specified as the input unit, HYSAT will respond with:

```
MOUNT INPUT TAPE
TYPE UNIT NUMBER,FILE NAME- NFILENMEXT
$
```

*N.B.: The DCT-2000 may be assigned to either IU (card reader) or OU (line printer), but not both at the same time.

on the teleprinter. Type the desired unit number (1, 2, or 3), file-name, and file-name extension in columns 1-10 and carriage-return. HYSAT will record the desired input unit number and file-name.

2.3 SPECIFYING OUTPUT UNIT

After the input unit has been specified and recorded, HYSAT types:

```
TYPE OUTPUT DIRECTIVE- 1-TTY,2-PAPER TAPE, 3-DECTAPE,4-LINE PRINTER
$
```

on the teleprinter. Type the desired directive* in column 1 and carriage-return. HYSAT will record the desired output unit (OU).

2.3.1 DECtape Output

If DECtape is specified as the output unit, HYSAT will respond with:

```
MOUNT OUTPUT TAPE
TYPE HANDLER NUMBER,FILE NAME- NFILENMEXT
$
```

on the teleprinter. Type the desired unit number (1, 2, or 3), file-name, and file-name extension in columns 1-10 and carriage-return. HYSAT will record the desired output unit number and file-name.

2.4 OPERATION

After recording input and output units, HYSAT types:

```
ENGAGE PATCH PANEL FOR TEST
TYPE CR WHEN READY
$
```

on the teleprinter. The user should engage the patch panel, set-up IU and OU if they are not already set-up, and carriage-return (CR) when ready to begin the test.

HYSAT responds to the carriage-return by reading data statements from IU and performing the specified operations. If IU is the teletype keyboard HYSAT will type

```
$
```

---

*N.B.: The DCT-2000 may be assigned to either IU (card reader) or OU (line printer), but not both at the same time.

on the teleprinter to indicate readiness to accept a data statement. Error messages will be output by HYSAT as they are detected and as directed by the error message allow/suppress data statements, along with the offending data statements in some cases.

Upon encountering an END data statement, HYSAT outputs:

END OF TEST

on both the teleprinter and OU and then transfers control to the continuation directive option (below).

2.5 CONTINUATION DIRECTIVE

When HYSAT finishes a set of operations (user data statements) by detecting an END statement on IU, it types:

TYPE CONTINUATION DIRECTIVE-
Ø-CONTINUE,1-TERMINATE,2-RESTART,3-CHANGE IU/OU
$

on the teleprinter. Type the desired continuation directive in column 1 and carriage-return.

2.5.1 Continue

If 0 is specified as the continuation directive, HYSAT will return to the operating mode (2.4 OPERATION) unless OU is DECtape, in which case control is transferred to (2.3.1 DECtape Output) in order to allow specification of a new output file. (IU is reOPENed from its beginning if it is DECtape).

2.5.2 Terminate

If 1 is specified as the continuation directive, control is returned to the Keyboard Monitor.

2.5.3 Restart

If 2 is specified as the continuation directive, HYSAT restarts from the beginning (2.1.3 Program Initialization); i.e., program switches and parameters are reset to their standard values.

2.5.4 Change IU/OU

If 3 is specified as the continuation directive, HYSAT restarts from (2.2 SPECIFYING INPUT UNIT); i.e., program switches and parameters are not reset to their standard values.

3.0 DATA STATEMENTS

HYSAT has 16 basic data statement types. The statements may be input in any order; the only restrictions are those imposed by the analog/hybrid environment. The characters of an input record are scanned until the last field is read or until 36 columns have been processed; the remaining characters of an input record may be used for comments. HYSAT recognizes one or more consecutive blanks or a comma as a single field delimiter. No delimiter is required between alphanumeric and integer fields. HYSAT will interpret empty or non-existent fields as zeroes (for integer fields) blanks (for alphanumeric fields) or "+" (for sign fields). All data statements may thus be free-form records. The data statement repertoire is summarized in Table 1.

Each data statement type is defined via a "form" of the general type: XX,N,$\pm$,M,I,J,E,S, a description of those fields required and/or optional with this data statement type, the result of successfully executing this data statement type and one or more examples. XX and E are alphanumeric fields; N,M,I,J, and S are integer fields; and $\pm$ is a sign field. The fields "I,J" and "E,S" are described first, as they are generally applicable to all data statement types for which they are legal.

3.1 DATA STATEMENT REPETITION

form: ...,I,J,...

where: I specifies the type of data statement repetition as follows:

I = blank (or empty) - no data statement repetition (the J field is still scanned, but any integral value for J will be ignored)

I = $\emptyset$ - execute this data statement J times, where J $\geq \emptyset$

I > $\emptyset$ - "implicit DO loop" - The data statement is executed repeatedly, with the value of field N being incremented by the value of field I after each execution, until the incremented value of field N is greater than the value of field J.

In FORTRAN notation:

```
      DO 1Ø NN = N,J,I
      < execute data statement XX,NN,±M,E,S >
1Ø CONTINUE
      < go to next data statement >
```

In algebraic notation:

the data statement is executed c times, using N = N,N+I,N+2I,...,N+(c-1)I, where c is the smallest integral value for which N + cI > J is true.

J specifies the number of executions (I = Ø) or the terminal value for N + cI (I > Ø), as described above.

EXAMPLE 1: XX,N,±,M,E,S
result - execute data statement XX,N,±,M,E,S (once).

EXAMPLE 2: XX,N,±,M,Ø,3,E,S
result - execute data statement XX,N,±,M,E,S three times
Note that XX,N,±,M,Ø,3,E,S
is equivalent to XX,N,±,M,E,S
XX,N,±,M,E,S
XX,N,±,M,E,S

EXAMPLE 3: XX,3,±,M,2,5,E,S
result - execute data statement XX,3,±,M,E,S
and data statement XX,5,±,M,E,S

3.2 ERROR CONTROL OPTION

form: ...,E,S

where: E specifies the action to be taken if a hardware error is detected during execution of the data statement:

- E = blank - ignore the error and continue with data statement execution.
- E = E - skip S data statements from IU after execution of this data statement if a hardware error <u>has</u> occurred while processing this data statement.
- E = H - immediately pause execution of this data statement and transfer control to the hardware error recovery routine (see 4.5.1) if a hardware error is detected while processing this data statement.
- E = S - skip S data statements from IU after execution of this data statement if a hardware error <u>has not</u> occurred while processing this data statement.

S specifies the number of succeeding data statements from IU to skip if hardware errors <u>are</u> detected (E = E) or if hardware errors <u>are not</u> detected (E = S) during execution of this data statement, as described above.

EXAMPLE 4: XX,N,±,M,I,J
result - Execution of the data statement XX,N,±,M,I,J is unaffected by machine errors.

EXAMPLE 5: XX,N,±,M,E4
result - If a hardware error is detected while processing the data statement XX,N,±,M, execution resumes with the 5th data statement following this data statement on IU. If no hardware error is detected while processing the data statement XX,N,±,M, execution resumes with the next consecutive data statement on IU.

EXAMPLE 6: XX,N,±,M,I,J,H
result - If a hardware error is detected while processing the data statement XX,N,±,M,I,J, execution of the data statement is immediately paused and control is transferred to the hardware error recovery routine (see 4.5.1). If no hardware errors are detected while processing the data statement XX,N,±,M,I,J, data statement execution is unaffected.

EXAMPLE 7: XX,N,±,M,Ø,6,S3
result - If a hardware error occurs while processing any of the six repetitions of the data statement XX,N,±,M, execution resumes with the next consecutive data statement on IU after the 6th repetition of XX,N,±,M. If no hardware error occurs while processing all six repetitions of the data statement XX,N,±,M, execution resumes with the 4th data statement following this data statement on IU.

3.3 ANALOG MODE CONTROL

form: XX,E,S

where: XX specifies the analog mode as follows:
- PC - potentiometer coefficient
- PP - patch panel
- ST - static test
- OP - operate
- HD - hold
- IC - initial condition
- SP - set potentiometer

E,S specifies the error control option as described in Section 3.2.

result: The analog mode is set to the specified state.

EXAMPLE 8: PC,E3
result: The analog mode is set to PC.
If the analog mode fails to set properly, the next 3 data statements from IU are skipped.

3.4 TIME CONSTANT CONTROL

form: XX,E,S

where: XX specifies the time constant as follows:
- NS - normal seconds (real time)
- FS - fast seconds (10 times real time)
- NM - normal milliseconds (1,000 times real time)
- FM - fast milliseconds (10,000 times real time)

E,S specifies the error control option as described in Section 3.2.

result: The time constant is set to the specified value.

EXAMPLE 9: NS,H
result - The analog time constant is set to normal seconds. If the time constant fails to set properly, control is passed to the hardware error recovery routine.

3.5 LOGIC MODE CONTROL

form: XX,E,S

where: XX specifies the logic mode as follows:
- RUN
- STOP
- CLEAR

E,S specifies the error control option as described in Section 3.2.

result: The logic mode is set to the specified state.

EXAMPLE 10: STOP,S6
result - The logic mode is set to STOP. If the logic mode sets properly, the next 6 data statements from IU are skipped.

3.6 ERROR TOLERANCE CONTROL

form: ER,N

where: N specifies the magnitude of the desired error tolerance in machine units times 10^{-4}; i.e., N = 1234 would specify an error tolerance of ± 0.1234 machine units (± 1.234 volts).

result: The error tolerance is set to the specified magnitude. The standard value for the error tolerance, set when HYSAT is first loaded and after every RESTART (see 2.5.3), is ± 0.0005 machine units (± 5 millivolts).

EXAMPLE 11: ER25
result - The error tolerance is set to ± 0.0025 machine units (25 millivolts).

3.7 ANALOG COMPONENT SELECTION

form: XX,N,±,M,I,J,E,S

where: XX specifies the component type as follows:
- A - amplifier
- D - derivative
- F - function generator
- T - trunk
- P - potentiometer (servo-set)
- PB - patch-board (patch panel - code "PP" is already used for analog mode control)
- D/ - derivative/10
- Q - quotentiometer (hand-set potentiometer)

N specifies the integer patchboard address (0-119) of the specified component type.

±,M specifies the desired value of the addressed component in machine units times 10^4; i.e., 3278 would specify +0.3278 machine units (+3.278 volts).

I,J specifies data statement repetition as described in Section 3.1.

E,S specifies the error control option as described in Section 3.2.

result: In all analog modes except SP, HYSAT will read the DVM value of the specified component and compare it to the specified desired value, comparing the difference against the error tolerance. In the SP mode, and if a servo-set potentiometer is specified, HYSAT will attempt to set the specified potentiometer coefficient to the given desired value ± the error tolerance.

EXAMPLE 12: SP
P22, 5000,1,25
result: The analog mode is set to SP.
The coefficients of potentiometers 22, 23, 24 and 25 are set to 0.5000 machine units (5 volts).
Hardware errors are ignored.

EXAMPLE 13: A19+5000H
result: The DVM value of amplifier 019 will be compared against +0.5000 machine units (+5 volts). If the difference is within the error tolerance, execution continues with the next successive data statement from IU. If the difference is greater than the error tolerance, data statement execution pauses and control transfers to the hardware error recovery routine (see 4.5.1).

3.8 ANALOG-TO-DIGITAL CONVERTER CHECKOUT

form: AD,N,±,M,I,J,E,S

where: N specifies the ADC channel 0-14 to be checked.
±,M specifies the desired value of the addressed ADC channel in machine units times 10^4; i.e., -3278 would specify -0.3278 machine units (-3.278 volts).

I,J specifies data statement repetition as described in Section 3.1.
E,S specifies the error control option as described in Section 3.2.

result: The analog value of the analog-to-digital converter channel specified is converted to digital, the digital value is unnormalized, and the unnormalized value is then compared to the specified desired value, the difference being compared against the error tolerance.

EXAMPLE 14: AD 3-5000
result - Analog-to-digital converter channel 3 is converted, the result is converted to an unnormalized value and compared against -0.5000 machine units (-5 volts) $\pm$ the error tolerance.
Hardware errors are ignored.

3.9 DIGITAL-TO-ANALOG CONVERTER SET-UP

form: DA,N,$\pm$,M,I,J

where: N specifies the DAC channel 0-11 to be loaded and transferred.
$\pm$,M specifies the unnormalized value, in machine units times 10^4, to be loaded into the addressed DAC channel and transferred.
I,J specifies data statement repetition as described in Section 3.1.

result: The specified unnormalized value is loaded into the specified DAC channel and transferred.

EXAMPLE 15: DA3-5000
result: -0.5000 machine units (-5 volts) is loaded into DAC channel 3 and transferred.

3.10 STATUS READOUT MODE

form: ON or OFF

result: When the status readout mode is "ON," the value of all addressed analog components or analog-to-digital converter channels is output to OU in a format suitable for subsequent reading by HYSAT; the values read are not compared against the "$\pm$,M" fields, which are completely ignored. Potentiometers cannot be set when the status readout mode is "ON." Data statements other than analog component selection and analog-to-digital converters checkout are not affected by the status readout mode. Status readout mode is intended to facilitate saving the status (especially potentiometers) of an analog/hybrid program. The standard state of status readout mode, set when HYSAT is first loaded and after every RESTART (see 2.5.3), is OFF.

EXAMPLE 16: PC
ON
A23
OFF
result: The analog mode is set to PC.
State readout mode is turned ON.
The value of amplifies 023 = +0.0000 machine units is read from the DVM.
A 23 0 is output to OU.
State readout mode is turned OFF.
Hardware errors are ignored.

3.11 LOGIC COMPONENT SETTING

form: XX,N,±,I,J

where: XX = FR to denote function relays or CL to denote output control lines.
N specifies the output control line number 0-15, or the function relay patchboard address as follows:

0-004	4-034	8-064	12-094
1-009	5-039	9-069	13-099
2-014	6-044	10-074	14-104
3-019	7-049	11-079	15-109

or N = 16 to specify all of the function relays or all of the output control lines.
± specifies the desired state of the logic component(s): + for function relays plus or output control lines high, - for function relays minus or output control lines low.
I,J specifies data statement repetition as described in Section 3.1.

result: The specified logic component(s) is (are) set to the specified state.

EXAMPLE 17: FR-2,14
result: Function relays 004,014,034,044,064,074,094, and 104 are set to the minus state.

EXAMPLE 18: CL16
result: All output control lines are set to the logical high state.

3.12 LOGIC COMPONENT READOUT

form: XX,N,±I,J,E,S

where: XX = CS to denote comparators or SL to denote sense lines.
N specifies the sense line number 0-7, or the comparator patchboard address as follows:

0-004	4-034	8-064	12-094
1-009	5-039	9-069	13-099
2-014	6-044	10-074	14-104
3-019	7-049	11-079	15-109

or N = 8 to specify all sense lines, or N = 16 to specify all comparators.

± specifies the desired state of the logic component(s): + for logical high, - for logical low.

I,J specifies data statement repetition as described in Section 3.1.

E,S specifies the error control option as described in Section 3.2.

result: The state(s) of the specified logic component(s) is (are) checked against the specified desired state. Sense lines are reset after each test.

EXAMPLE 19: SL8-E7

result: All sense lines are tested for the logical low state. If one or more sense lines are in the logical high state, they are reset and the next 7 data statements on IU are skipped.

EXAMPLE 20: CS,3 5

result: Comparators 004 and 019 are checked for the logical high state.
Hardware errors are ignored.

3.13 GENERAL PURPOSE INTERRUPT CHECKOUT

form: GP,N,±,I,J,E,S

where: N specifies the general purpose interrupt number 0-7 or N = 8 to specify all general purpose interrupts.

± specifies the desired state of the general purpose interrupt(s): + for logical high and interrupt condition, - for logical low and no interrupt.

I,J specifies data statement repetition as described in Section 3.1.

E,S specifies the error control option as described in Section 3.2.

result: The specified general purpose interrupt(s) is (are) first checked for the specified desired state; then the specified general purpose interrupt(s) is (are) unmasked, checked for occurrence or non-occurrence of the interrupt(s), and finally masked off.

EXAMPLE 21: GP2+H

result: General purpose interrupt 2 is first checked for the logical high state; then it is unmasked, checked for actual occurrence of the interrupt, and masked off. If either test fails, control is immediately transferred to the hardware error recovery routine.

EXAMPLE 22: GP3-2,5E2

result: General purpose interrupts 3 and 5 are checked for the logical low state; and unmasked, checked that the associated interrupt does not occur, and masked off. If either general purpose interrupt is in the logical high state, or either interrupt actually occurs, the next 2 data statements from IU are skipped.

3.14 ERROR MESSAGE OPTION

form: YES or NO

result: If the error message option is YES, all hardware errors cause the appropriate error message to be outputted to OU. If the error message option is NO, no hardware error messages are output on OU (unless the error control option is H), but the error control option is still effective. The standard state of the error message option, set when HYSAT is first loaded and after every RESTART (see 2.5.3) is YES.

EXAMPLE 23: ER20
PC
NO
A0+1000S3

result: The error tolerance is set to ±0.0020 machine units (±20 millivolts).
The analog mode is set to PC.
The value of amplifier 0 = 0 machine units (0 volts) is read from the DVM and compared to +0.1000 machine units (+1 volt) with an error tolerance of ± 20 millivolts.
The test fails: no error message is output to OU, and execution resumes with the next consecutive data statement from IU (3 data statements from IU are not skipped).

3.15 BLANK

form:

result: No execution is caused by this data statement. Its primary purpose is to terminate temporary teletype keyboard input.

3.16 SKIP

form: SK,N

where: N specifies the number of consecutive data statements from IU which are to be skipped.

result: The next N data statements from the current IU are read and ignored. Data statement execution resumes with the (N + 1)st data statement following the SK,N data statement on IU.

EXAMPLE 24: SK13
result: The next 13 data statements from IU are skipped.

3.17 COMMENT

form: CO, < any desired character string >

result: The data statement is output to OU and otherwise ignored.

EXAMPLE 25: CO THIS IS A COMMENT.
result: The data statement is output to OU.

3.18 END

form: END

result: END OF TEST is output to the teleprinter and to OU (unless they are the same), IU and OU are CLOSEd if they are DECtape, and control is transferred to the continuation directive option (see 2.5).

TABLE 1. Data Statement Summary

TYPE	USE	CODE	ARGUMENTS
analog mode control	potentiometer coefficient	PC	E SS
	patch panel	PP	
	static test	ST	
	operate	OP	
	hold	HD	
	initial condition	IC	
	set potentiometer	SP	
time constant control	normal seconds	NS	E SS
	fast seconds	FS	
	normal milliseconds	NM	
	fast milliseconds	FM	
logic mode control		RUN	E SS
		STOP	
		CLEAR	
error tolerance control		ER	NNNN
analog component selection	amplifier	A	NNN ± MMMM III JJJ E SS
	derivative	D	
	function generator	F	
	trunk	T	
	potentiometer	P	
	patch board	PB	
	derivative/10	D/	
	hand-set potentiometer	Q	
analog-to-digital converter	checkout	AD	NN ± MMMM II JJ E SS
digital-to-analog converter	set-up	DA	NN ± MMMM II JJ
logic component setting	function relays	FR	NN ± II JJ
	control lines	CL	
logic component readout	sense lines	SL	N ± I J E SS
	comparators	CS	NN ± II JJ E SS
general purpose interrupt	checkout	GP	N ± I J E SS
status readout mode	on	ON	
	off	OFF	
error message option	allow output	YES	
	suppress output	NO	
blank			
skip		SK	NN
comments		CO	<any desired character string
end	stop execution	END	

4.0 ERROR PROCESSING

4.1 INITIALIZATION ERROR

If HYSAT detects an error condition while attempting to INIT the EAI 680 and/or EAI 693, the following message will be output on the teleprinter:

CONSOLE FAILED TO SELECT

and control will be returned to the Keyboard Monitor.

4.2 SKIP ERRORS

If a user-specified skip (through an E or S error switch or an SK data statement) causes an attempt to read IU past end-of-file, the appropriate Keyboard Monitor I/O handler will assume control. In the case of paper tape, ↑P will be typed on the teleprinter to request that another paper tape be loaded. Note that this does not preclude more than one END statement in the IU file.

4.3 DIRECTIVE ERRORS

If a user makes an illegal reply to a HYSAT directive request, then one of the following error messages will be output to the teleprinter:

**ILLEGAL IU
**ILLEGAL OU
**ILLEGAL CONTINUATION DIRECTIVE
**IU AND OU CANNOT BOTH BE THE DCT-2ØØØ

Error recovery is automatically performed by HYSAT (the directive request is repeated).

4.4 INPUT ERRORS

The error messages in this group relate to illegal specifications on input data statements from IU. They are distinguished by an "**" in columns 4 and 5 of the message. Columns 1 and 2 of these messages contains "CO" to facilitate the use of status readout mode (OU will never contain non-data statement records unless it is the teleprinter).

All error messages in this group are output to both OU (unless OU is the teleprinter) and the teleprinter. A copy of the first 40 columns of the data statement in error immediately follows the error message on OU (with "CO" in columns 1 and 2 and only if OU is not the teleprinter), and the teleprinter (unless the teletype keyboard is the current input unit). HYSAT then outputs:

CORRECT STATEMENT(S)-

$

on the teleprinter. HYSAT then proceeds as though the keyboard is the input unit until a blank statement or a skip statement is encountered, at which time the user-specified IU once again becomes the current input unit. This "temporary teletype input" procedure enables the user to correct the data statement specification error.

EXAMPLE 26:

teletype log	comments
CO ** INCONSISTENT REPETITION SPECIFICATION	error message
CS 2+ 1 1	copy of data statement in error (2 > 1)
CORRECT STATEMENT(S)-	input error recovery directive
$	keyboard becomes input
CS ØØ+ 1 1	corrected statement
	blank statement (IU becomes input unit)

The following are data statement input errors:

**UNRECOGNIZABLE FIELD
**ILLEGAL COMMAND CODE
**ILLEGAL FIELD
**INCONSISTENT REPETITION SPECIFICATION
**INVALID ARGUMENT IN M FIELD
**INVALID ARGUMENT IN N FIELD

4.5 HARDWARE ERRORS

The action taken upon detection of these errors depends upon the error control option selected for the data statement in error (see Section 3.2).

If the error control option field for this data statement is blank, HYSAT will ignore the error and continue with consecutive data statement execution. If the error control option field is E or S, HYSAT will respectively skip or not skip the number of data statements specified by the S field after completing execution of this data statement. If the error control option field is H, HYSAT will attempt error recovery (below). In any case, all error messages in this group are output to OU (if the error messages option is YES) with "CO" in columns 1 and 2 to facilitate the use of status readout mode (OU will never contain non-data statement records unless it is the teleprinter).

4.5.1 Recovery

If the error control option field is H for a data statement which caused a hardware error, then data statement execution is immediately paused and HYSAT enters the error recovery routine described below.

The appropriate error message is output on the teleprinter (unless OU is the teleprinter) followed by the hardware error recovery directive:

TYPE R (RETRY), C (CONTINUE), OR NEW STATEMENT(S)-

$

If the user responds to the directive with an R (in column 1), HYSAT will attempt to repeat the operation which caused the hardware error. If the user responds with a C (in column 1), HYSAT will ignore the hardware error and continue with data statement execution from the point at which it was paused. If the user wishes to alter the data statement which caused the hardware error, or to investigate the error condition in more detail, he may respond with new data statements: HYSAT will proceed as though the teletype keyboard is the input unit until a blank statement or a skip statement is encountered, at which time, the user-specified IU once again becomes the current input unit.

EXAMPLE 27: (assume the error tolerance is 5 millivolts and the data statement is: A5H)

teletype log	comments
CO COMPONENT A 5 WRONG BY -6 MILLIVOLTS TYPE R (RETRY), C (CONTINUE), OR NEW STATEMENT(S)-	error message hardware error recovery directive
$	keyboard becomes input unit
ER1Ø	increase error tolerance to 10 millivolts
A5H	repeat the erroneous data statement
	blank statement (IU becomes input unit)

4.5.2 Messages

The following are hardware error messages:

```
ANALOG MODE    FAILED TO SET PROPERLY
TIME CONSTANT FAILED TO SET PROPERLY
LOGIC MODE     FAILED TO SET PROPERLY
COMPARATOR              nn IN WRONG STATE
SENSE LINE               n IN WRONG STATE
GENERAL PURPOSE INTERRUPT  n IN WRONG STATE
GENERAL PURPOSE INTERRUPT  n FAILED TO OCCUR
GENERAL PURPOSE INTERRUPT  n OCCURRED
COMPONENT XX nnn WRONG BY ±mmmmm MILLIVOLTS
COMPONENT XX nnn GAVE ERROR NO. n
```

The error number in the last error message above corresponds to the values of IERROR in the hybrid linkage routines RAVA, SPOT, or CRAC, as listed in Table 2.

TABLE 2. Linkage Routines Error Codes.

Linkage Routine	IERROR	Meaning
RAVA	2	hardware malfunction (type and/or address failed to set properly)
	4	hardware malfunction (analog address or DVM busy too long)
	5	DVM overrange (DVM > 1.1799)
	7	keyboard interference
SPOT	2	hardware malfunction (SP mode or pot address failed to set properly)
	4	hardware malfunction (analog address, analog value, pot set, or DVM busy too long)
	5	pot failed to set within tolerance
	6	pot null failure
	7	keyboard interference
CRAC	2	ADC overrange
	4	hardware malfunction (conversion took too long)

5.0 ACCUMULATOR SWITCH OPTIONS

The switch options described in this section are intended to provide additional control facilities for the sophisticated hybrid user and/or maintenance engineer. If all accumulator switches are left in the DOWN (Ø) position during use of HYSAT, then this section can be disregarded. Accumulator switch options are summarized in Table 2.

5.1 DIRECTIVE ABBREVIATION

While accumulator switch 0 is in the UP position, all HYSAT directives and error recovery directives (input or hardware errors) will be abbreviated. This is a useful time-saving option for experienced HYSAT users.

5.2 CONTINUOUS RETRY

While accumulator switch 1 is in the UP position, HYSAT will continuously repeat any operations which cause hardware errors without operator intervention (the hardware error recovery routine is bypassed). This switch can be used in place of a typed response to the hardware error recovery directive. The erroneous operation will be repeated automatically until it executes successfully or until switch 1 is returned to the DOWN position.

5.3 STATUS READOUT MODE

While accumulator switch 2 or 3 is in the UP position, HYSAT's internal status readout mode switch (see 3.10) is overriden. Accumulator switch 2 UP forces status readout mode to be ON; switch 3 UP forces status readout mode to be OFF. Switch 2 takes precedence over switch 3. Status readout mode is in the last program-specified state if both switch 2 and switch 3 are DOWN.

5.4 ERROR MESSAGE OPTION

While accumulator switch 4 or 5 is in the UP position, HYSAT's internal error message option switch (see 3.14) is overriden. Accumulator switch 4 UP forces the error message option to YES; switch 5 UP forces the error

message option to NO. Switch 4 takes precedence over switch 5. The error message option is in the last program-specified state if both switch 4 and switch 5 are DOWN.

5.5 HARDWARE ERROR PAUSE

While accumulator switch 6 is in the UP position, HYSAT will automatically pause data statement execution and transfer control to the hardware error recovery routine whenever a hardware error occurs, regardless of the actual contents of the E field on the data statement (E fields of E, S, or blank are overridden).

5.6 ABORTS

Accumulator switches 7, 8, 9, and 10 will cause data statement execution to be aborted immediately prior to reading the next data statement from the current input unit if one or more of the switches is in the UP position. Data statement execution cannot be aborted while in an error recovery routine (input or hardware). Switch 7 has the highest priority, switch 10 the lowest. Accumulator switch 7 UP terminates HYSAT (TERMINATE); switch 8 UP simulates a RESTART (see 2.5.3); switch 9 UP simulates a CHANGE IU/OU (see 2.5.4); switch 10 UP simulates an END data statement (see 3.18). IU and OU are CLOSEd on an abort if they are DECtape.

5.7 TELETYPE INPUT

While accumulator switch 11 is in the UP position, the teletype keyboard is the current input unit. The current input unit reverts to the program-specified device (IU or temporary teletype input) when switch 11 is returned to the DOWN position.

TABLE 3. Accumulator Switch Options

Switch	Effect
0	abbreviates HYSAT directives
1	automatically repeats operations which cause hardware errors
2	forces status readout mode ON
3	forces status readout mode OFF
4	forces error control option to YES
5	forces error control option to NO
6	transfers control to hardware error recovery routine on occurrence of hardware errors
7	aborts data statement execution and TERMINATE
8	aborts data statement execution and RESTART
9	aborts data statement execution and CHANGE IU/OU
10	simulates END statement
11	assigns teletype keyboard as current input unit

The following seven sheets are an update to the LINKAGE (cherry red HYBRID) section of the hybrid lab user's manual.

TABLE OF CONTENTS

2.1.1 PROGRAM: Initialize

NAME: INIT

PURPOSE: 1. Select the 680 console.

2. Clear and mask off all general purpose and special purpose interrupt conditions.

3. Initialize control linkage for software level, general purpose, and fault interrupt handlers.

4. Reset all sense lines and output control lines.

5. Check function of CAF, SIMB, CSIS, CDIR, RCSR, TSLO-7, and RBCL.

6. Enable the API system.

AUTHOR: C. L. Cross

CALLING SEQUENCE: CALL INIT (IERROR)

ARGUMENTS: IERROR indicates errors as follows:
IERROR = 0 no error
= 2 hardware malfunction - console failed to select properly

EXECUTION TIME: 257 μsec

STORAGE REQUIREMENTS: 308 locations

COMMENTS: 1. INIT must be included in every program which uses any hybrid linkage routines and it must be called before any other hybrid linkage routine is called. Results are unpredictable if this restriction is violated.

2. If INIT is called more than once from any single Fortran IV program, the EAI 693 interface (and the API system) is effectively re-initialized each time, i.e., all interrupt conditions are cleared.

3. The value of the ADC multiplexer address, the DAC channel address, and the DAC and ADC buffers are undefined after execution of INIT.

4. Havoc results if a user attempts to load both INIT and INIT9 from one Fortran IV program. INIT9 should be used only for those programs which use the PDP-9 software API levels (with ISLI) but do not use the 693 interface; INIT must be used in all other cases.

4.0 LOGIC SIGNAL INTERFACE

The logic control interface provides for communication of binary control signals between the patchable logic section of the 680 and a digital program. The elements provided include 16 output control lines from the digital to the analog, and 8 sense lines and 8 general purpose interrupts from the analog to the digital.

4.1 CONTROL LINES

The patch panel terminations and the internal logic of the output control lines (OCL) are shown in figure 4. The outputs of the flip-flops which drive the patch panel points are clocked so that the signals may be used with no additional timing. The I/O bus of the PDP-9 is steered to set or reset one or more lines or to load the entire sixteen lines with a digital word. In additon, patch panel reset holes are provided for the first ten lines. These holes are normally low with nothing patched, so that the state of the OCL is exclusively under control of the digital computer program. If the RST hole for a particular line is brought high by patching in a logic "ONE," the output of the flip-flop will be a blip (level high for one clock time) each time the line is set. The RST hole may also be driven from any logic signal on the patch panel, so that the OCL may be tested at some point in a logic program and then reset.

All output control lines are reset after execution of INIT.

4.1.5 PROGRAM: Load Bank of Control Lines

NAME: LBCL

PURPOSE: 1. Load the specified values (high or low) into the output control lines.

2. Check function of LOCL.

AUTHOR: C. L. Cross

CALLING SEQUENCE: CALL LBCL (IPATTERN)

ARGUMENTS: Bits 0-15 of IPATTERN specify the values of output control lines 0-15:
if bit i of IPATTERN is "1", then output control line i is set (high);
if bit i of IPATTERN is "0", then output control line i is reset (low).
Bits 16 and 17 of IPATTERN have no effect.

EXECUTION TIME: 25 μsec

STORAGE REQUIRED: 14 locations

4.2 SENSE LINES

Eight sense line inputs originate at the 680 patch panel. A simplified logic configuration is shown in Figure 5.

When a sense line is tested individually, it is reset to acknowledge that the test has been performed. Thus, a single test (TRSL) of the sense line input will determine whether it has been or is still high since the last test. A double test (TRST) of the sense line will determine whether the sense line input is in the high (continuous level) state. It is also possible to read the eight sense lines in parallel as an eight-bit word. When the sense lines are read in parallel (RSSL or RBSL), the individual sense lines are not reset.

All sense lines are reset after execution of INIT.

Sense line inputs are clocked (the sense line flip flop goes high on the next clock pulse after the sense line input patch panel hole is brought high).

4.2.3 PROGRAM: Read Single Sense Line (Function)

NAME: RSSL

PURPOSE: 1. Read the state of the specified sense line.

2. Check function of RSLI.

AUTHOR: C. L. Cross

CALLING SEQUENCE: RSSL (ISENSE)

ARGUMENTS: ISENSE specifies the sense line 0-7.

RETURN: RSSL is .TRUE. if sense line ISENSE is high.

RSSL is .FALSE. if sense line ISENSE is low, or if an illegal sense line is requested.

EXECUTION TIME: 47 to 49 μsec

STORAGE REQUIRED: 27 locations

PROGRAMMING HINTS: RSSL must be declared LOGICAL in the FORTRAN IV calling program.

This writeup is intended as an addendum to the Digital Equipment Corporation PDP-9 FORTRAN IV manual (DEC-9A-AF4B-D). It contains descriptions of additional features and/or restrictions which are not fully described in the DEC manual. This addendum should be inserted by the user in his own DEC FORTRAN IV manual.

CARNEGIE-MELLON UNIVERSITY

HYBRID COMPUTATION LABORATORY

FORTRAN IV

Addendum

Series 69-1

January, 1969

TABLE OF CONTENTS

1. FILE MANIPULATION

The FORTRAN-callable subroutines described below provide the user with a means of communication with file-oriented DECtape. Note that, since a DECtape may be either file-oriented or non file-oriented but not both simultaneously, the BACKSPACE, REWIND, and ENDFILE commands should never be used on the same DECtape that the following subroutines are manipulating.

In the following subroutine descriptions, N is the device number of the DECtape to be manipulated, I is a BOOLEAN variable, and A and B represent the name of a two-element (REAL) array which contains a 9-character ASCII file-name and file-name extension.

1.1 SEEK (N,A)

SEEK is used to find and OPEN an input file A on DECtape N.

EXAMPLE 1:
```
DIMENSION FILENM (2)
DATA FILENM (1),FILENM(2)/5HINPUT,4HDATA/
CALL SEEK(1,FILENM)
```

result: The user file INPUTD ATA on DECtape 1 will be found and OPENed in preparation for input to the user's program.

1.2 ENTER (N,A)

ENTER is used to create and open an output file A on DECtape N.

EXAMPLE 2:
```
DIMENSION FILENM (2)
DATA FILENM(1),FILENM(2)/5HOUTPU,4HTANS/
CALL ENTER(2,FILENM)
```

result: A new file called OUTPUT ANS is created on DECtape 2 in preparation for output from the user's program.

1.3 CLOSE (N)

CLOSE terminates input to or output from the currently open file on DECtape N. CLOSE must be used if SEEK or ENTER has been used.

EXAMPLE 3: (assume example 2)
```
CALL CLOSE(2)
```

result: The output file OUTPUT ANS on DECtape 2 is CLOSED.

1.4 FSTAT(N,A,I)

FSTAT checks for the presence of file A on DECtape N. If the file is found, I is set to .TRUE.; if the file is not found, I is set to .FALSE.

EXAMPLE 4: (assume examples 2 and 3)
```
BOOLEAN I
CALL FSTAT (2,FILENM,I)
```

result: The directory of DECtape 2 is searched for the file OUTPUT ANS. Since this file was created via a CALL of ENTER and closed via a CALL of CLOSE, it will be present and I = .TRUE. upon exit from FSTAT.

1.5 RENAM(N,A,B,I)

RENAM checks for the presence of file A on DECtape N, and renames the file as B if it is found, I is set .TRUE. if the file was found, .FALSE. if the file was not found.

EXAMPLE 5: (assume OLDNAM SRC exists on DECtape 3)
```
DIMENSION OLDFIL(2)
REAL NEWFIL(2)
DATA OLDFIL(1),OLDFIL(2)/5HOLDNA,4HMSRC/
DATA NEWFIL(1),NEWFIL(2)/5HNEWNA,4HMSRC/
CALL RENAM (3,OLDFIL,NEWFIL,I)
```

result: File OLDNAM SRC on DECtape 3 will be renamed NEWNAM SRC and I = .TRUE. upon exit from RENAM.

1.6 DLETE(N,A,I)

DLETE checks for the presence of file A on DECtape N, and deletes the file if it is found. I = .TRUE. if the file was found, .FALSE. if not.

EXAMPLE 6: (assume example 5)
```
CALL DLETE (3,OLDFIL,I)
```

result: Since file OLDNAM SRC was renamed (in example 5), nothing is deleted and I = .FALSE. upon exit from DLETE.

2. ARRAY MANIPULATION

The following descriptions present three new FORTRAN-callable subroutines for adjustable array dimensioning, and a method of programming around the FORTRAN IV bug which prevents the use of array names as subprogram arguments.

2.1 ADJ1

Class: External Subroutine

Purpose: To provide dimension adjustment on a one dimension array.

Calling Sequence:
```
DIMENSION B(1)
CALL ADJ1  (B,A)
```

where B is the array whose storage begins at A. A must be an array element (such as C(200)) with sufficient storage beyond A to allow for all the entries of array B. The dimensions or type of array A do not have to agree with array B.

B cannot be a dummy argument in a subroutine but A can be a dummy argument.

EXAMPLE 7:
```
DIMENSION A(300), B(1), C(1)
CALL ADJ1 (B,A(101))
CALL ADJ1 (C,A(201))
```

After the calls to ADJ1, the arrays B and C may be referenced as if they had been dimensioned as (100) each. No further calls to ADJ1 have to be made.

Size: 17 octal locations

Error Conditions: None

2.2 ADJ2

Class: External Subroutine

Purpose: To provide dimension adjustment for a two dimension array

Calling Sequence:

```
DIMENSION B(1,1)
      ≷
CALL ADJ2 (B,A,NR)
```

where NR is the number of rows to appear in array B. See ADJ1 for comments on B and A.

EXAMPLE 8:

```
DIMENSION A(3ØØ), B(1,1), C(1,1)
      ≷
CALL ADJ2(B,A(1),1Ø)
CALL ADJ2(C,A(1Ø1),2Ø)
      ≷
```

After the calls to ADJ2, the arrays B and C may be referenced as if they had been dimensioned (10,30) and (20,10) respectively. No further calls to ADJ2 have to be made.

Size: 36 octal locations

Error Conditions: None

2.3 ADJ3

Class: External Subroutine

Purpose: To provide dimension adjustment for a three dimension array

Calling Sequence:

```
DIMENSION B(1,1,1)
      ≷
CALL ADJ3 (B,A,MR,NC)
```

Where NR and NC are the number of rows and columns respectively to appear in array B. See ADJ1 for comments on B and A.

Size: 41 octal locations

Error Conditions: None

2.4 Subprogram Arguments

The current FORTRAN IV compiler fails when a subprogram contains both a COMMON statement and a dimensioned array in the argument list. Example 9 presents a method of avoiding this compiler bug.

EXAMPLE 9: normal coding:

main program

```
DIMENSION A(200)
COMMON B
     ⋮
CALL SUB(A)
```

subroutine

```
SUBROUTINE SUB(A)
DIMENSION A(200)
COMMON B
```

result: The compiler fails.

corrected coding:

main program

```
DIMENSION A(200)
COMMON B
     ⋮
CALL SUB (A(1))
```

subroutine

```
SUBROUTINE SUB (AX)
DIMENSION A(1)
COMMON B
CALL ADJ1(A,AX)
```

result: The array A can now be referenced as it would be in the normal coding above if it were not for the compiler bug.

3. WARNINGS AND RESTRICTIONS

3.1 Function Type Declarations

Any function references (statement functions, intrinsic functions, or external functions/which are not implicitly REAL or INTEGER must appear in the appropriate type statement.

EXAMPLE 10:
```
DOUBLE PRECISION B,X,DABS,DATAN
     B = DATAN(DABS(X))
```

If DABS and DATAN were not declared DOUBLE PRECISION, improper code would be generated by the compiler.

3.2 Logical IF Statements

The FORTRAN IV compiler is easily confused on logical IF statements where the logical expression contains more than one logical variable. Also, a logical IF statement can only be continued on a new record if it is broken before the closing parenthesis.

EXAMPLE 11:
```
IF (X.NE.
1 Y) GO TO 2Ø        compiles correctly

 IF (X.NE.Y)
1 GO TO 2Ø           causes an error.
```

3.3 Dummy Arguments

The FORTRAN IV compiler fails when a dummy argument is a function name (or an array name if the subprogram contains a COMMON statement).

3.4 IFIX

When the IFIX function is used with numbers in the range $-(2^{17}-1)$ to $-(2^{16})$ the sign of the result is lost.

3.5 Re-entrancy

FORTRAN IV operating time system subroutines are not re-entrant. This may cause unpredictable results if the HANDLER for any interrupt condition involves real calculations, array addressing, implicit functions, etc.

This writeup is intended as an addendum to the Digital Equipment Corporation PDP-9 Editor manual (part of DEC-9A-GUAB-D, PDP-9 Utility Programs manual). New features of the PDP-9 Editor, which are not described in the DEC manual, are described in this writeup. The user should insert this writeup in his own DEC Utility Programs manual.

CARNEGIE-MELLON UNIVERSITY

HYBRID COMPUTATION LABORATORY

EDITOR

Addendum

Series 69-1

January, 1969

TABLE OF CONTENTS

1. NEW COMMANDS

1.1 OUTPUT

form: OUTPUT {ON / OFF}

OUTPUT mode is initially set to ON (when the Editor is first loaded into core). If the user gives an OUTPUT OFF command, then the Editor will no longer output any source input text. The user may thus examine any source file at will without the necessity of updating it. If the user gives a WRITE or CLOSE command while OUTPUT mode is OFF, the message "NOTHING IN FILE" will be typed and no other action results. An input source file can never be changed while OUTPUT mode is OFF.

1.2 RENEW

form: RENEW

The RENEW command acts just like a WRITE command followed by a READ command (in BLOCK mode).

2. ADDITIONAL FILE MANIPULATION COMMANDS

2.1 CALL Function

The CALL function allows user access to the file renaming and deletion facilities of the Editor. This command may not be issued while any file (either input or output) is open (i.e., it may only be used immediately after the Editor is loaded or restarted).

2.1.1 RENAME

form: CALL RENAME {INPUT / OUTPUT} OLDNAM EX1 NEWNAM (EX2)

The file OLDNAM EX1 on .DAT slot -14 (if INPUT is specified: normally DECtape unit 2) or .DAT slot -15 (if OUTPUT is specified: normally DECtape unit 3) is given the new name NEWNAM EX2. If EX2 is not specified, then the new file name will be named NEWNAM EX1.

2.1.2 DELETE

form: CALL DELETE {INPUT / OUTPUT} FILENM (EXT)

The file named FILENM EXT is deleted from the device specified (INPUT or OUTPUT, DECtape unit 2 or 3). If EXT is not specified, SRC is assumed.

2.2 KEEP

form: KEEP SAVNAM (EXT)

The KEEP function effects preservation of the <u>original</u> file as it appears on .DAT slot -14 (normally DECtape unit 2). That file will be renamed SAVNAM EXT during Editor processing of the next CLOSE or TOP request. If EXT is not specified, SRC is assumed. SAVNAM EXT must be unique in the directory of the DECtape (unit 2) assigned to .DAT slot -14.

If it is not, the name SAVFIL EDT will be assigned. The KEEP request must be issued while the input file is on .DAT slot -14 (i.e., the number of TOP commands given must be even: 0,2,4,...). As many KEEP requests as needed may be issued.

EXAMPLE 1:

EDITOR command	comments
EDITOR V4A	
> OPEN THSFIL	OPEN original input file
EDIT	
> KEEP SAVE1 SRC	save a copy of the original file
>	} normal editing commands
> TOP	normal TOP
> NEXT	locative request issued to allow physical file transfer at second TOP
> TOP	even number of TOPs-returns edited file to -14
> KEEP SAVE2 SRC	save a copy of the partly-edited file
>	} more normal editing commands
> CLOSE	save the new edited file

User typed commands are underlined in example 1. The resulting directory on .DAT slot -14 would include:

THSFIL SRC	new edited file
SAVE1 SRC	first copy - original unedited file
SAVE2 SRC	second copy - partly edited file

2.3 ICLOSE

form: ICLOSE

The ICLOSE command effects the closing of the current input file only. The output file remains open. A new input file may be referenced after the ICLOSE request by issuing an OPEN command. ICLOSE provides a facility for combining source files during an editing run.

EXAMPLE 2:
```
EDITOR V4A
> OPEN PART1
EDIT
> BOTTOM
> ICLOSE
> OPEN PART2
> NEXT
> CLOSE UNION
```

The above sequence of commands would create a new file UNION SRC, which would be a combination of the two files PART1 SRC and PART2 SRC.

2.4 SCLOSE

form: SCLOSE NEWNAM

The SCLOSE command is intended to allow the placement of the edited file on the current output device. Its use, in place of the CLOSE command, normally causes the edited SRC file to be left on .DAT slot -15. This allows the closing of long files, after trivial changes, without the tedious recopy process from .DAT slot -15 to .DAT slot -14 (DECtape unit 3 to DECtape unit 2). NEWNAM must be a different file name from that which was used with the OPEN command, or a fatal IOPS error will probably occur.

EXAMPLE 3:
```
EDITOR V4A
> OPEN LONGFL
EDIT
>  {
   {
> SCLOSE BIGFIL
```

The above sequence of commands will leave an edited version of LONGFL SRC called BIGFIL SRC on .DAT slot -15.

3. ADDITIONAL ERROR MESSAGES

3.1 NOTHING IN FILE

This error message results from issuing the CLOSE command when the output file is empty (the entire input file has been deleted, there was no input file to start with and none was created, or OUTPUT mode is OFF). In any case, the input file is unchanged if this error message is typed.

3.2 Identical File Names

If the Editor finds a file of the same name as the file being edited on .DAT slot -14 or .DAT slot -15 while processing a TOP or CLOSE command, the following action(s) result:

(1.) The Editor first attempts to leave the edited file on .DAT slot -14 (DECtape unit 2) as FILENM EXT (the user-specified name).

(2.) If it cannot do (1); i.e., another FILENM EXT already exists in the directory on .DAT slot -14, then it tries to leave the file on .DAT slot -14 (DECtape unit 2) as .TFIL1 EDT.

(3.) If it cannot do (2) either; i.e., .TFIL1 EDT already exists in the directory on .DAT slot -14, then it leaves the file on .DAT slot -15 (DECtape unit 3) as .TFIL1 EDT.

(4.) The Editor is restarted.

Also, the following message is typed:

FILE FILENM EXT IS PRESENT ON OUTPUT DEVICE.

YOUR EDITED FILE IS ON .DAT $\left\{\begin{matrix}-14\\-15\end{matrix}\right\}$ AS $\left\{\begin{matrix}.\text{TFIL1 EDT}\\\text{FILENM EXT}\end{matrix}\right\}$.

ORIGINAL FILE DELETED.

The user thus knows exactly the location and name of the edited file, and may take the appropriate action accordingly. This scheme protects against files of the same name on both .DAT slot -14 and -15, and against two or more files of the same name on .DAT slot -14.

3.3 .TFIL1

If the Editor's temporary file (.TFIL1 EDT) is found on the output device, (.DAT slot -15) the following message is typed:

FILE .TFIL1 IS PRESENT ON OUTPUT DEVICE. PLEASE
RENAME IT OR IT WILL BE DELETED.

If the file is not explicitly deleted or renamed (via the CALL function), it will be automatically deleted.

This writeup is intended as an addendum to the Digital Equipment Corporation PDP-9 PIP (Peripheral Interchange Program) manual (part of DEC-9A-GUAB-D PDP-9 Utility Programs manual). It contains descriptions of new and/or additional facilities in PIP, and should be inserted by the user in his own DEC Utility Programs manual.

CARNEGIE-MELLON UNIVERSITY

HYBRID COMPUTATION LABORATORY

PIP

Addendum

Series 69-1

January, 1969

TABLE OF CONTENTS

1. ABBREVIATIONS

This section will list command abbreviations and variations which are acceptable to PIP and increase its flexibility and ease of use.

1.1 File-Name Extensions/Data Modes

The file-name extensions SRC and BIN may be omitted in the Transfer (T) or Verify (V) commands if the appropriate data mode switch (A or B) appears in the command string.

EXAMPLE 1: command: T DT2 (A)←DT1 FILENM

result: FILENM SRC is transferred from DECtape unit 1 to DECtape unit 2. (If FILENM (no extension) exists on DECtape unit 1, it will be transferred.)

EXAMPLE 2: command: V DT1 FILENM (B)

result: FILENM BIN on DECtape unit 1 is verified.

Conversely, if SRC or BIN appears as an extension in the command string (T or V), data mode A or B, respectively, will be assumed.

EXAMPLE 3: command: T DT2 (A)←DT1 FILENM

result: same as T DT2 (A)←DT1 FILENM SRC

1.2 Device Designation

I/O device designations in the PIP command string may include a third letter.

EXAMPLE 4: command: L TTA←DT1

result: The letter "A" in the teleprinter designation "TTA" is acceptable to PIP (and ignored).

1.3 Use of Positive .DAT Slots

1.3.1 DECtape Unit Numbers

PIP no longer requires the exact unit number of a desired DECtape unit to be in the positive .DAT table. Provided that a DECtape handler is assigned to the positive .DAT table, PIP will automatically have the ability to reference any DECtape unit, 0-7.

EXAMPLE 5: (assume normal positive .DAT assignments: DTA1-1, DTA2-2, DTA3-3)

command: T DT6←DT7 FILENM SRC

result: PIP will perform the transfer correctly, even though DECtape units 6 and 7 are not actually assigned to the positive .DAT table.

1.3.2 DECtape Copy

The internal logic of the Copy (C) command with H switch specified; i.e., a direct DECtape copy in its entirety, has been changed to use all of available core memory for a very much faster copy (approximately 3.5 minutes in 16K). The fastest possible copy is achieved by assigning only the DECtape handler necessary for the copy to the positive .DAT table before loading PIP. DECtape handler DTD is strongly recommended for H mode copies.

EXAMPLE 6: commands:
```
MONITOR V4B
$ASSIGN DTD 1,2,3,4,5,6,7,10
$PIP
PIP V7A
>C DT1←DT2(H)
```

result: DECtape 2 is copied onto DECtape 1 at the fastest attainable speed.

1.3.3 Teletype

PIP now uses .DAT slot -3 (always TTA) when referencing the teletype, thus obviating the necessity of a TTA assignment in the positive .DAT table.

1.3.4 PIP Loading Time

PIP loading time may be reduced and its core utilization increased by reducing the number of I/O handlers required. This is accomplished by assigning "NONE" (or redundant handlers) to all positive .DAT slots which are not needed (the standard associated I/O device will not be used in the forthcoming PIP session).

2. NEW SWITCHES

2.1 N/S

S is now a permissible switch (the only one) in a Newdir (N) command. The S switch effects the reservation of a ↑Q area in addition to the directory and file bit map blocks reserved by the N command.

EXAMPLE 7: command: N DT2

result: The directory on DECtape 2 is cleared and directory and file bit map blocks (10 blocks) are reserved.

EXAMPLE 8: command: N DT2 (S)*

result: The directory on DECtape 2 is cleared and directory, file bit map, and ↑Q area blocks (110 blocks) are reserved.

The S switch remains permissible wherever else it is specified in the DEC PIP manual, but it no longer copies a system tape. S and N, however, are mutually exclusive when they are both used as switches.

2.2 T

The T switch effects the deletion of trailing spaces in an ASCII file. T, E, and C are mutually exclusive switches.

2.3 F

The F switch causes the insertion of a form feed (FF) and carriage return (CR) after every 56 (decimal) lines if no FF has thus far been encountered.

---

* The user is again reminded that the S switch must be used when initializing a DECtape which will subsequently be used for saving ↑Q core images.

2.4 Q

The Q switch effects the deletion of ID/sequence numbers from ASCII input lines during a Transfer (T) command. The switch is primarily intended for converting punched cards to some other medium. Columns 73-80 are deleted leaving a 72 character line with a carriage return as the 73rd character. The Q switch is legal with an A data mode in a Transfer command. It may be combined with the C (spaces to tabs) or T (delete trailing spaces) switches.