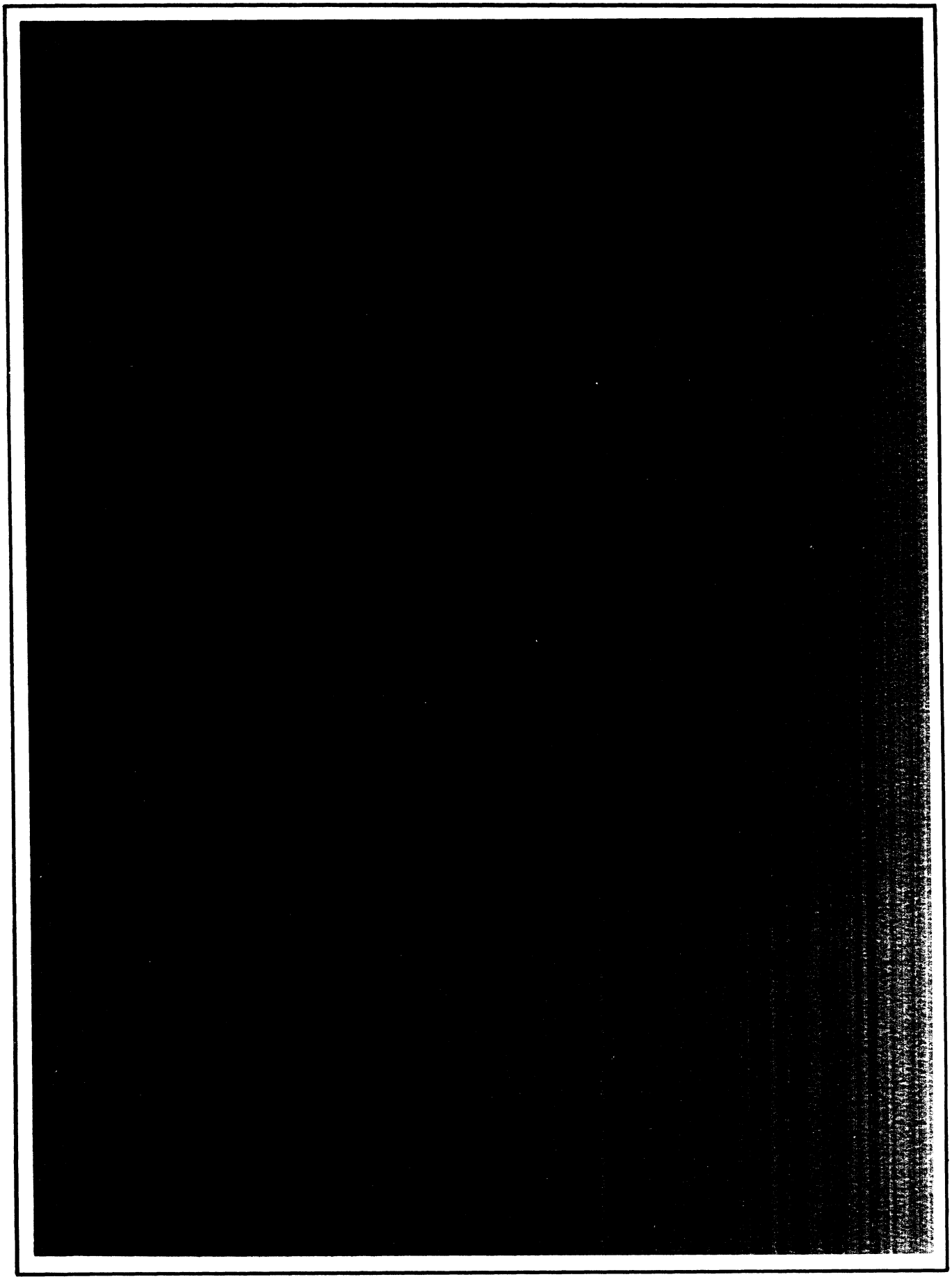
DEC 08-NGCA-D



DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

DEC 08-NGCA-D

PDP-8
CONSOLE
MANUAL

DECEMBER 1966

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

Reprinted

March 1967
October 1967

CONTENTS

READIN (RIM) LOADER

DEC Library Tape No: Digital-8-1-U

The PDP-8® Readin Mode Loader Program Write-Up, Digital-8-1-U, contains a complete discussion of the RIM Loader.

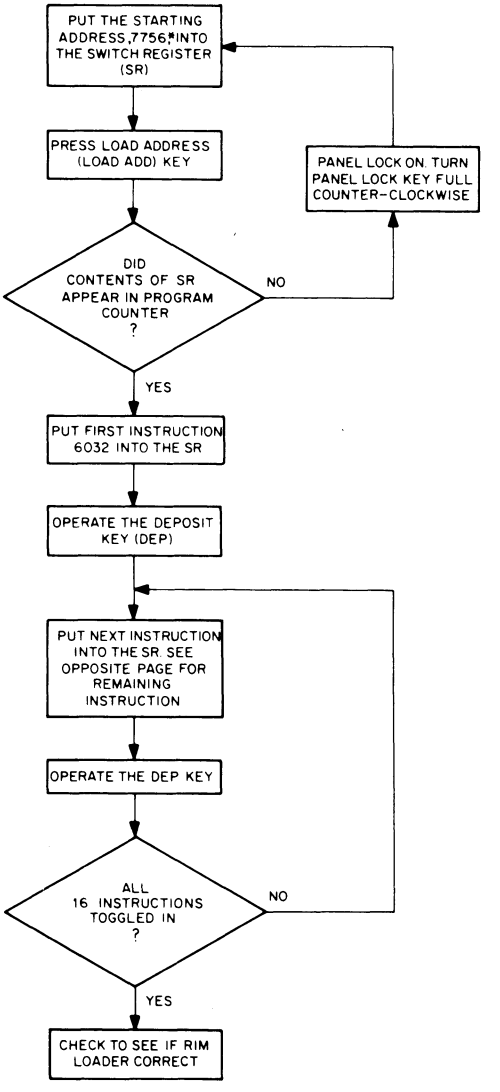
The RIM LOADER is a program used to load a program on a RIM format tape from the ASR 33 Reader into the PDP-8 memory. The RIM Loader must be toggled into memory initially from the console of the computer but should never have to be toggled in again unless destroyed by errors in the user's program (see figure 1).

From the programmer's viewpoint, the principal use of the RIM Loader is to load the Binary (BIN) Loader into memory (see page 4). However, since some diagnostic program tapes (Maindecs) are in RIM format, the RIM Loader is used to load them when checking the performance of the system.

Figure 2 gives in flow chart form instructions for checking the RIM Loader.

When using the high speed reader, the following RIM Loader program should be toggled into memory.

Address	Instruction (Octal)
7756	6014
7757	6011
7760	5357
7761	6016
7762	7106
7763	7006
7764	7510
7765	5374
7766	7006
7767	6011
7770	5367
7771	6016
7772	7420
7773	3776
7774	3376
7775	5357



* SWITCH DOWN = 0 LIGHT OFF
SWITCH UP = 1 LIGHT ON

Figure 1 Toggling in the RIM Loader

®PDP is a registered trademark of the Digital Equipment Corporation.

RIM LOADER

TABLE 1 RIM LOADER PROGRAM (ASR 33)

Address	Instruction (Octal)	Symbolic Program		
7756	6032	BEG,	KCC	/Clear AC and flag
7757	6031		KSF	/Skip if flag = 1
7760	5357		JMP .-1	/Looking for character
7761	6036		KRB	/Read buffer
7762	7106		CLL RTL	/Rotate channel
7763	7006		RTL	/8 into AC0
7764	7510		SPA	/Checking for leader
7765	5357		JMP BEG+1	/Found leader
7766	7006		RTL	/Channel 7 into link
7767	6031		KSF	/Skip if flag = 2
7770	5367		JMP .-1	/Looking for character
7771	6034		KRS	/Read, do not clear
7772	7420		SNL	/Checking for address
7773	3776		DCA I TEMP	/Store contents
7774	3376		DCA TEMP	/Store address
7775	5356		JMP BEG	/Return to get next character
7776	0000	TEMP,	0	/Temporary storage

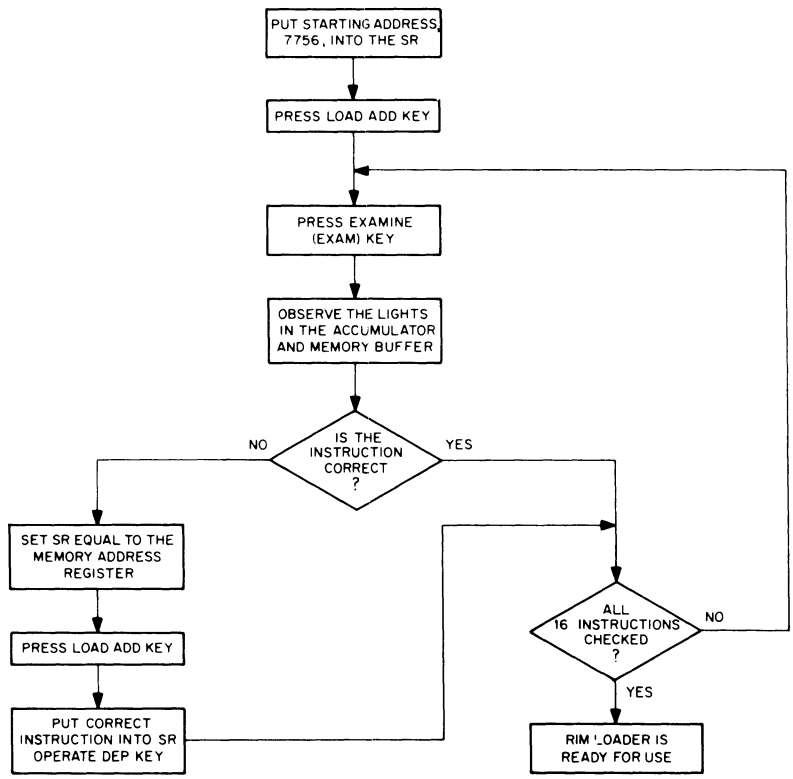


Figure 2 Checking the RIM Loader

BINARY (BIN) LOADER

DEC Library Tape No: Digital-8-2-U

The PDP-8 Binary Loader Program Write-Up, Digital-8-2-U, contains a complete discussion of the BIN Loader. Figure 3 gives instructions in flow chart form for loading the Binary Loader.

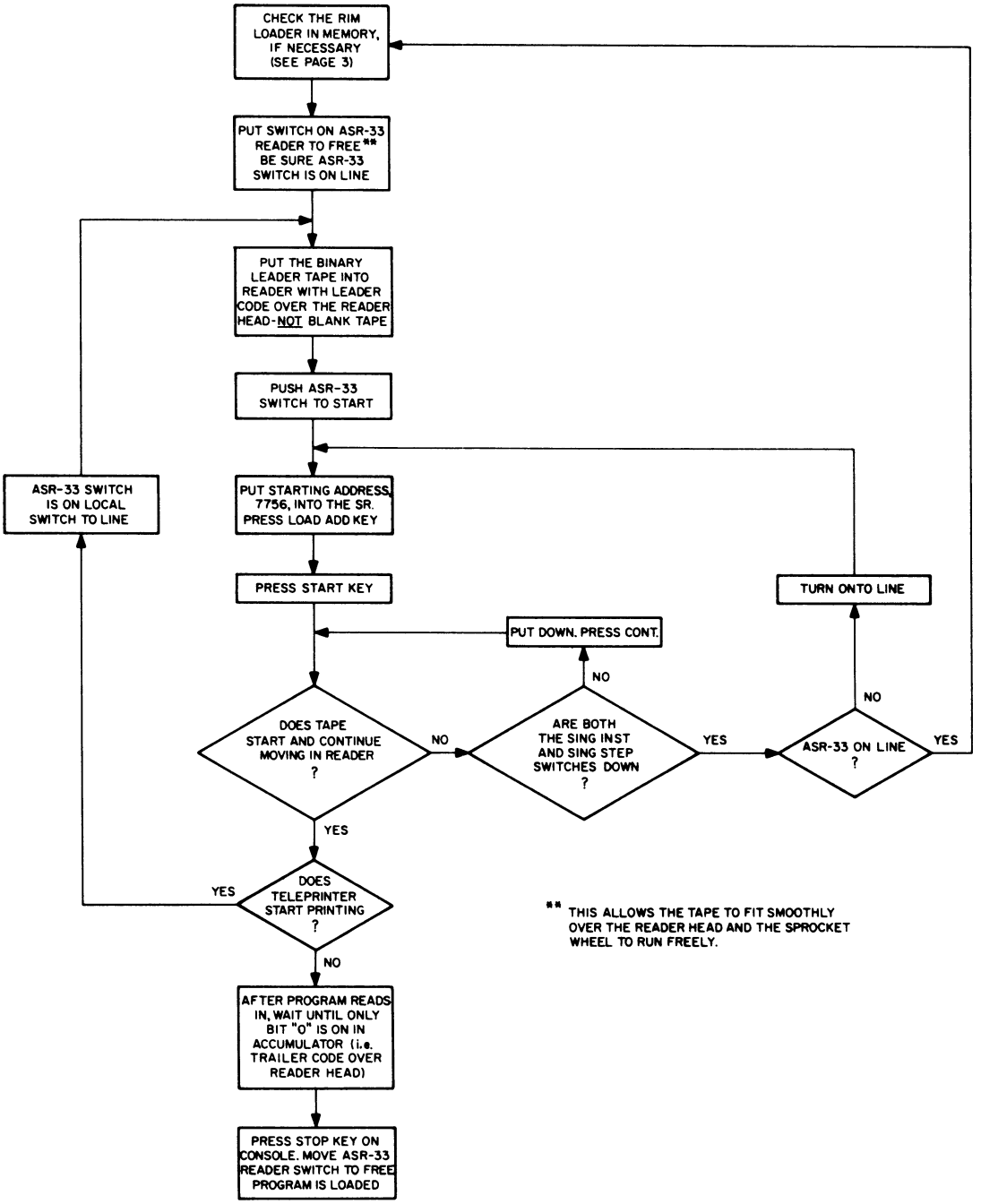


Figure 3 Loading the Binary Loader

BINARY LOADER

The BIN Loader is a program used to load a program on a binary format tape* from either the ASR 33 Reader (10 cps) or the high-speed photoelectric reader (300 cps) into the PDP-8 memory. The RIM Loader should load the BIN Loader into memory (see page 2). The BIN Loader never has to be loaded again unless destroyed by errors in the user's program. None of the binary programs supplied with the machine use the locations occupied by the Binary Loader. Only the user's program can disturb these locations.

From the programmer's viewpoint, the BIN Loader program loads:

Symbolic Tape Editor
PAL III
MACRO-8
DDT-8
FORTRAN Compiler and Operating System
Any tape from pass 2 of PAL III or MACRO-8
Any other tape in binary format (e.g., Floating-Point Packages)

The starting address of the BIN Loader is always 7777. After any given program is loaded correctly into memory, the user sets the starting address of the program into the SWITCH REGISTER (SR). This starting address is on the label of all system programs supplied with the PDP-8. The user should mark clearly the starting address on the binary tape of every program produced by pass 2 of the assembler.

In general, start the given program which was just loaded in the following manner:

1. Place starting address (from tape label) into SR.
2. Press LOAD ADD key.
3. Press START key.

Figure 4 illustrates how to use the BIN Loader. For the specific starting procedure for each program, the user should refer to the appropriate program write-up.

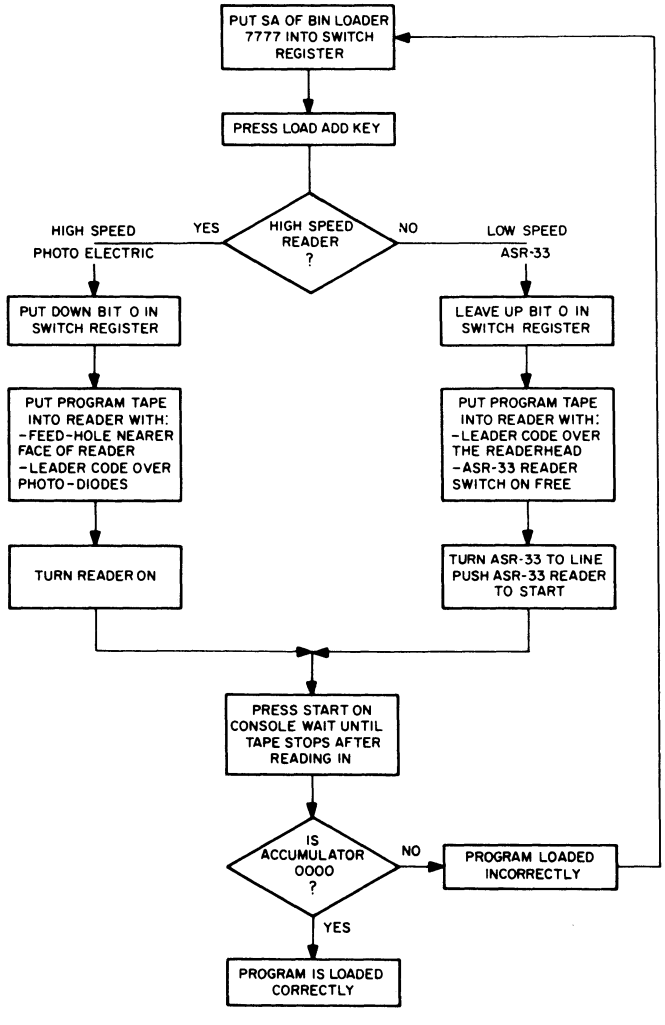


Figure 4 Using the Binary Loader

*All systems programs (except Maindecs), supplied with the PDP-8, which are in binary format have blue labels. The tapes with red labels are symbolic programs in ASCII code.

SYMBOLIC TAPE EDITOR

DEC Library Tape No: Digital-8-1-S ASR 33
Digital-8-1-S 75/750 (for high speed paper tape equipment)

OPERATING PROCEDURES

The reader must have a thorough knowledge of the PDP-8 Symbolic Tape Editor Programming Manual (Digital-8-1-S) before using the operating procedures outlined here. To generate and edit a program refer to figure 6. To punch a correct program see figure 7.

Modes

Command Mode - In this mode all characters typed on the keyboard of the ASR 33 are interpreted as commands to the Editor. These commands must be in one of the three following forms:

X)
nX)
n,mX)

where "X" represents a general command; n and m represent decimal numbers specifying lines in the symbolic program; and) represents the operator typing the RETURN key.

Command mode is the initial mode of the Editor.

Text Mode - This mode can only be entered by an appropriate command. In this mode, the Editor enters as text all characters typed on the ASR 33 keyboard or read from the reader in accord with the immediately preceding command.

The commands marked with an asterisk below leave the Editor in text mode. To return to command mode, the operator must type the CTRL/FORM combination. If the command R) has been given, the form feed character on the tape (placed there by the F) command during punchout) performs these functions. In either case, the return to command mode is accompanied by the ringing of the ASR 33 bell.

The unmarked commands neither remain in text mode nor ring the bell.

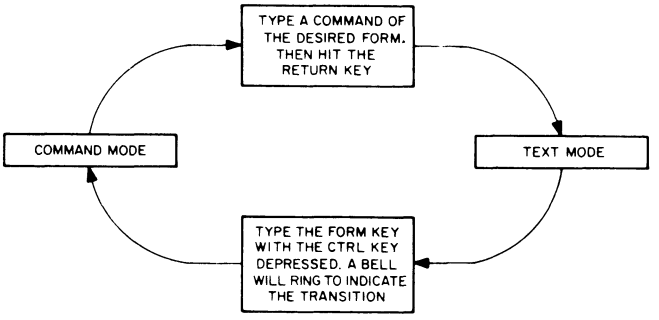


Figure 5 Transition Between Modes

Errors in Operation

If the operator types an illegal command or meaningless argument(s), the Editor responds with "?." Retype the command correctly.

If the operator types a command in command mode and the computer does not respond as expected, the operator probably forgot to hit the RETURN key.

If the operator types a command in text mode, that command is treated as text. The operator must signal the Editor by typing CTRL/FORM that he wishes to return to command mode.

Restart

If the operator stops the computer for some reason, he may restart it at location 0177 without disturbing the text in the buffer. A carriage return, line feed combination is generated. This is the usual restart procedure. If no carriage return, line feed is desired, restart at 0200.

CAUTION

If the Editor is restarted at 0176, all the text currently in the buffer will be lost.

SYMBOLIC TAPE EDITOR

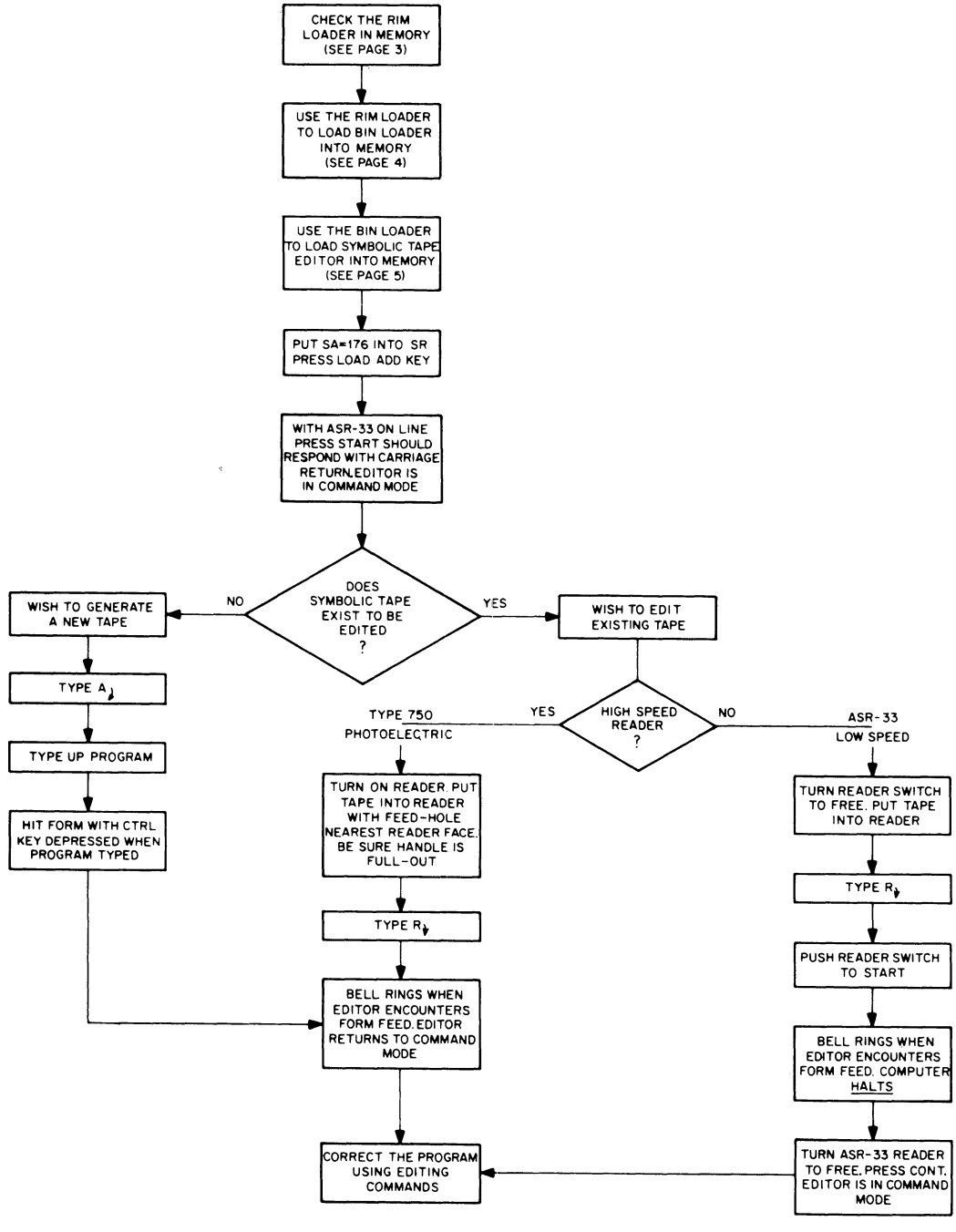


Figure 6 Generating and Editing a Program

SYMBOLIC TAPE EDITOR

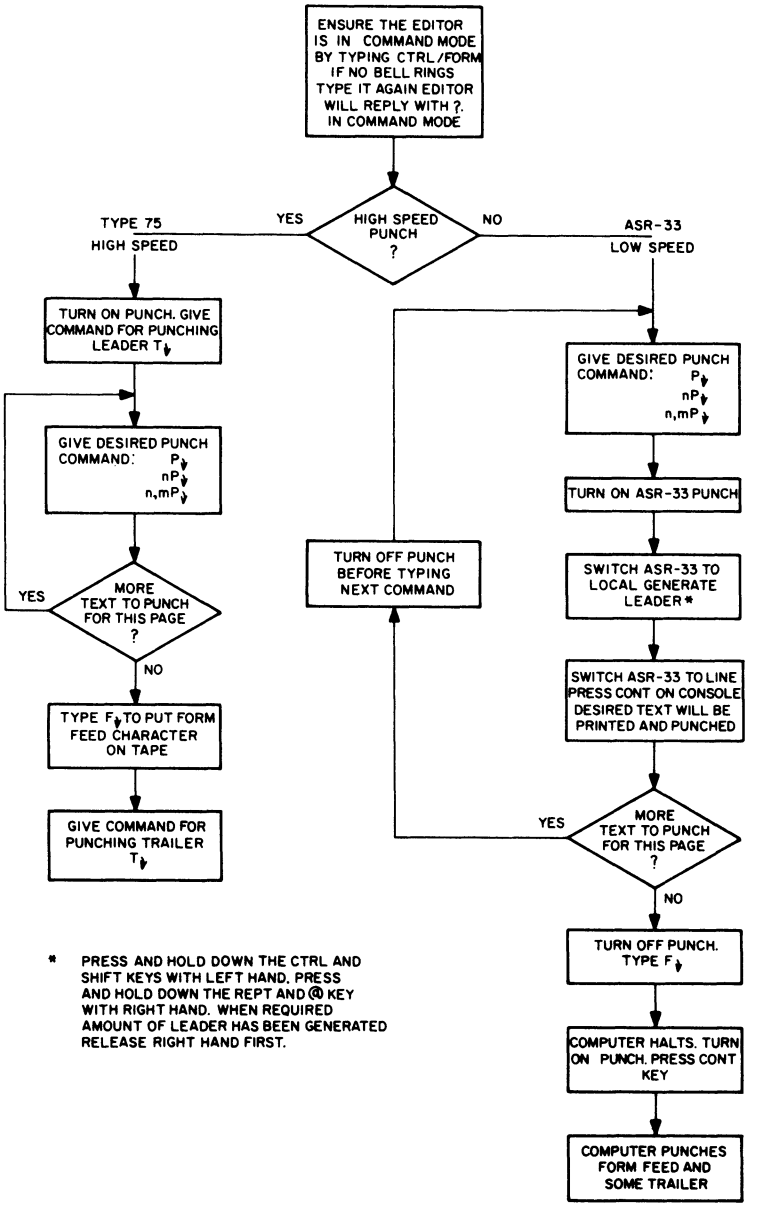


Figure 7 Punching Out a Correct Program

SYMBOLIC TAPE EDITOR

TABLE 2 SUMMARY OF EDITOR COMMANDS
(See Appendix 1 of PDP-8 Symbolic Tape Editor Programming Manual)

Command	No. of Arguments	Function
	Input	
R) *	0	Read incoming text from reader. Append to text buffer.
A) *	0	Append incoming text from keyboard to text buffer.
	Editing	
L)	0	List entire text buffer.
nL)	1	List line n.
n,mL)	2	List lines n through m.
nC) *	1	Change line n.
n,mC) *	2	Change lines n through m.
nI) *	1	Insert above line n.
nD	1	Delete line n.
n,mD)	2	Delete lines n through m.
K)	0	Delete entire text buffer.
	Output	
P)	0	Punch entire text buffer.
nP)	1	Punch line n.
n,mP)	2	Punch lines n through m.
F)	0	Punch form feed.
nN)	1	Equivalent to P), F), K), R) repeated n times.

*These commands leave the Editor in text mode until the operator types CTRL/FORM (in the 75/750 high-speed equipment version until the CTRL/FORM character is encountered on the symbolic tape).

TABLE 3 SUMMARY OF SPECIAL CHARACTERS FOR THE EDITOR
(See appendix 1 of PDP-8 Symbolic Tape Editor Programming Manual)

Character	Function
	Key Functions
Back arrow (←)	Cancels entire line in either text or command mode.
CTRL/TAB	On input, acts as a tab of ten spaces. May be suppressed on output; put up SR bit 1.
	Special Arguments (In Command Mode Only)
Virgule (/)	Decimal value of last line of current page.
Period (.)	Decimal value of current line being processed.
Equals (=)	Decimal value of preceding argument is typed out.
Space ( )	Allows line specification relative to current line (.).
Minus (-)	Allows line specification relative to current line (.) or last line (/)
	In 75/750 Version Only
ALT MODE	List line following current line (.).
<	List line immediately preceding current line (.).

PAL III ASSEMBLER

PAL III SYMBOLIC ASSEMBLER

DEC Library Tape No: Digital-8-3-S

The PDP-8 PAL III Symbolic Assembler Programming Manual presents a complete discussion of the PAL III Program. Specifically, chapter 5 of the manual contains the operating instructions. The flow diagram on the opposite page shows only one of the possible operating procedures for a complete assembly. For other switch settings, refer to chapter 5 of the Manual.

At the beginning of pass 1, PAL III senses by program whether the system has a Type 750 High Speed Photoelectric reader. If so, the symbolic tape must be assembled using this high speed reader.

If using the ASR-33 during pass 2, random characters are printed on the page which have absolutely no relevance to the programmer they should be ignored. However, diagnostic messages are printed during pass 2 which are important to the programmer. These may be easily recognized since the typing mechanism returns to the left hand side of the page before typing a message of the following format:

AB XXXXXX AT YYYY

where A and B represent two alphabet characters, and XXXXXX and YYYY represent numbers or characters depending on the diagnostic.

For a summary of the diagnostics that may occur during assembly, the reader is referred to pages 5-2 and 5-3 of the PAL III Manual (Digital-8-3-S) or, for a summary, appendix 2 of this manual.

If the program is to be debugged with DDT-8, the symbol table punched on the tape at the end of pass 3 should be saved on the same tape as the output of pass 2 and loaded at debugging time. To do this, turn on the punch at the end of pass 3 immediately after the "$" is encountered on the tape while the teleprinter is still operating (i.e., the computer does not halt).

PAL III ASSEMBLER

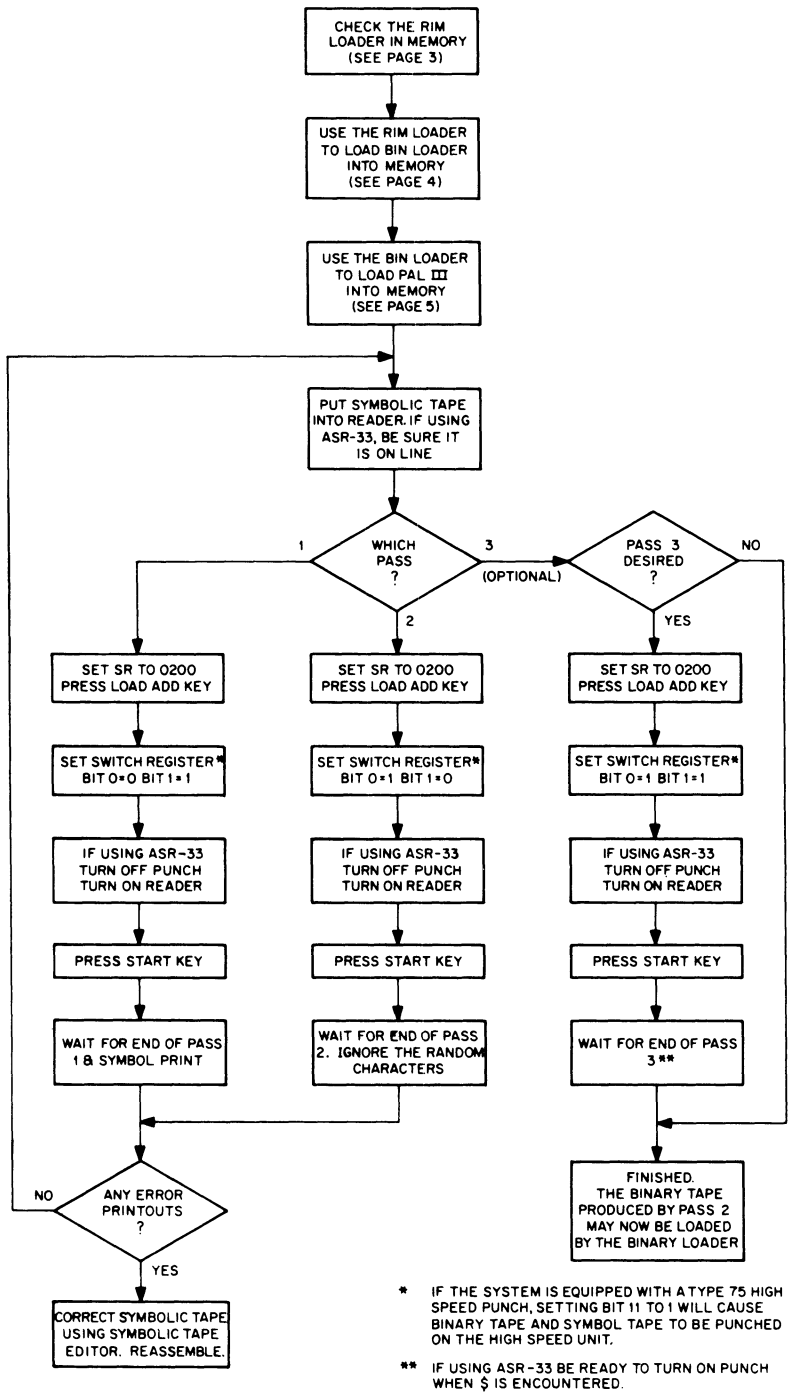


Figure 8 PAL III Flow Diagram

MACRO-8 ASSEMBLER

DEC Library Tape No: DEC-08-CMAA-D

The PDP-8 MACRO-8 Assembler Manual contains a complete discussion of the MACRO-8 Assembler. It is suggested that the beginning programmer thoroughly understand the features of PAL III before attempting to use MACRO-8. The additional features of MACRO-8 are found on page 5-1 of the manual.

The flow diagram on the opposite page is oriented toward the ASR 33 version of the Assembler and shows only one of the possible operating procedures for a complete assembly. For other operating alternatives and SWITCH REGISTER settings, the reader is referred to chapter 9 of the manual.

As in PAL III, printed on the page are random characters which have absolutely no relevance to the programmer and should be ignored. However, diagnostics messages are printed during pass 2 which are important to the programmer. These may be easily recognized since the typing mechanism returns to the left hand side of the page before typing a message of the following format:

AB ADDRESS

where AB represents two alphabet characters and ADDRESS represents the address where the error occurred relative to the latest tag. For a summary of the diagnostics that may occur during assembly, the reader is referred to chapter 8 of the MACRO-8 manual, or for summary, appendix 2 of this manual.

If the program is to be debugged with DDT-8, the symbol table punched at the end of pass 2 should be saved on the same tape as the binary object program, and loaded at debugging time.

Active Register Storage

[A [Y are registers within DDT which contain the contents of the AC and state of the link respectively at the time the breakpoint is encountered. These may be opened, modified and closed exactly like any general register, k, in the user's program.

MACRO-8 ASSEMBLER

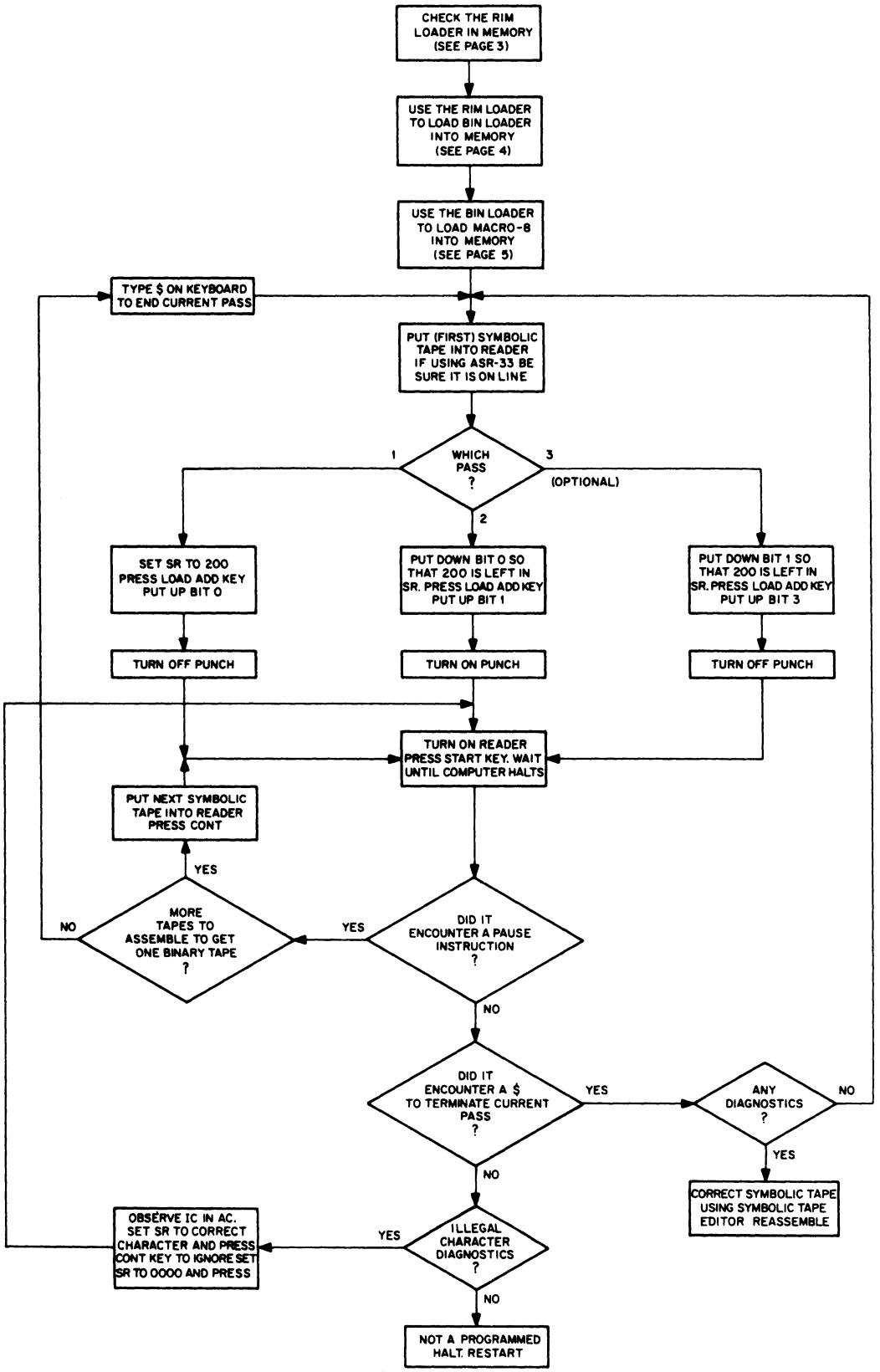


Figure 9 Macro-8 Flow Diagram

DDT-8

DEC Library Tape No: Digital-8-4-S

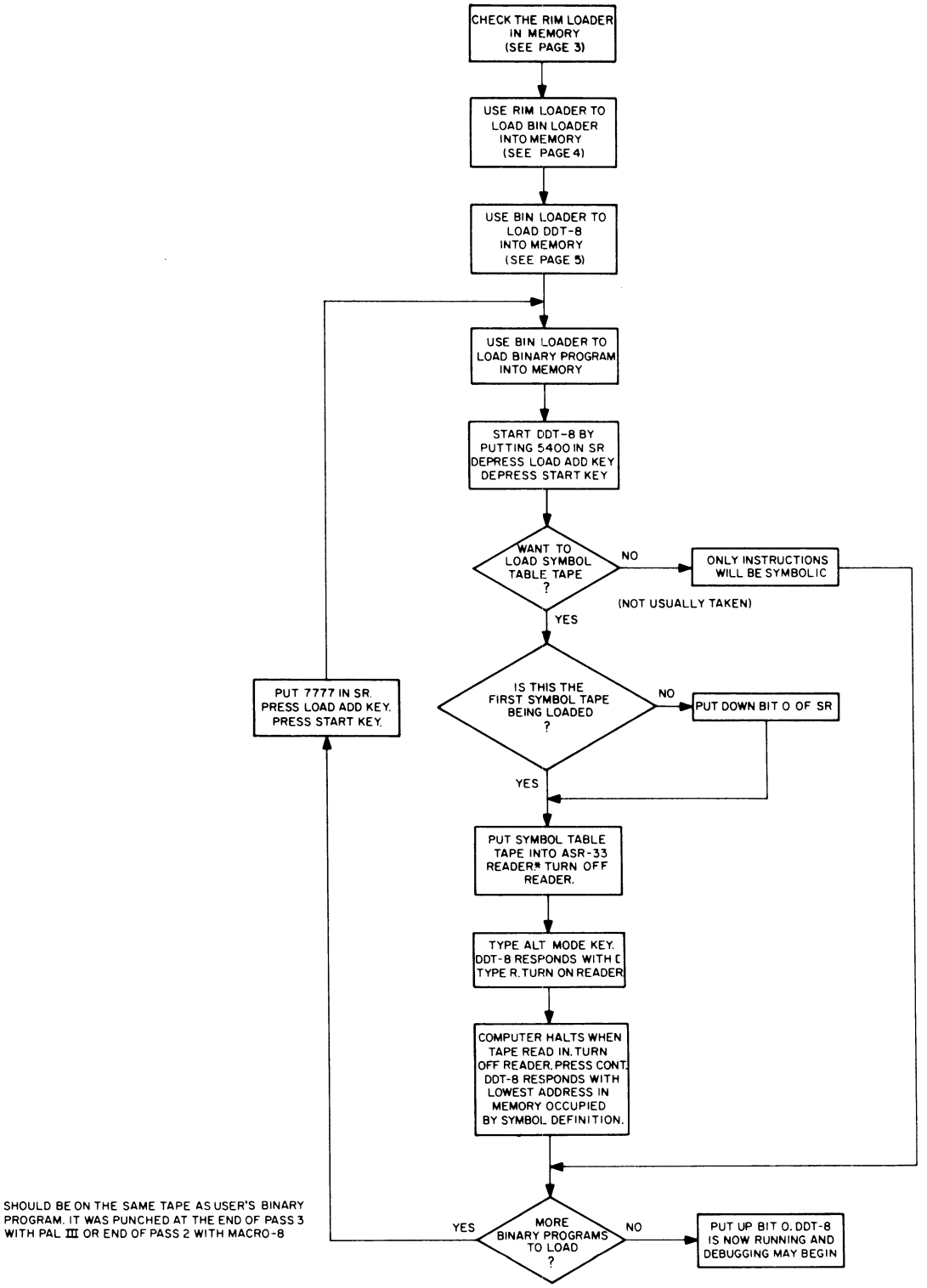


Figure 10 Loading Programs for Debugging

DDT-8

The PDP-8 DDT Programming Manual, Digital-8-4-S, contains a complete discussion of the debugging program. The reader must have a thorough knowledge of this manual before using the commands summarized in table 4. Figure 10 illustrates the process for loading programs to be debugged. Figure 11 shows how new symbols are opened during debugging. Figure 12 illustrates the procedure for punching out a correct binary tape using DDT-8.

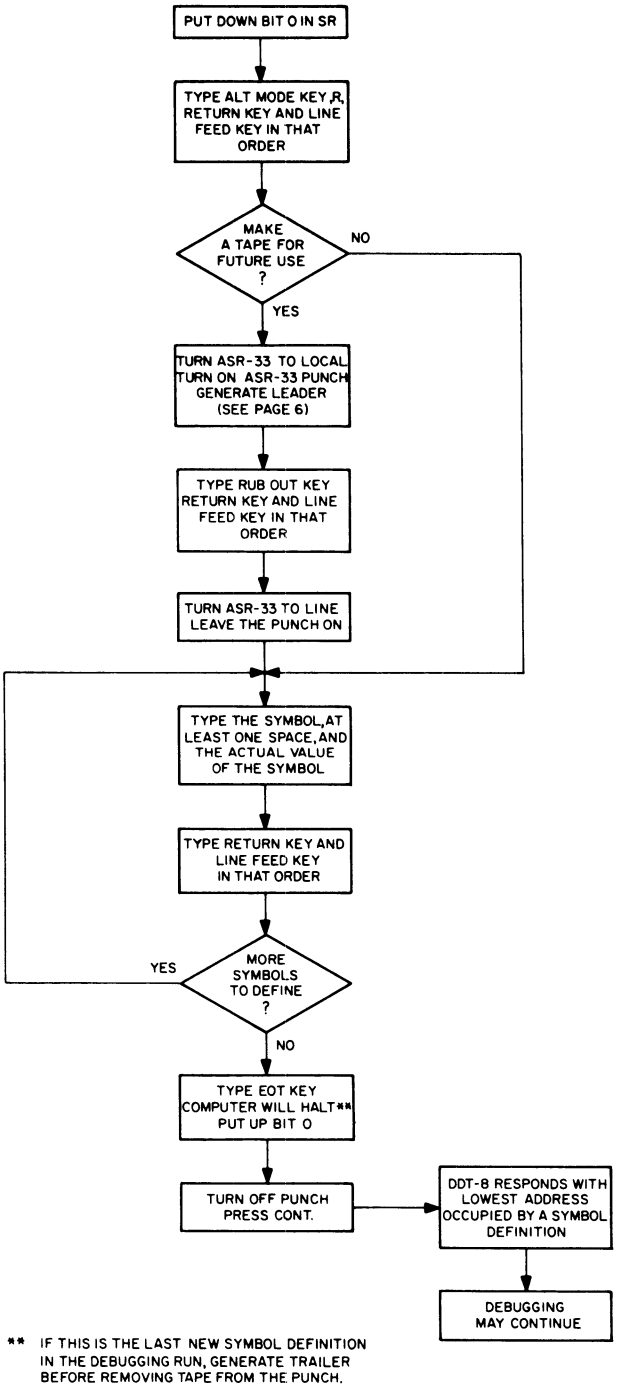


Figure 11 Defining New Symbols During Debugging (Option: Making New Symbol Tapes)

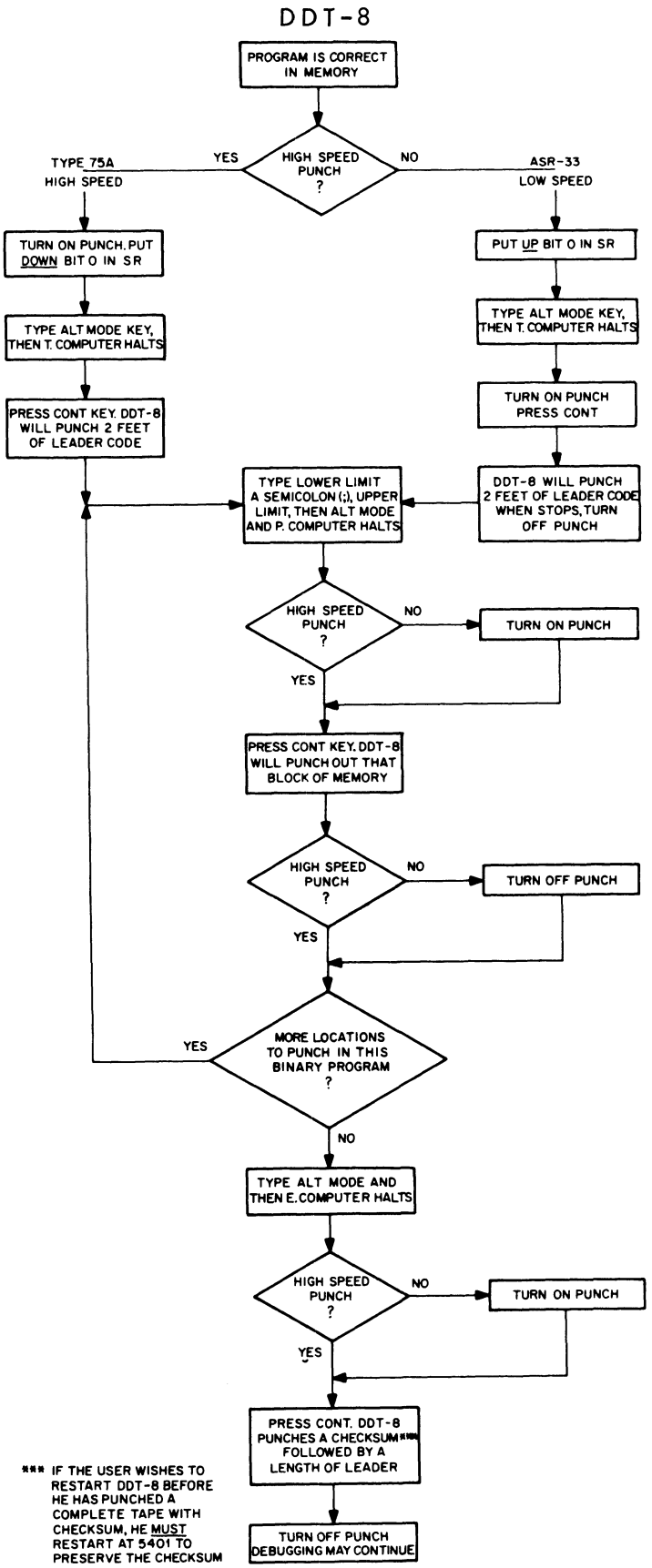


Figure 12 Punching Out a Correct Binary Tape

DDT-8

TABLE 4 SUMMARY OF DDT COMMANDS

Command	Action
	Input Commands
[R	Read symbol tape into external table from ASR 33 Reader. Define new symbol from keyboard.
	Program Examination and Modification Commands
k/	Open register k (where k may be either octal or symbolic).
RETURN key ( )	Close register currently open. Enter modification, if any.
LINE FEED key	Close register currently open and open next sequential register. Enter modification, if any.
t	Close register currently open and open register address therein. Enter modification, if any.
	Breakpoint Insertion and Control Commands
k [B	Insert a breakpoint at location k. If k is absent, the current breakpoint is removed.
k [G	Go to location k and start program execution.
n [C	Continue from breakpoint. DDT allows program to execute the breakpoint n times before DDT returns control to the user. If n is absent, it is assumed to be 1.
	Word Searches
N [W	Begin word search for all occurrences of the expression N masked by the contents of [M between the limits imposed by [L and [U. [M, [L, and [U are registers within DDT which may be opened, modified, and closed exactly as any general register k in the user's program ([M/7777).
	Mode Control (The initial mode of DDT is symbolic)
[0	Sets DDT to type out in octal mode.
[S	Sets DDT to type out in symbolic mode.
	Output Commands
[T	Punch leader/trailer code.
a;b [P	Punch binary tape from memory bounded by the addresses "a" and "b."
[E	Punch end of type (i.e., checksum block and trailer code).
	Special Characters
Space	Separation character.
+	Arithmetic plus. Used to specify address arguments relative to symbols.
−	Arithmetic minus. Used in the same was as "+."
.	Current location. Used in address arguments.
=	Type last quantity as an octal digit.
←	Delete line currently being typed. Continue typing on same line.

Note: The character "[" is generated by typing the ALT MODE key.

FORTRAN

DEC Library Tape No: Digital-8-2-U

COMPARISON OF FORTRAN II AND PDP-5/8 FORTRAN

The following features are common to both FORTRAN II and PDP-5/8 FORTRAN

1. Floating point functions (sine, cosine, square root, arc tangent, exponential, logarithm)
2. Arithmetic statements with usual operators (=, +, ., *, /, **) and standard order of computation
3. Fixed and floating point numbers with usual initial letter designation
4. Variables from I-N, fixed point; variables starting with any other letter, floating point
5. Control statements: IF, DO, GOTO, Computed GOTO, CONTINUE, STOP and PAUSE
6. Compile statements: DIMENSION, END
7. I/O statements: READ, WRITE, ACCEPT, TYPE
8. Format statements: H (modified), I, and E with "/" for CR/LF
9. Continuation character
10. Diagnostics at both compile and run-time

Only FORTRAN II has the following features:

1. User defined subroutines and functions
   - No SUBROUTINE statement
   - No CALL statement
   - No COMMON statement
   - No user defined FUNCTIONS
   - No EQUIVALENCE statement
2. "F" format on output
3. More than one subscript on variables
4. I/O statements for card, IBM-compatible tape, or line printer

Only PDP-5/8 FORTRAN has the following features

1. Facility for insertion of machine language instructions into object program
2. Dynamic debugging allows correction of errors in source tape at time of compilation
3. Multiple assignments in statements
4. Arithmetic statements may be embedded within:
   - arithmetic
   - control
   - input/output statements
5. One-pass compiler with facility for multiple compilation of more than one program.

FORTRAN

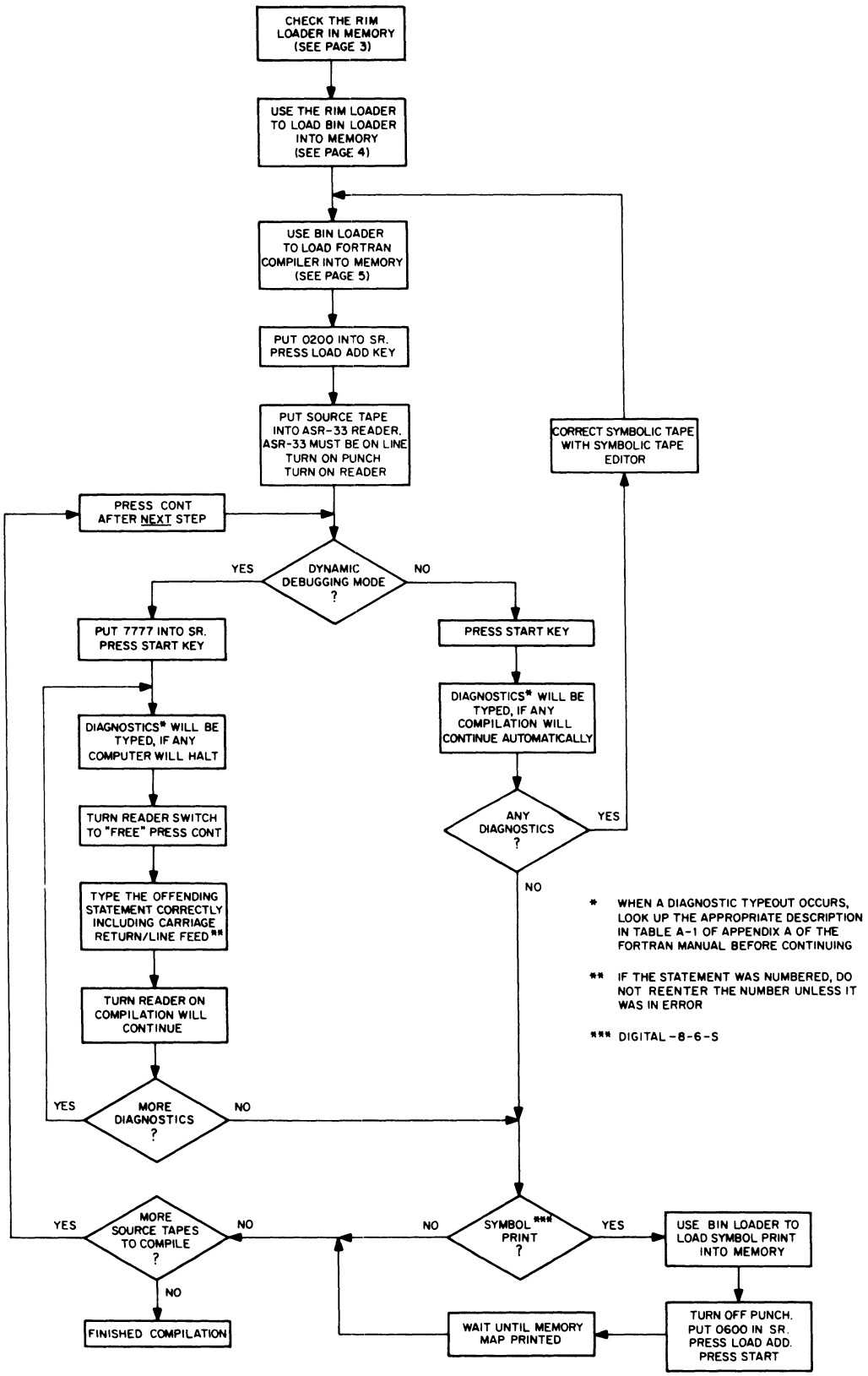


Figure 13 FORTRAN Compiler

PDP-8 SUBROUTINES

Name	Digital Number	Calling Sequence*	Memory Locations (Decimal)	Execution Time
		Function Subroutines		
1. Square Root Single Precision	8-9-F	- /Square in AC JMS SQRT /Call - /Return with root	23	30+25.5N μsec where N is the answer
2. Signed Multiply Single Precision	8-11-F	- /Multiplier in AC JMS MULT /Call ADDRESS /Address of multipli-/cand - /Return. High order product in AC Low order in MP1	44	360 μsec (max)
3. Signed Divide Signle Precision	8-12-F	- /High dividend in AC JMS DIVIDE /Call LOWD /Low dividend DIVSOR /Divisor - /Return quotient in AC. Remainder in HDIVND	62	460 μsec (avrg)
4. Signed Multiply Double Precision	8-13-F	- /AC ignored JMS DMUL /Call HORDMD /Address of high order multiplicand HORDMR /Address of high order multiplier - /Return high order product in AC. Remainder of product in B, C, D.	125	1.4 msec (avrg) 1.5 msec (max)
5. Signed Divide Double Precision	8-14-F	- /Address of high order dividend in AC JMS DUBDIV /Call HORDDR /Address of high order divisor - /Return. High order quotient in AC. Low order quotient in DIVND4. High and low remainder in DIVND1 and DIVND2	105	1.65 msec (avrg) 1.75 msec (max)

*All of the calling sequences here assume that the data is in the correct format and that there are no overflow conditions to check upon completion. For details on the data and indicators for overflow conditions, the user is referred to the appropriate program write-up.

PDP-8 SUBROUTINES

Name	Digital Number	Calling Sequence*	Memory Locations (Decimal)	Execution Time
		Function Subroutines (cont)		
6. Sine Routine Double Precision	8-16-F	- /AC = 0000 JMS DSIN /Call ADDRESS /Address of high order word - /Return. AC = 0 1 = 0 Answer in ADDRESS and ADDRESS + 1	248 (+ double precision multiply	Without EAE multiply 10.4 msec (avrg) 10.6 msec (max) With EAE multiply 2.61 msec (avrg) 2.78 msec (max)
7. Cosine Routine Double Precision	8-18-F	- /AC = 0000 JMS DCOS /Call ADDRESS /Address of high order word - /Return. AC = 0 1 = 0 Answer in ADDRESS and ADDRESS + 1	64 (+ double precision sine and double precision multiply)	0.075 to 0.093 msec longer than sine routine
8. Four-Word Floating Point Package	8-20-F	See Floating Point System Programming Manual, Digital-8-5-S	1041	See 8-5-S
9. Signed Multiply (EAE) Single Precision	8-21-F	/Multiplier in AC JMS MULT /Call ADDRESS /Address of multiplicand - /Return. Most significant product in AC; least significant in MPI		9 μsec (min) 71 μsec (max) 66 μsec (avrg)
10. Signed Divide (EAE) Single Precision	8-22-F	- /High dividend in AC JMS SPDIV /Call LOWD /Low dividend DIVSOR /Divisor - /Return. Quotient in AC. Remainder in DVD.	45	46.5 μsec (min) 98.0 μsec (avrg) 106.5 μsec (max)
11. Signed Multiply (EAE) Double Precision	8-23-F	- /AC ignored JMS DMUL /Call HORDMD /Address of high order multiplicand HORDMR /Address of high order multiplier /Return high order product in AC. Remainder in B, C, and D.	104	284.3 μsec (min) 270.0 μsec (avrg) 293.3 μsec (max)

*All of the calling sequences here assume that the data is in the correct format and that there are no overflow conditions to check upon completion. For details on the data and indicators for overflow conditions, the user is referred to the appropriate program write-up.

PDP-8 SUBROUTINES

Name	Digital Number	Calling Sequence*	Memory Locations (Decimal)	Execution Time
		Function Subroutines (cont)		
12. EAE Floating Point Package	8-25-F	See Floating Point System Programming Manual, Digital-8-5-S	See 8-5-S	See 8-5-S
		Utility Programs		
Punch Programs				
1. RIM Punch	8-4-U	Binary tape (see write-up)		I/O bound
2. BIN Punch ASR-33 75A	8-5-U	Binary tape (see write-up)		I/O bound
Processor Programs				
1. Logical Subroutines Inclusive OR	8-7-U	- /One argument in AC JMS INCOR /Call ADDRES /Address of second argument - /Return	12	30.0 μsec
Exclusive OR		JMS EXCOR /One argument in AC /Call ADDRES /Address of second argument /Return	14	34.5 μsec
2. Arithmetic Shift	8-8-U	(General Calling Sequence) - /Negative number of shifts in AC JMS** /Call ADDRES /Address to be shifted - /Return with shifted number in AC		N represents the number of shifts
Shift Left, Single Prec.		**SPSL	12	21 + 6.0N μsec
Shift Right, Single Prec.		**SPSR	15	(+)21 + 9.0N μsec (-)21 + 10.5N μsec
Shift Left, Double Prec.		**DPSL (Least significant part in LSH)	24	37.5+19.5N μsec
Shift Right, Double Prec.		**DPSR (Least significant part in LSH)		+37.5+22.5 μsec -37.5 +24.0N μsec

*All of the calling sequences here assume that the data is in the correct format and that there are no overflow conditions to check upon completion. For details on the data and indicators for overflow conditions, the user is referred to the appropriate program write-up.

PDP-8 SUBROUTINES

Name	Digital Number	Calling Sequence*	Memory Locations (Decimal)	Execution Time
		Utility Programs (cont)		
3. Logical Shift	8-9-U			
Shift Right, Single Prec.		**LSRSP	12	21 + 6.0N μsec
Shift Right, Double Prec.		**LSRDP (Least significant part in LESTSG)	24	34.5+22.5N μsec
(Left Shift Logical identical to left shift arithmetic)				
4. EAE Instruction Set Simulator	8-17-U	(See write-up)		
BCD - Binary Conversion				
1. BCD-to-Binary Conversion	8-10-U	- /BCD number in AC JMS DCDBIN /Call	26	51.0 μsec
2. BCD-to-Binary Conversion	8-11-U			
3. Binary-to-BCD	8-14-U	- /Binary number in AC JMS BCD /Call - /BCD number in AC	33	216 μsec (min) 235 μsec (max)
4. Binary-to-BCD Conversion (Used primarily for writing mag tape in BCD Format)	8-15-U	- /AC contains binary number JMS BCD /Call - /Return. BCD number in ONE and TWO	53	304.5 μsec (min) 328.5 μsec (max)
Teletype Message Sub-Routines				
1. Alphanumeric Message	8-18-U	- /AC ignored JMS MESAGE /Call - /First two characters of message .) .) /Remaining characters .) .) "XX" represents the last character XX00 /End of message code - /Return	51	I/O bound

*All of the calling sequences here assume that the data is in the correct format and that there are no overflow conditions to check upon completion. For details on the data and indicators for overflow conditions, the user is referred to the appropriate program write-up.

PDP-8 SUBROUTINES

Name	Digital Number	Calling Sequence*	Memory Locations (Decimal)	Execution Time
		Utility Programs (cont)		
2. Teletype Output Package	8-19-U	Return is to location following "call" AC = 0000	75	I/O bound
Type One Character		- (/AC 0-5 = 00 (/AC 6-11 = trimmed code JMS TYPE /Call		
Type Two Characters		- /AC 0-5 = 1st character /AC 6-11 = 2nd character JMS TY2 /Call		
Type a Digit		- /AC 8-11 = digit JMS TDIG /Call		
Type a Space		JMS TYSP /Call		
Type a CR and LF		JMS TYCR /Call		
Type a Tab		JMS TYTB /Call		
3. Character String Typeout	8-20-U	- /AC = initial address JMS TYPSTG /Call of string - /Return. AC clear	64	I/O bound
Decimal Print Subroutines				
1. Unsigned Decimal Print, Single Prec.	8-22-U	- /AC contains numbers JMS DECPRT /Call - /Return. AC clear	38	I/O bound
2. Signed Decimal Print, Single Prec.	8-23-U	- /AC contains number JMS SSPRNT /Call - /Return. AC clear	48	I/O bound
3. Unsigned Decimal Print, Double Prec.	8-24-U	- /AC ignored JMS UDPRNT /Call ADRESS /Address of high order word - /Return. AC clear	73	I/O bound
4. Signed Decimal Print, Double Prec.	8-25-U	- /AC ignored JMS SDPRNT /Call ADRESS /Address of high order word - /Return. Ac clear	86	I/O bound

*All of the calling sequences here assume that the data is in the correct format and that there are no overflow conditions to check upon completion. For details on the data and indicators for overflow conditions, the user is referred to the appropriate program write-up.

PDP-8 SUBROUTINES

Name	Digital Number	Calling Sequence*	Memory Locations (Decimal)	Execution Time
		Utility Programs (cont)		
Decimal Input Routines				
1. Decimal to Binary Input. Signed or Unsigned, Single Prec.	8-28-U	- /AC ignored JMS SICONV /Call - /Return. AC contains number	74	I/O bound
2. Decimal to Binary Input, Signed or Unsigned, Double Prec.	8-29-U	- /AC ignored JMS DICONV /Call ADRESS /Address for high order word - /Return. AC clear	110	I/O bound
Miscellaneous				
1. Octal Memory Dump	8-6-U	None	77	I/O bound
2. DECtape Library System Loader	8-3-U	(See write-up)	17	I/O bound
3. Incremental Plotter Subroutines	8-12-U	(See write-up)	128	I/O bound
4. Symbolic Tape Format Generator	8-21-U	(See write-up)	-	I/O bound
5. DECtape Subroutine	8-27-U	(See write-up)	256	I/O bound

*All of the calling sequences here assume that the data is in the correct format and that there are no overflow conditions to check upon completion. For details on the data and indicators for overflow conditions, the user is referred to the appropriate program write-up.

APPENDIX 1

OFF-LINE PREPARATION AND EDITING OF SYMBOLIC TAPE USING ASR-33

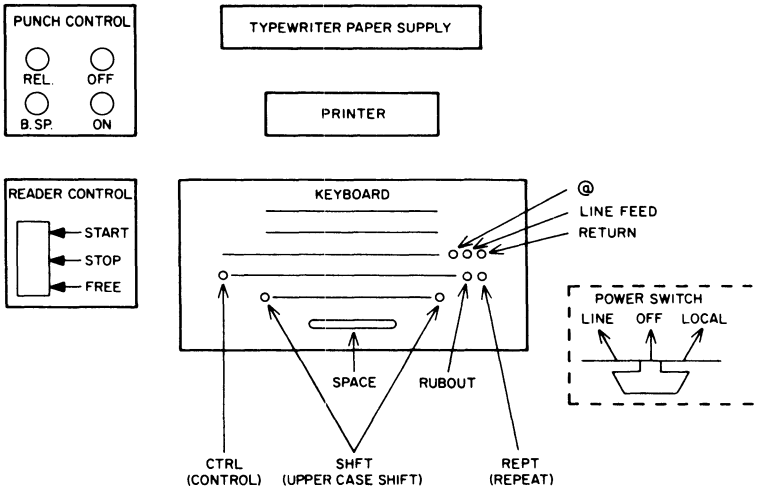


Figure 14 ASR 33 Teletype Console

DESCRIPTION OF FUNCTIONS

Punch Control

REL — Release. The tape is released from the punch mechanism and may be pulled forward or backward.

B.SP. — Backspace. The tape in the punch is backspaced by one character. The backspace is used in correcting tapes.

OFF — Punch is turned off.

ON — Punch is turned on. Codes of characters printed on printer are punched onto tape.

Reader Control

FREE — Reader is stopped. The operator may advance or reverse the tape.

STOP — Reader is stopped. The tape may not be advanced.

START — Power switch on LINE--The tape is read and advanced by KRB instruction from computer.

Power switch on LOCAL-- tape is read, character on printer, and advanced to next character. Used for listing tapes off-line.

Power Control

LINE — The ASR 33 is ON and connected electrically to the computer. The unit responds to any commands from the computer. Characters typed on the keyboard or read from the reader will not be printed on the printer unless a print command is given by the computer.

LOCAL — The ASR 33 is ON but disconnected electrically from the computer. The unit does not respond to any commands from the computer. Characters typed on the keyboard or read from the reader are printed on the printer.

OFF — The ASR 33 is off.

OPERATING PROCEDURES

Symbolic Tape Generation

1. Turn power switch to LOCAL.

2. Check quantity and positioning of typewriter paper and paper tape for punch.

3. Press the punch ON button.

4. Generate about 1-2 feet of leader code (200_8) as follows:

   a. Press and hold down keys marked CRTL and SHFT with left hand.

   b. Press and hold down REPT key with right hand.

   c. Press key marked @ and hold all four until desired length of leader is generated.

   d. Release REPT key first and then the remaining three keys.

5. Type the symbolic program.

6. Generate about 1-2 feet of trailer code (200_8) as described in step 4.

7. Tear off tape on cutting edge of punch slot.

Editing of Symbolic Tape

1. During generation of symbolic tape - An incorrect character may be typed while preparing the symbolic tape. If the error is detected N characters after typing the incorrect character, press the punch B.SP. (backspace) button N + 1 characters. Press the RUB OUT key N + 1 times to obliterate the N + 1 characters (i.e., punch 377_8 on tape). Then type the correct character and continue. Usually the incorrect character is detected immediately after typing. In that case, N = 0.

2. After generation of entire symbolic tape.

   a. Insertions - Duplicate the tape up to the point at which it is desired to make the insertion as follows:

      Turn on punch with reader switch set to FREE.

      Place symbolic tape in reader.

      Turn reader at desired point using printout as guide.

   Type the insertion.

Continue by turning reader switch to START.

   b. Deletions

   Duplicate the tape up to the point at which it is desired to make deletion as in a, "Insertions."

      Turn off punch.

      Start reader and continue until end of deletion using the printout as guide.

      Stop reader.

      Turn on punch.

      Continue by turning reader switch to START.

These procedures are given to assist operator when on-line tape editing cannot be done using the Symbolic Tape Editor.

The procedures given in part 2 may also be used to reproduce tapes, either binary or symbolic, but verification is not provided.

APPENDIX 2
ERROR DIAGNOSTICS DURING ASSEMBLY

PAL III

Pass 1

The Assembler reads the source tape, defines all user symbols, and outputs the user symbol table in alphabetic order. Pass 1 diagnostics are:

IC dddd AT xxxx Illegal Character

where dddd is the value of the illegal character and xxxx is the value of the current location counter when the character was processed. The character is ignored.

RD XXXXXX AT dddd ReDefinition

where XXXXXX is the symbol being redefined and dddd is the value of the current location counter at the point of redefinition. The symbol is redefined.

DT XXXXXX AT dddd Duplicate Tag

An attempt is being made to redefine a symbol using the comma. XXXXXX is the symbol and dddd is the value of the current location counter. The previous value of the symbol is retained, and the symbol is not redefined.

ST XXXXXX AT dddd Symbol Table Full

where XXXXXX is the symbol causing the overflow and dddd is the value of the current location counter at the point of overflow. The Assembler halts and may not be restarted.

UA XXXXXX AT dddd Undefined Address

where XXXXXX is the symbol that was used, but never defined, and dddd is the value of the current location counter when the symbol was first processed. This is typed with the symbol table at the end of pass 1. The symbol is assigned a value equal to the highest address on the memory page where it was first used.

Pass 2

The Assembler reads the source tape and using the symbol table defined during pass 1, generates and punches the binary code. This binary tape may then be loaded by the Binary Loader. The pass 2 diagnostic is:

IR dddd AT xxxx Illegal Reference

where dddd is the address being referenced and xxxx is the value of the current location counter. The illegal address is then treated as if it were on the proper memory page. Example:

```
*7306
 JMP  307
```

produces:

```
 IR 0307 AT 7306
```

and generates 5307 to be loaded into location 7306.

Pass 3

The Assembler reads the source tape and, using the symbol table defined during pass 1, generates and types the code represented by the source statements. The current location counter, the contents, and the source statement are typed side by side on one line. If bit 11 of the switch register is a 1 and the machine configuration includes the high-speed punch, the assembly listing is punched in ASCII. The pass 3 diagnostic is Illegal Reference.

PE Current Nonzero Page Exceeded - An attempt was made to override a literal with an instruction, or override an instruction with a literal. This can be corrected by decreasing the number of literals on the page or decreasing the number of instructions on the page.

ZE Zero Page Exceeded - Same as PE only with reference to page 0.

ID Illegal Redefinition of a Symbol - An attempt was made to give a previously defined symbol a new value not via the "=". The symbol was not redefined. (This is similar to the duplicate-tag diagnostic of PAL III.)

IC Illegal Character - # % ' : ? @ \ were processed in neither a comment nor a TEXT field. The character is ignored and the assembly continued.

A nonvalid character was processed. The computer halts with the illegal character displayed in the accumulator. Assembly may be continued by putting the desired character in the SWITCH REGISTER and depressing CONTINUE.

IE Illegal Equals - An equal sign was used in the wrong context.

II Illegal Indirect - An out of page reference was made, and a link could not be generated because the indirect bit was already set.

LG Link Generated - A link was generated for an out of page reference at this address. This is not an error message but only an indicator for the programmer.

SE Symbol Table Exceeded - The symbol table overlaps the macro table or vice versa. Assembly is halted and cannot be continued.

IM Illegal Format in a Macro Definition - The expression after the DEFINE pseudo instruction does not comply with the macro definition, position, or structural rules.

US Undefined Symbol - A symbol has been processed during pass 2 that was not defined by the end of pass 1.

MP Missing Parameter in a Macro Call - An argument, called for by the macro definition, is missing.

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

Printed in U.S.A.

12/66