PDP-9 SUPERVISORY PROGRAMS FOR THE DATA-LINKAGE TO IBM 360/67

by

NAM NG

B.Sc. (Eng.) University of Hong Kong, 1967

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS OF THE DEGREE OF

MASTER OF APPLIED SCIENCE

in the Department of
Electrical Engineering

We accept this thesis as conforming to the
required standard

Research Supervisor ....

Members of the Committee ....

....

Acting Head of the Department ....

Members of the Department
of Electrical Engineering
THE UNIVERSITY OF BRITISH COLUMBIA

August, 1970

Department of Electrical Engineering

The University of British Columbia
Vancouver 8, Canada

Date Aug 20, 1970

ABSTRACT

PDP-9 programs are provided for the inter-computer linkage of the PDP-9 in the Department of Electrical Engineering to the IBM 360/67 in the Computing Centre of the University of British Columbia. Details of the communication procedure used are described. Automatic error detection and recovery is incorporated to ensure correct message transmissions. A program "TRSFER" has been written to operate the PDP-9 as a conversational terminal to the IBM 360/67. It also allows file transfers between the two computers. Details of the data-phone handler routine and operation of the program "TRSFER" are given in the Appendix.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

Page

ACKNOWLEDGEMENT

I would like to express my gratitude to my supervisor, Dr. G.F. Schrack, for his constant advice and encouragement during the whole course of the project. I would also like to thank Dr. J.S. MacDonald for reading the thesis and his valuable comments.

A note of appreciation is given to Mr. W. Dettwiler of the Computing Centre and his staff for their co-operation during the experiments, and to Mr. G. Austin for his technical assistance.

Thanks to Mr. C.Y. Suen and Mr. N. Borowski for proof reading the thesis, and especially to Miss Heather DuBois for typing the manuscript.

Financial assistance through National Research Council Grant (No. 67-0148) and a University of British Columbia Graduate Fellowship is gratefully acknowledged.

1. INTRODUCTION

With the development of time-sharing computers, and the fast growing variety of data processing applications, the demand for digital data transmission has been ever increasing. It brought about the interconnection of different computers, especially small scientific computers at remote locations to large central time-sharing computers, enabling users at one installation to communicate with, and to make use of the facilities of the other computer.

For these reasons, the PDP-9 computer of the Department of Electrical Engineering is linked to the IBM 360/67 of the Computer Centre. This enables users of the PDP-9 to communicate with the IBM 360/67, and to make use of the large storage capacity and fast computational ability of the IBM 360/67. With this linkage established, more sophisticated data-processing applications can be carried out on the PDP-9.

Communication between the two computers is made over voice-grade telephone cables. Two pairs of telephone cables are used, one for transmission, and the other for reception. Full-duplex transmission is possible, but the linkage operates essentially as a half-duplex system. However, there is still the advantage of reducing time used in line reversals in half-duplex systems, which serves to increase the transmission efficiency considerably.

The linkage operates in a standardized way. Communication code and procedures follow closely those of USASCII standards[1] (United States of America Standard for Information Interchange)--now the ANS(American National Standard). Hardware has been purchased from The Digital Equipment Corp. and is built according to the EIA (Electronic Industries Association) Standards[2].

Transmission and reception of data follow binary synchronous conventions[3]. Data is transmitted serial-by-bit and serial-by-character. Correct reception of data depends on bit and character synchronization of both stations. This is achieved at the beginning of each transmission by transmitting a synchronization pattern, and once synchronized, the character phase is maintained until the end of the transmission.

The two computers are linked together in such a way that the PDP-9 essentially operates as a conversational terminal to the IBM 360/67. Data transmission rate is 2400 bits per sec. Due to the data processing ability of the PDP-9, data received at the PDP-9 can be processed before they are sent to the IBM 360/67. However, because the IBM 360/67 is a time-sharing computer and there is no priority for data communication over the linkage, real time data communication is not possible. To ensure correctness of data received, automatic error recoveries by retransmission are incorporated.

An overall view of the communication linkage between the two computers is shown in Fig. 1. The PDP-9 is connected to the communication system via the data-phone DPØ9A[4] from the Digital Equipment Corporation. The

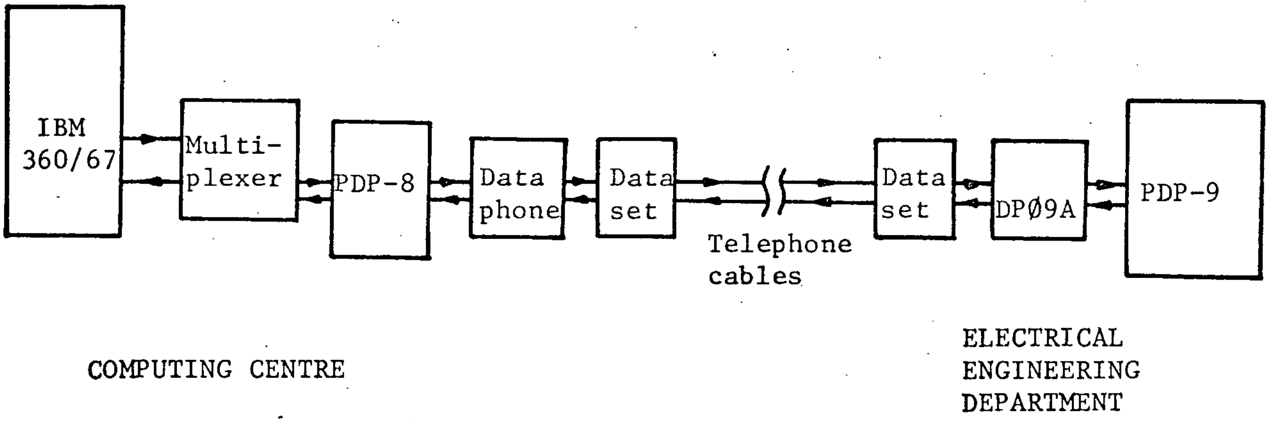


Fig. 1 Overall view of the communication linkage between PDP-9 and IBM 360/67

DPØ9A is a binary synchronous full-duplex communication channel for interconnecting the PDP-9 to the data set via the input-output interface of the PDP-9. It is essentially a serial-parallel binary data converter. Data from the PDP-9 accumulator are transferred to the DP∅9A via the I/O bus, where it is converted into a serial binary stream. It is then transmitted over telephone cables to the IBM 360/67 via the data set. The data sets, technically known as modems, are built according to the EIA standards. Data received from the data set are treated in a reverse manner. They are first assembled to form parallel 8-bit characters in the data phone before they are transferred to the PDP-9 accumulator via the I/O bus.

At the computing centre end, data are treated in a similar manner in the PDP-8. But, before the data are transferred to the central processing unit of the IBM 360/67 via the multiplexer, a code conversion is necessary since the communication code for the two DEC-computers is USASCII code whereas the IBM 360/67 works with EBCDIC code (extended binary-coded-decimal interchange code).

The two computers are linked by dedicated lines. Users of the PDP-9 can communicate with the IBM 360/67 at any time when the latter is operational. The procedure used for this linkage is similar to any of the other remote terminals of the IBM 360/67.

2. COMMUNICATION PROCEDURES

2.1 Communication Code

The USASCII code is used as the communication code. A table of the codes is shown in Appendix I. Each communication character is formed from the seven bits USASCII code with a parity bit as the eighth bit. Odd parity, as suggested by USASCII standards on communication codes[5], is used. Since the data-phone DP∅9A installed with the PDP-9 works with even parity synchronous idle characters, the logical circuitry of the DP∅9A has been modified to make it compatible with the communication system.

2.2 Message Format

In deciding the message format, the proposed standards have been followed closely. However, slight discrepancies are inevitable. A typical communication message is shown in Fig. 2.

SYN	SYN	SYN	SYN	SOH	SEL	STX	TEXT	↑Q	ETX	BCC

Fig. 2 Typical communication message format

To initiate transmission, the transmitting station sends a number of--in this system, four--synchronous idle characters to establish synchronization with the receiving station. Following the synchronization procedure, the Start of Heading character (SOH) is transmitted. This indicates the start of the heading of the message. The heading is generally intended to give some indication of the content of the message and is usually not revealed to the user.

A Select Character is included as the heading of the message, to increase the error detecting ability of the communication scheme. This character can only assume two values, viz. in 7 bits octal, 040 and 060, with parity included as the eighth bit. Consecutive messages are transmitted with the Select Character assuming these two values alternately. Should retransmission be necessary, the Select Character assumes its last value. Thus, a message currently received is considered correct and acceptable if and only if the Select Character is different from the one in the last accepted message. This procedure provides verification on the proper sequence of the messages received. It should be noted that the two values of the Select Character chosen are quite arbitrary. Any two values could have been chosen. In some systems[6] only one bit rather than one character is used for the same purpose.

The Start of Text (STX) character is transmitted following the Select Character. This character indicates the end of the heading and the start of the actual message or text that is communicated. The text is one complete line of data under the Michigan Terminal System (MTS) used by the IBM 360/67. In terms of punch card mode, this is equivalent to one card image; and in teletype mode, one line of text up to CNTRL Q (↑Q) or carriage return. The ↑Q character is always transmitted at the end of the message.

The text is terminated by the End of Text (ETX) character, followed by the Block Check Character (BCC, which is also known as block parity). The BCC is a text checking character. It is formed by taking the sum (modulo 2) of the message, starting from the character following the SOH character, i.e.

the Select Character, up to and including the ETX character. It thus acts as a longitudinal block parity checking character. Block Check Characters are formed at both ends of the linkage independently and the receiving station is responsible for comparing them. Identical characters indicate that the reception has been error free and will be accepted if no other errors are detected. Should the two be different, the message received is considered to be incorrect and request for retransmission is made.

After transmitting the BCC, an all zero bits character is sent to indicate the end of the transmission. The receiving station can then initiate to send the acknowledgement or reply.

2.3 Message Transfer Control Procedures

Though basically a full-duplex system, the linkage is operated in half-duplex mode. Nevertheless, there is still the advantage of obviating the need of line reversals, and this aids to simplify the communication control procedures.

To initiate communication, the transmitting station sends a complete line of message in the format described above. It then waits in search mode for the acknowledgement from the other station. While in search mode, the station searches for synchronous idle characters. Once detected, data-reception is initiated and the acknowledgement is examined. The acknowledgement is in the form of two consecutive identical characters. The only valid acknowledgement characters are the Positive Acknowledgement (ACK) and the Negative Acknowledgement (NAK). An ACK indicates that the last message has been correctly received, and the transmitting station can then proceed to transmit a new line of message, with the Select Character value alternated. Message transfer continues in this manner until all messages have been transmitted.

If a NAK is received, it indicates that the last transmission has not been correctly received and re-transmission of the message is required. The message is re-transmitted with the Select Character value unaltered and the transmitting station waits for the acknowledgement again. Up to five re-transmissions are attempted for error recovery for each message. If the error condition still persists, the communication is called to a halt. An error message is printed on the PDP-9 teletype. Under such circumstances, human intervention is necessary for communication recovery.

If neither ACK nor NAK is received, the acknowledgement is assumed to be garbled. An Enquiry (ENQ) character is sent for a re-transmission of the acknowledgement. Should this occur for more than five times, the communication is again halted and human intervention is required.

After transmitting all messages, the transmitting station terminates transmission by sending an End of Transmission (EOT) character. It then waits in receive mode, giving up its right of transmission to the other station.

The message transfer procedure is quite symmetrical for both stations. However, communication can only be initiated by a request from the PDP-9 since the PDP-9 is usually in a passive state as far as this linkage is concerned. The IBM 360/67, on receiving the request, can proceed to initiate message transfers. Fig. 3 shows a flow-chart for the above procedures implemented in the linkage.

2.4 Error Detection and Error Recovery Procedures

Inevitably, errors will occur during message transmissions. It is desirable to be able to detect these errors and if possible correct them automatically without human intervention. Error detection and recovery schemes are incorporated for such purposes. Though the scheme is not perfect, since it is not possible to detect all errors, the procedure is quite sufficient to

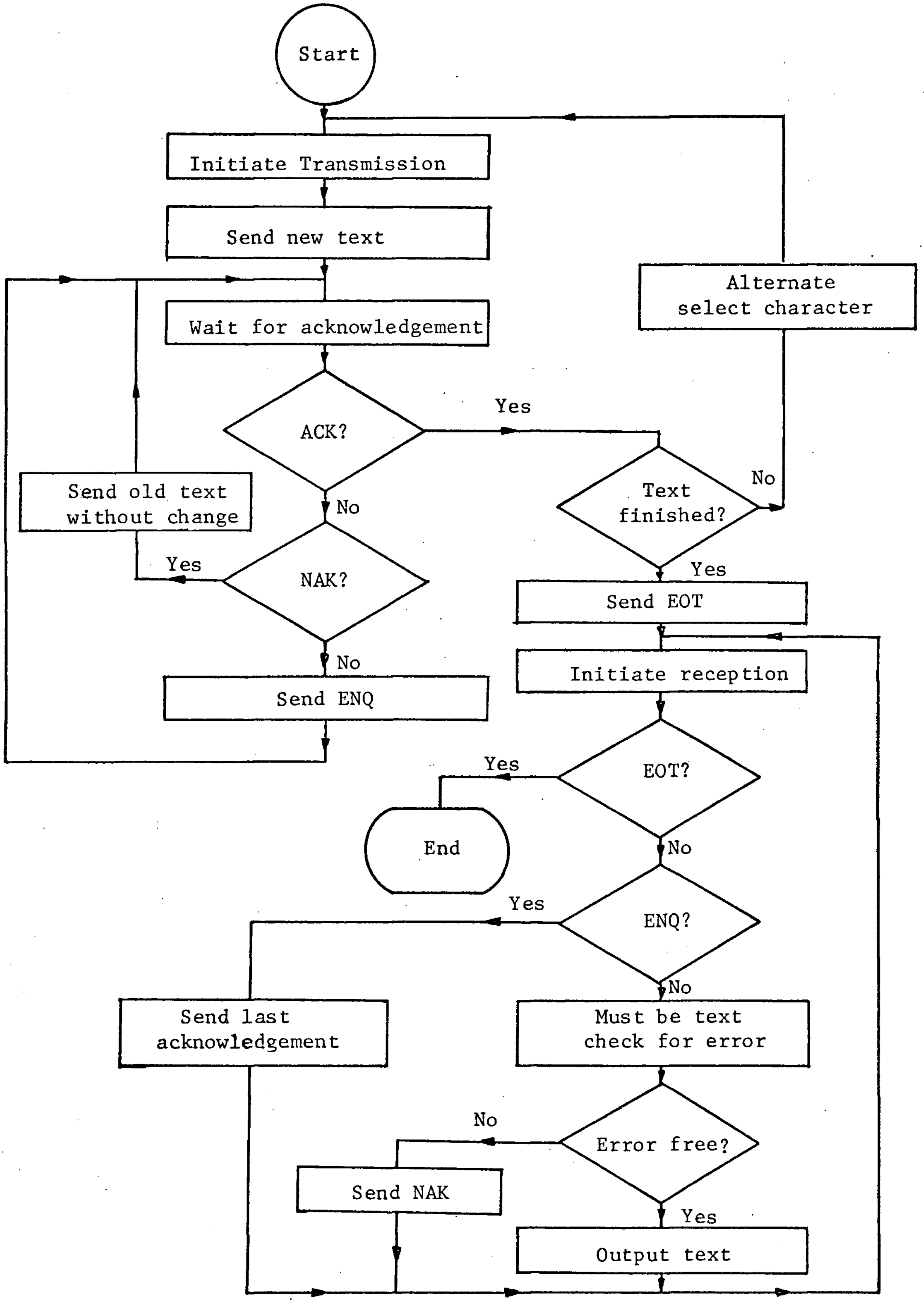


Fig. 3. General Flow-Chart of the Message Communication Scheme

handle most of the errors.

Errors are detected in several ways. To ensure that bits and characters are sampled at proper times, reception of data is initiated only after synchronization of the two stations. A "Time-out" restriction is also imposed on data reception by the data-phone: If any bit is not received within 1.5 bit times, the endflag is set and interrupts the computer[4], indicating the loss of true character phase.

The received data is tested for vertical parity before identification--whether data or control characters. This serves to detect all odd numbers of errors that occur in the seven bits of each character. The message is further tested for longitudinal parity using the BCC. This detects all odd numbers of errors that occur in any bit position of the seven-bit codes. The two parity checking methods combined act together in such a way that errors are not detected only if an even number of errors have occurrred both vertically for each character in the message and logitudinally for each of the bit positions of the message.

Checking by means of the logical sequence of the data received is also performed. At every point of reception, data are tested to ensure that correct or acceptable control characters are received. Messages can only begin with SOH, followed by the Select Character and STX, and terminated by ETX and the BCC. Failure to detect the control characters in correct sequence will be flagged as an error, and an error recovery procedure is called into effect.

The Select Character also plays an important role in the detection of errors in message block transmission. It detects the proper sequence of successive message blocks received and prevents the loss of message blocks. If there are an odd number of message drop-outs, the error is always detected.

A "Time-out" condition is also imposed on the acknowledgement of messages. The transmitting station expects to receive the acknowledgement within a certain period (10 seconds for the PDP-9 and 3 seconds for the IBM 360/67) after it has transmitted a message. If no reply whatsoever is received within this period, it is assumed that the receiving station has missed the last message completely and re-transmission is called into effect. This safeguards the blank-out of the entire communication.

Finally, to ensure that the key control characters are received correctly, they are always transmitted in duplicate. Misinterpreting any of the control characters could be disastrous. A simple example is the misinterpretation of NAK as ACK. The transmitting station would proceed to a new message while the receiving station expects the last message. Though this condition can still be detected through the select character, there is no way to recover without human intervention. The use of duplicate control characters should reduce the possibility of being confronted with such conditions.

To correct the errors detected, recovery by means of re-transmission is called into effect. The transmitting station, on detecting a negative acknowledgement, assumes that the last transmission was in error and will re-transmit the last message. Until it receives an ACK from the receiving station, the transmitting station will not proceed to the next message.

The above error detection and correction scheme is not ideal in that it cannot detect all errors, especially consecutive, multiple or bursts of errors. However, the scheme is quite sufficient for detecting single or isolated errors, which occur more frequently than multiple errors. With the incorporation of the above scheme, automatic operation of the linkage system is possible to a great extent. This is important as far as transmission efficiency is concerned. With the reduction of human intervention, the effective throughput is considerably increased.

3. PROVISIONS MADE FOR THE PDP-9

3.1.1 The data-phone handler

In providing system programs for handling data communication within the PDP-9, the data link to the IBM 360/67 is treated as one of the input-output devices of the PDP-9. The software is made available in the form of a standard I/O handler routine of the PDP-9 Input-Output Programming System[7]. I/O macro definitions are provided and can be called from PDP-9 assembly language programs. When any of these macros are called, the handler is entered via a CAL, the appropriate MTS command is generated and transmitted to the IBM 360/67 via the link.

To the IBM 360/67, the PDP-9 looks like one of the medium speed terminals. When a MTS command from the PDP-9 is received the command will be executed and the output will be directed back to the PDP-9. This will be dealt with by the message reception routine in the PDP-9. The routine first checks for errors and if any are detected, will initiate request for re-transmission; otherwise, the message received is assumed to be correct and will be output on one of the output devices of the PDP-9.

Though treated as one of the I/O devices of the PDP-9, the data-link differs slightly from the other handlers because the IBM 360/67 is an interactive device. It will respond in some fashion whenever it receives a command from the PDP-9. There is thus no distinction as to whether a data transfer on the device is input or output alone. Message transmission from the PDP-9 is always accompanied by message reception. Besides, since the IBM 360/67 is a time-sharing computer, there is no control over the instant the IBM 360/67 responds. Once the handler has transmitted a message, it has to wait in a receive mode for the message from the IBM 360/67. Thus, both input and output of data have to be handled whenever a macro is called.

The provision of the software as a standard I/O handler does not only relieve the programmer from writing his own routine for handling data transfers over the link, it also ensures compatibility with the Monitor System of the PDP-9. In this way, the unified way of handling all the I/O devices of the PDP-9 is preserved. Details pertaining to the handler are given in Appendix II.

3.1.2 The Macros

Ten different macros are provided. Details are given in Appendix III. The Macros provided are by no means sufficient for a one to one conversion to all MTS commands. Only the more likely used commands are made available. They include commands for sending data to the IBM 360/67, storing the data in a file, running the data with a compiled object program, and transmitting the subsequent output back to the PDP-9.

To send a particular MTS command other than those provided by the macros, the macro .LINK can be called. This macro not only allows the user to make up his own commands, it also allows the user to specify the data mode, whether source or binary, and the particular output devices used.

For data transfer to and from the core of PDP-9, which is the most efficient way of transferring data, the macros .READ and .WRITE can be used. No data transfer to any peripheral device is made and this serves to increase the effective throughput rate of data transfers. One disadvantage is that the volume of data transferred is limited by the size of the core available.

3.1.3 Data Modes

Data transfer in USASCII codes is the only data mode available at present. Binary data transfer is desirable since the PDP-9 works with binary data. This would require the incorporation of transparent mode binary data communication procedures[3]. Besides, because the PDP-9 has an 18-bit accumulator whereas the IBM 360/67 uses bytes (8 bits), a compromise has to be made in the number of bits used for each data word to ensure compatibility. The procedures for this data transfer scheme has not been fixed yet, but incorporation of the scheme is to be made.

One data conversion scheme is made available. This scheme converts the 18 bits binary data of the PDP-9 into six decimal digits in USASCII* before transmitting them to the IBM 360/67. Similarly, for data received, conversion back to binary is made before they are stored in the core of the PDP-9. The only restriction is that the data must be positive integers with magnitude less than 2^{18}. Further details of the data modes are given in Appendix IV.

Any of the above data mode transfers can be made via the macro .READ, .WRITE and .LINK by specifying the corresponding data mode when called. Details of the calling procedures can be found in Appendix III.

3.1.4 Error diagnostic messages

Error diagnostic messages are printed on the PDP-9 teletype whenever errors are detected by the error detection scheme and automatic recovery fails. Error messages are printed in the form DPERYY (Data-phone error YY) where YY indicates the type of the error that has caused the communication to come to a standstill. Errors of this category are mainly due to the garbling of messages

* (I6 in FORTRAN format specification).

on the line and hence errors are usually transients only, i.e. the same error may not occur again if the transmission is repeated.

Another set of error messages, pertaining to the Input-Output Programming System (IOPS) of the PDP-9, is also made use of. This set of error messages is indicative of the programming errors of the user, and is not related to errors encountered during data transfer.

A complete list of the error diagnostic messages is given in Appendix V. It is hoped that these two sets of error diagnostic messages should provide sufficient diagnostic ability for using the data-link.

3.1.5 The handler routine

A schematic flow chart of the handler routine is given in Fig. 4. It can be divided into three sections. The first section deals with the entry into the handler and initialization for the particular macro that is being called. The second part deals with the actual handling of data transmission and reception. The third part deals with error detection. Detailed flow charts for the handler are given in Appendix VI.

3.2 The program "TRSFER"

3.2.1 Operation as a terminal teletype

The program "TRSFER", when operated under "command mode", works essentially in the same way a terminal teletype does. Users can type in commands from the keyboard in the normal way the MTS operates. Outputs are printed on the teletype.

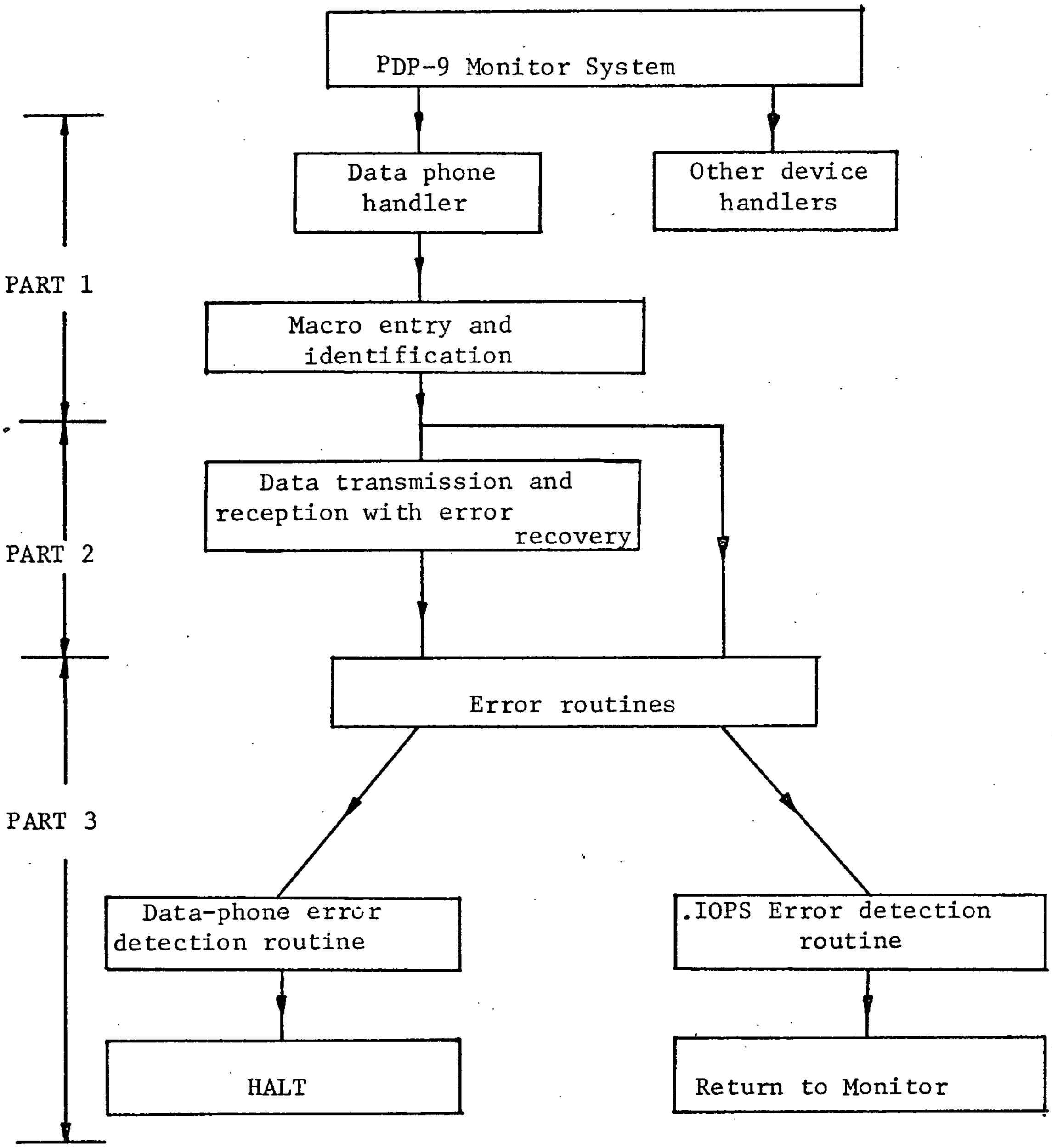

PDP-9 Monitor System
Data phone handler
Other device handlers
Macro entry and identification
Data transmission and reception with error recovery
Error routines
Data-phone error detection routine
.IOPS Error detection routine
HALT
Return to Monitor
PART 1
PART 2
PART 3

Fig. 4. Schematic Flow Chart for the Data-Phone Handler Routine

3.2.2 Data file transfers

When operated under "transfer mode", the program "TRSFER" can be used for file transfers between the I/O devices of the PDP-9 and the IBM 360/67. Operation is essentially similar to the Peripheral Interchange Program (PIP)[7] used under the Monitor System. Input devices include keyboard, paper tape reader and DECtapes. Output devices can be teletype, paper tape punch or DECtapes. The structure of a typical file transfer command is as follows:- T OPDEV : OPFILE ← IPDEV : IPFILE ; EXT↓ *

where OPDEV and IPDEV specify the output and input devices respectively.

OPFILE and IPFILE specify the output and input filenames respectively.

and EXT is the file extension specifying the type of data transfer--whether source or binary.

For example, the command

```
T 360 : FILE1 ←  DT1 : FILE2 ; SRC ↓
```

causes data stored in source file FILE2 on DECtape unit 1 to be transferred to FILE1 in IBM 360/67.

Filenames are required only when the device is file-oriented. For non-file-oriented devices, no filename specification is necessary.

A schematic flow chart for this program (TRSFER) is shown in Fig. 5. Details of the operation procedure under both command and transfer modes are given in Appendix VII and detail flow charts are given in Appendix VIII.

* ↓ denotes carriage return or alternate mode

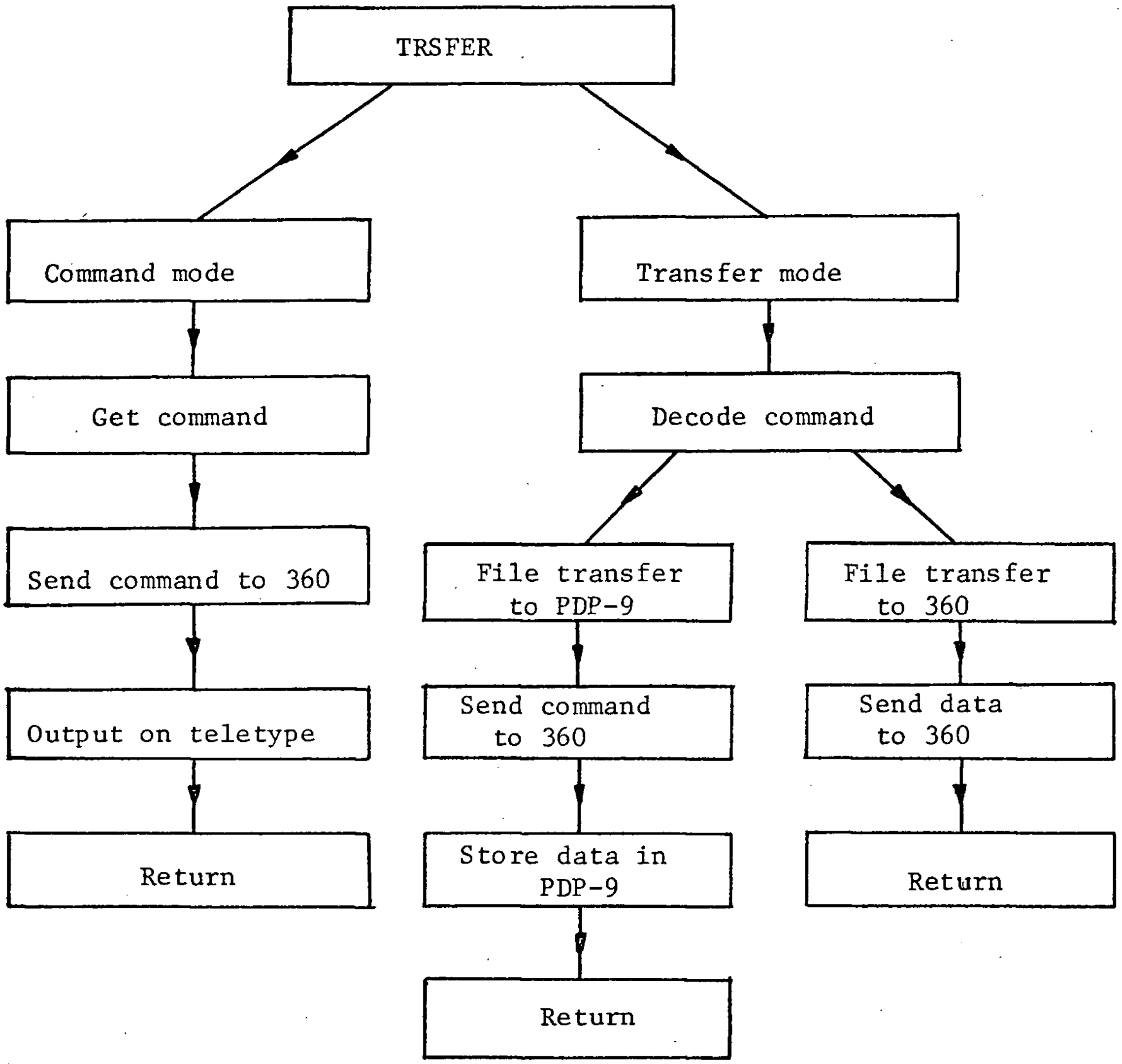


Fig. 5. Schematic Flow Chart for Program "TRSFER"

4. DISCUSSION

The data-link was still not fully operational at the time when this thesis was written. The cause of incompletion is due to the lack of a device support routine in the PDP-8 at the Computing Centre to handle the interactive operation of the linkage. Only data transfer from the IBM 360/67 to the PDP-9 is possible at present. Command for the transfer has to be entered from a teletype terminal. As far as this part is concerned, the linkage is now operational. The tests have been carried out quite successfully. Slight communication problems arose when output was made on the PDP-9 teletype. This is because the teletype is a slow device and it usually takes more than 3 seconds--the "time-out" period--to complete typing out a line. Very often, transmission and reception occur at the same time, resulting in the loss of the acknowledgement message. This problem has been overcome by sending an ACK whenever a message which has the same Select Character as the last accepted message is received. There is the risk of missing message blocks, but that is an irrecoverable error under any circumstances. For all the tests that have been carried out, no such problem has occurred.

Transmission of data from the PDP-9 to the computing centre has also been tested by sending data to the DCT 2000 line-printer in the Computing Centre. The DCT 2000 is an ASCII communication output device and works with the same communication procedure as our linkage. Data received are printed on the DCT 2000. No communication problem arises, especially after the incorporation of the "time-out" scheme in the PDP-9.

For completion of the linkage, the interactive communication scheme has to be tested. It is hoped that this will be completed in the near future.

The present communication system, after completion, should provide useful applications for the PDP-9 users. More sophisticated data processing problems than before can be carried out on the PDP-9. However, there is still room for improvement. A higher data transmission rate is desirable. This will be a first step to the process of carrying out real-time programming over the linkage. Of course, there will still be the necessity of having priority over other I/O devices attached to the IBM 360/67. Also, in increasing the transmission rate, although the present handler works with no timing problems at the transmission rate of 2400 bits/sec. (and should still work properly when the rate is increased to 4800 bits/sec. which is likely to be effective soon), detailed considerations have to be given to the execution time necessary to ensure that the program will still execute without timing problems when the transmission rate is increased beyond 4800 bits/sec. The incorporation of an attention interrupt is also desirable so that PDP-9 users can interrupt the data communication at any time. With the incorporation of these features, the linkage would become a very useful communication system for the two computers.

APPENDIX I: Table of USASCII codes for inter-computer communications*

Bit Positions											
b7				0	0	0	0	1	1	1	1
b6				0	0	1	1	0	0	1	1
b5				0	1	0	1	0	1	0	1
b4	b3	b2	b1								
0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2	"	2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB	'	7	G	W	g	w
1	0	0	0	BS	CAN	(	8	H	X	h	x
1	0	0	1	HT	EM	)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	K	[	k	{
1	1	0	0	FF	FS	,	<	L	\	l	|
1	1	0	1	CR	GS	-	=	M	]	m	}
1	1	1	0	SO	RS	.	>	N	^	n	¬
1	1	1	1	SI	US	/	?	O	_	o	DEL

░ First category (standard USASCII definitions)

Control characters

▨ Communication control characters

▧ Second category designated as Key characters for additional controls

□ Third category designated as argument characters (includes the USASCII control character DEL)

▥ Control character DEL included in the argument set

* This table is taken from Bhushan and Stotz's paper[8].

APPENDIX II: The Data-Phone Handler

2.1 General

Communication between the user program and the handler is made via a CAL instruction, followed by the pertinent argument list. The handler is of the non-reentrant type, which uses the DBR (debreak and restore) instruction to leave the handler at priority level 4, and restoring the link, the extend mode and the memory protect. The content of the accumulator at the time of CAL is not saved.

2.2 Incorporation of the handler into the system tape

The handler is incorporated into the system tape in the usual way. Details can be found in Section 3.4.4.7 (P.3-21) in the PDP-9 Advanced Software System Monitors Manual (DEC-9A-MAAO-D). Information pertaining to this handler is as follows:

a. The handler name is DPH.

b. Four skip IOTs should be in SKIP CHAIN for this device handler.

STF	702521	SKIP ON TRANSMIT FLAG
SRF	702621	SKIP ON RECEIVE FLAG
SEF	702541	SKIP ON RECEIVE END FLAG
SRI	702561	SKIP ON RING INDICATOR

Since all the above IOT skips occur when the flag is NOT set, it is necessary to precede each of these IOTs by a minus sign when entering them into the SKIP CHAIN.

After the new system tape has been generated, the binary program DPH. has to be updated into the system library program (.LIBR). Once this has been completed, the data-phone handler is then available for usage.

The present working version of the data-phone handler has a memory size of 3567_8 core locations. A shorter version of the handler can be obtained by dispensing with the macros that are not being used. For example, by deleting the .SIGON, .SIGOF and .RUN macros, the handler would only require 3335_8 core locations.

APPENDIX III: The macros

Macros provided for using the linkage are as follows:

NAME	FUNCTION CODE	PURPOSE
.INIT	1	Initialize the device handler
.SIGON	2	Initiate communication between PDP9 & 360
.COPY	3	Initiate .READ macro
.CREAT	4	Initiate .WRITE macro
.RUN	5	Send a $RUN command to 360
.SIGOF	6	Terminate communication
.LINK	7	Send commands as designated by user
.READ	10	Transfer data to user line buffer
.WRITE	11	Transfer data from user line buffer
.WAIT	12	Detect completion of each command to 360
.TRAN	13	Illegal function code to 360

3.1 .INIT

FORM: .INIT a,F,R

VARIABLES: a = Device Assignment Table (.DAT)

slot number (in octal radix)

F (not used) Therefore can have either the value 0 or 1

R (not used)

EXPANSION: CAL + F_{7-8} + a_{9-17}

1

R

400

DESCRIPTION: The macro .INIT initializes the data-phone handler. It performs a .SETUP for the IOT skip flags. .INIT must be given prior to any I/O commands referencing .DAT slot a; a separate .INIT command must be given for each .DAT slot referenced by the program. No file type specification is necessary for this device as input and output occurs for all commands.

3.2 .SIGON

FORM: .SIGON a, D1,N1,D2,N2

VARIABLE: a = .DAT slot number (octal radix)

D1 = address of ID number

N1 = ID number word count in octal (=4)

D2 = address of password and other optimal parameters

N2 = Password character count (octal)

EXPANSION: CAL + a

2

D1

N1

D2

N2

DESCRIPTION: The macro .SIGON initiates communication between the PDP-9 and the IBM 360/67. On encountering this macro, the handler sends the command: $SIG IDNO PW = PSWD *

to the IBM 360/67. From then on, the linkage is active and data transfer between the two computers can be made. Both IDNO and PSWD are assumed to be packed in 5/7 ASCII at locations D1 and D2 respectively. No header is required. For the parameter PSWD, it can include parameters other than the password itself

* The underlined parameters must be replaced by the user with the appropriate parameter.

by increasing the word count N2. Thus if location D2 contains 'PSWD T=50' and N2 is specified as 11 (octal), then the time limit parameter is also sent.

3.3 .COPY

FORM: .COPY a,D,N

VARIABLE: a = .DAT slot number (octal radix)

D = address of filename for .READ

N = filename character count (octal)

EXPANSION:

```
CAL + a
3
D
N
```

DESCRIPTION: The .COPY macro initiates action for subsequent .READ macros. D contains the IBM 360/67 filename which is to be transferred to the PDP-9 core location specified by the ..READ macro. On encountering the .READ macro, the command

```
$COP FILENAME >SYN5
```

is sent to the IBM 360/67. Note that '>SYN5' is the device name for directing output of IBM 360/67 to PDP-9.

3.4 .CREAT

FORM: .CREAT a,D,N

VARIABLE: a = .DAT slot number (octal radix)

D = address of filename for .WRITE

N = filename character count (octal)

EXPANSION:

```
CAL + a
4
D
N
```

DESCRIPTION: The .CREAT macro initiates action for subsequent .WRITE macros. D contains the name of the file, packed in 5/7 ASCII, which is to be created in the 360. It must precede any .WRITE macros. On encountering this macro, the following two commands are sent.

```
$CRE   FILENAME
$NUM
```

The user should check beforehand that he has sufficient file space for entering user's data when the .WRITE macro is executed.

3.5 .RUN

FORM: .RUN a,D1,N1,D2,N2,D3,N3

VARIABLE: a = .DAT slot number (octal radix)

D1 = address of the object filename in IBM 360/67 which is to be executed

N1 = Object file filename character count (octal)

D2 = address of the filename where data is stored in IBM 360/67

N2 = Data file filename character count (octal)

D3 = address of the file name where result is to be stored in IBM 360

N3 = Output file filename character count (octal)

EXPANSION:

```
CAL + a
5
D1
N1
D2
N2
D3
N3
```

DESCRIPTION: The .RUN macro sends the command $RUN OBFILE 5 = DATAFILE 6 = OPFILE to the IBM 360/67. The object file OBFILE stored in 5/7 ASCII at location D1 is to be executed. Data is obtained from DATAFILE which is stored at D2, and results are output on OPFILE stored at D3.

3.6 .SIGOF

FORM: .SIGOF a

VARIABLE: a = .DAT Slot number (octal radix)

EXPANSION: CAL + a

6

DESCRIPTION: The .SIGOF macro when called sends $SIG SHORT command to the IBM 360/67, thus terminating the communication between the two computers. Further communication is re-established only if .SIGON macro is again called.

3.7 .LINK

FORM: .LINK a,M,D,L,N

VARIABLE: a = .DAT slot number (octal radix)

M = data mode

D = output device

0→TTA teletype

1→DT1 DECtape 1

2→DT2 DECtape 2

3→PP paper punch

L = address of the line buffer where the user's command is stored

N = number of characters in the command (octal)

EXPANSION: CAL + M_{6-8} + a_{9-17}

7

D

L

N

DESCRIPTION: The .LINK macro when called sends the user's command (in MTS) to the IBM 360/67 for execution. The command is stored at location L in 5/7 ASCII. N is the number (in octal) of characters in the command. D specifies the output device where the result after the execution of the command is to be transferred to, and M specifies the data mode (0 = binary, 2 = 5/7 ASCII).

3.8 .READ

FORM: .READ a,M,L,W

VARIABLES: a = .DAT slot number (octal radix)

M = Data mode 0 = binary

1 = image 7 bits ASCII

2 = 5/7 ASCII

L = Data buffer address

W = Data Buffer character count (decimal radix), including the two word header

EXPANSION: CAL + M_{6-8} + a_{9-17}

10

L

.DEC

-W

DESCRIPTION: The .READ macro causes the command $COP FILENAME >SYN5↓to be sent to IBM 360/67. The FILENAME must be specified by a previous .COPY macro. Data is transferred to the data buffer area at location L, and in the data mode as specified by M. The number of words read into L is delimited by W or when an EOF is encountered as indicated by the IBM 360/67. Each line of data is terminated by a carriage return.

3.9 .WRITE

FORM: .WRITE a,M,L,W

VARIABLES: a = .DAT slot number (octal radix)

M = Data mode 0 = binary

1 = image 7 bits ASCII

2 = 5/7 ASCII

L = Data buffer address

W = Data buffer character count (decimal radix) including the two-word header

EXPANSION: CAL + M_{6-8} + a_{9-17}

11

L

.DEC

-W

DESCRIPTION: The .WRITE macro transfers data specified by L to the IBM 360/67. Data transfer stops when the character count goes to zero or when an "alternate mode" is encountered. Each line is terminated by a"carriage return". The data is transferred to the currently active file, and is normally specified by the .CREAT macro. At the end of data transfer, $UNN followed by $ENDFILE commands are sent before exit from this macro.

3.10 .WAIT

FORM: .WAIT a

VARIABLE: a = .DAT slot number (octal radix)

EXPANSION: CAL + A

12

DESCRIPTION: .WAIT macro detects the completion of each macro command. It should be used after all the data-phone macros with the exception of .LINK before the line buffer associated is used again, to ensure the data transfer is complete. If .LINK is used, location 32_8 should be checked for input-output completion. When it contains a -1, all input and output for this macro are finished, and control can then be returned to user's program.

APPENDIX IV: Data Transmission Paths

4.1 Line Buffer Headers

Line buffers for this handler have the same format as those used by the I/O Monitor System (Section 2.1.1 of PDP-9 Monitor Manual). This ensures compatibility with the other I/O devices of the system. Two header words are reserved for each buffer and the actual data starts after header word 1. Header word 1 is not used by this handler. For header word 0, bits 14-17 are used to indicate the I/O mode. The possible bit combinations are as follows:-

0000 = binary mode 0
0001 = Image ASCII mode 1
0010 = 5/7 ASCII mode 2
0101 = EOF (end of file),

no other bit combinations being used. Bits 1-8 are used for the word pair count, including the header word pair. But they only give the correct word pair count if the data input or output word pair count is less than 2^8-1. Bits 0, and bits 9-13 are not used.

4.2 Data Modes

4.2.1 Binary mode 0

In this mode, data has the form of 18 bits binary in the line buffers. Each word is first converted to decimal in ASCII before transmission. It is converted into the format I6. For data coming from the IBM 360/67, they are first converted back to 18 bits binary before they are packed into the line buffers. Again the format of the data coming in is assumed to be I6.

4.2.2 Image ASCII mode 1

In this mode, each word in the buffer is assumed to be ready in the form of 7 bits ASCII and is thus transmitted directly without conversion word by word. Data received are packed in a similar way.

4.2.3 5/7 ASCII mode 2

Data is assumed to be packed in the form of 5/7 ASCII. They are compatible with the IOPS ASCII except that the eighth bit used during data transfer is taken as the odd parity bit rather than even parity.

APPENDIX V: Error Detection and Handling

Two sets of error messages are incorporated in the data-phone handler. The first set, pertaining to the .IOPS, is useful for debugging programs. The second set is pertinent to data transmission over the linkage and usually indicates irrecoverable errors.

5.1 Data-Phone Error Messages (DPER)

Error messages are in the form of DPERXX where XX indicates the error code.

(1) DPERØØ = Message transmitted not accepted by IBM 360

(2) DPERØ1 = Acknowledgement from IBM 360 garbled

(3) DPERØ2 = Acknowledgement not accepted by IBM 360

(4) DPERØ3 = End of transmission not received by the IBM 360

(5) DPERØ4 = Garbled control characters received at the PDP-9 end

(6) DPERØ5 = Data set not ready while entering receive mode. (IOPSØ4 is indicated if power is not turned on at the beginning of transmission)

(7) DPERØ6 YY: Garbled text received by the PDP-9. YY indicates the cause of text garbling.

YY = Ø1 → Wrong select characters i.e. an odd number of text lines are missed.

= Ø2 → No Start of Text (STX) character received

= Ø4 → Wrong block checking character

= 1Ø → Wrong vertical parity detected in one or more characters of the text

= 2Ø → No end of text character received

For multiple errors, YY will appear as a combination of the above codes. Thus YY = 14 indicates that both vertical and longitudinal parity error occurred.

5.2 .IOPS error messages

Error message is in the form of

IOPS NN XXXXXX

where NN is the error code and XXXXXX the additional related information, usually address of CAL. (Refer to PDP-9 Monitors Manual Section 2.6 for details).

(1) IOPSØ4 - Data-set not ready

(2) IOPSØ6 - Illegal function for device handler. In this handler, function code 13 (octal) is illegal

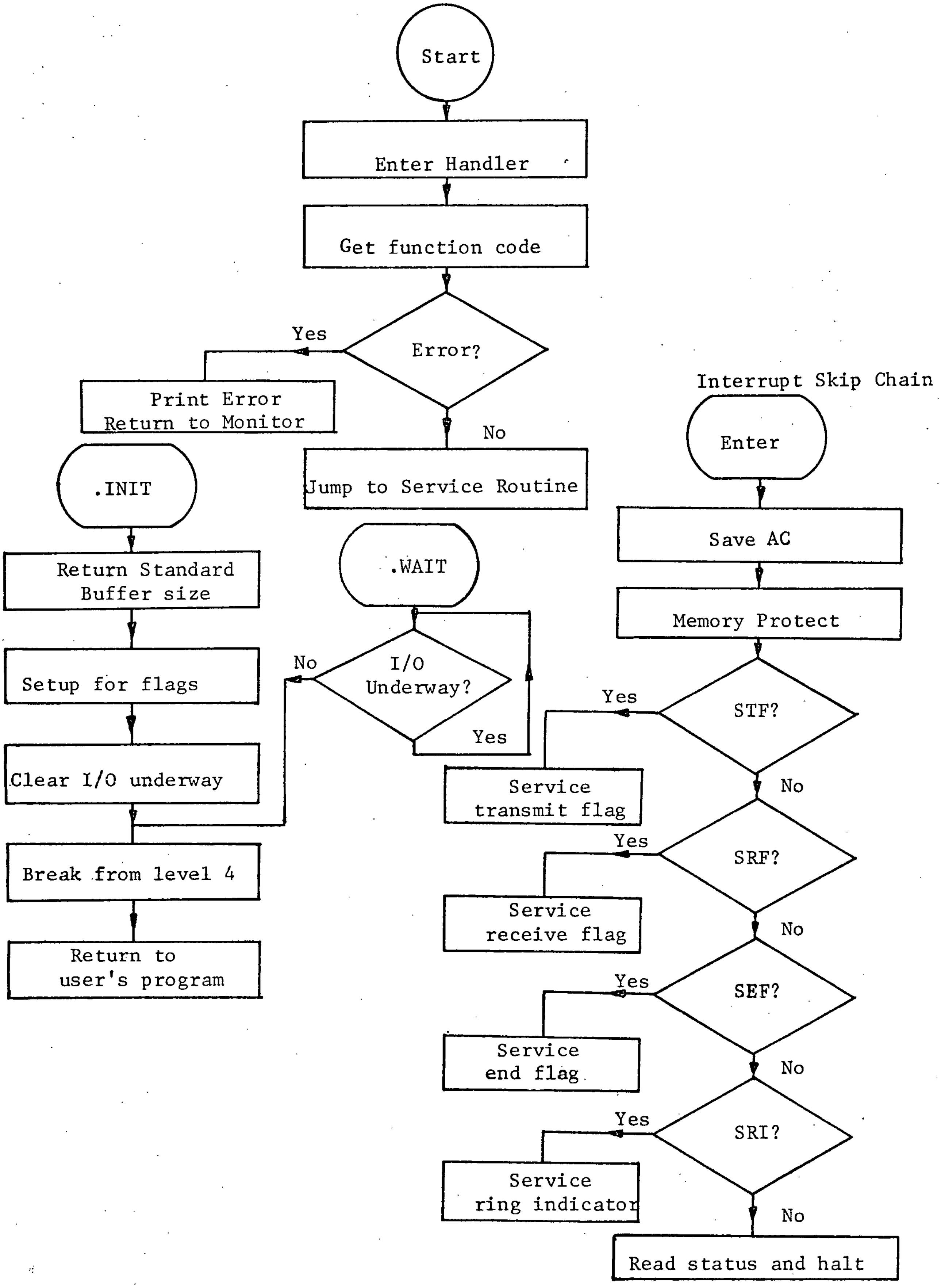
(3) IOPSØ7 - Illegal data mode for device handler. At present, only mode 0,1,2 are legal data modes for .READ and .WRITE

(4) IOPS11 - .COPY (.CREAT) not executed before .READ (.WRITE) macro

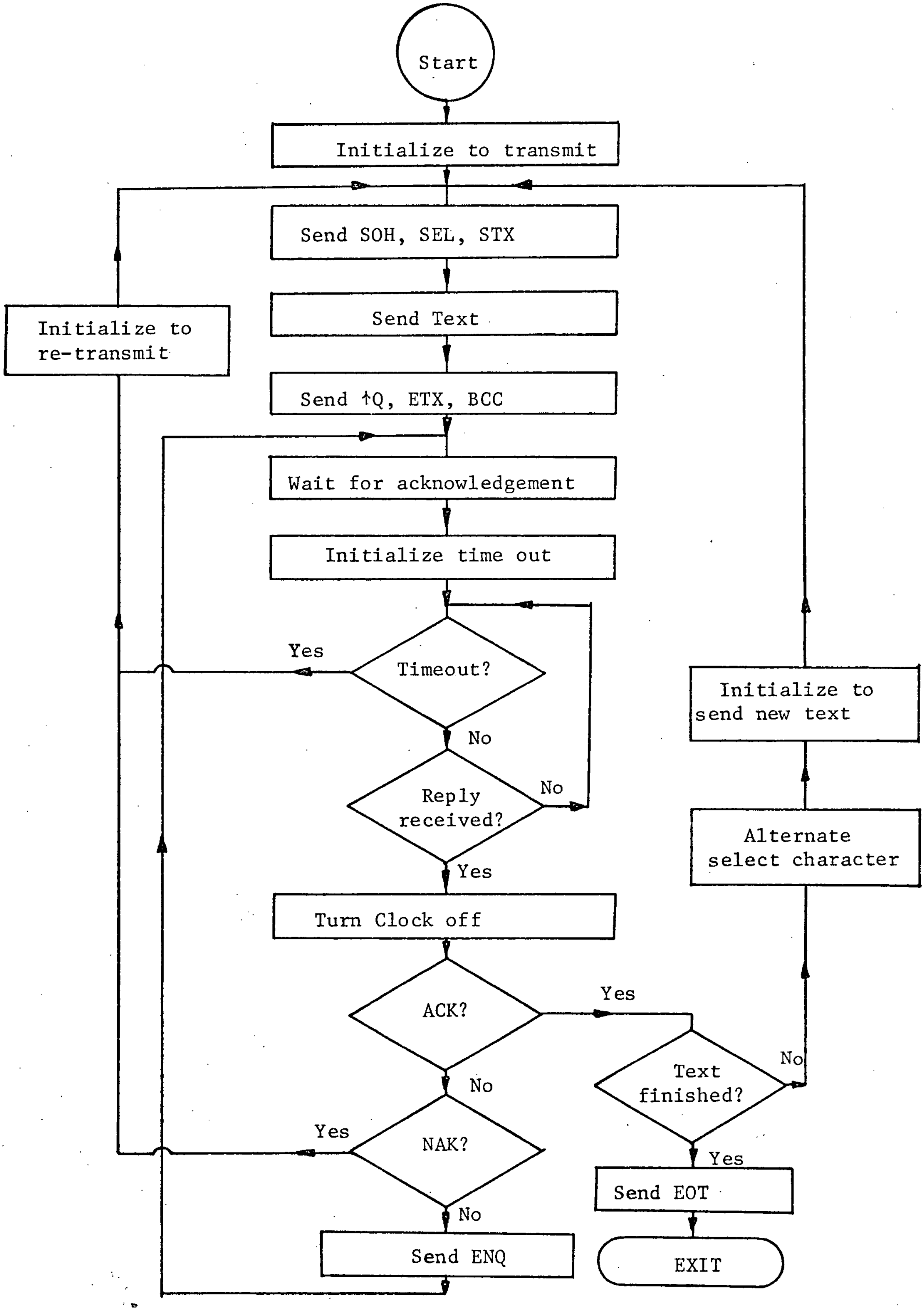
(5) IOPS16 - Output buffer overflow. This occurs on .READ when more characters are received than specified.

The above set of error messages usually indicates errors made by the programmer. They do not indicate errors encountered during transfer of data over the data link.

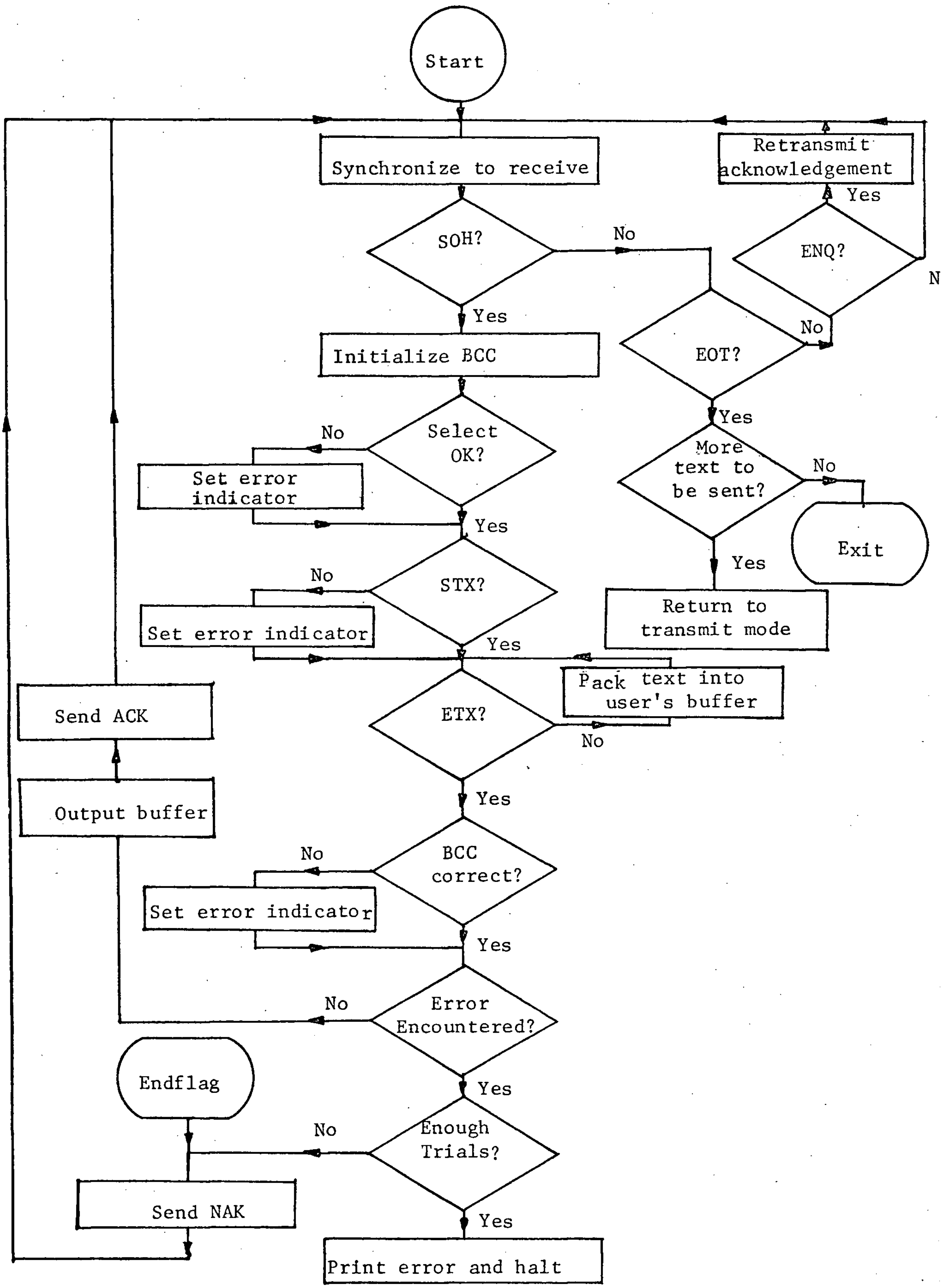
APPENDIX VI



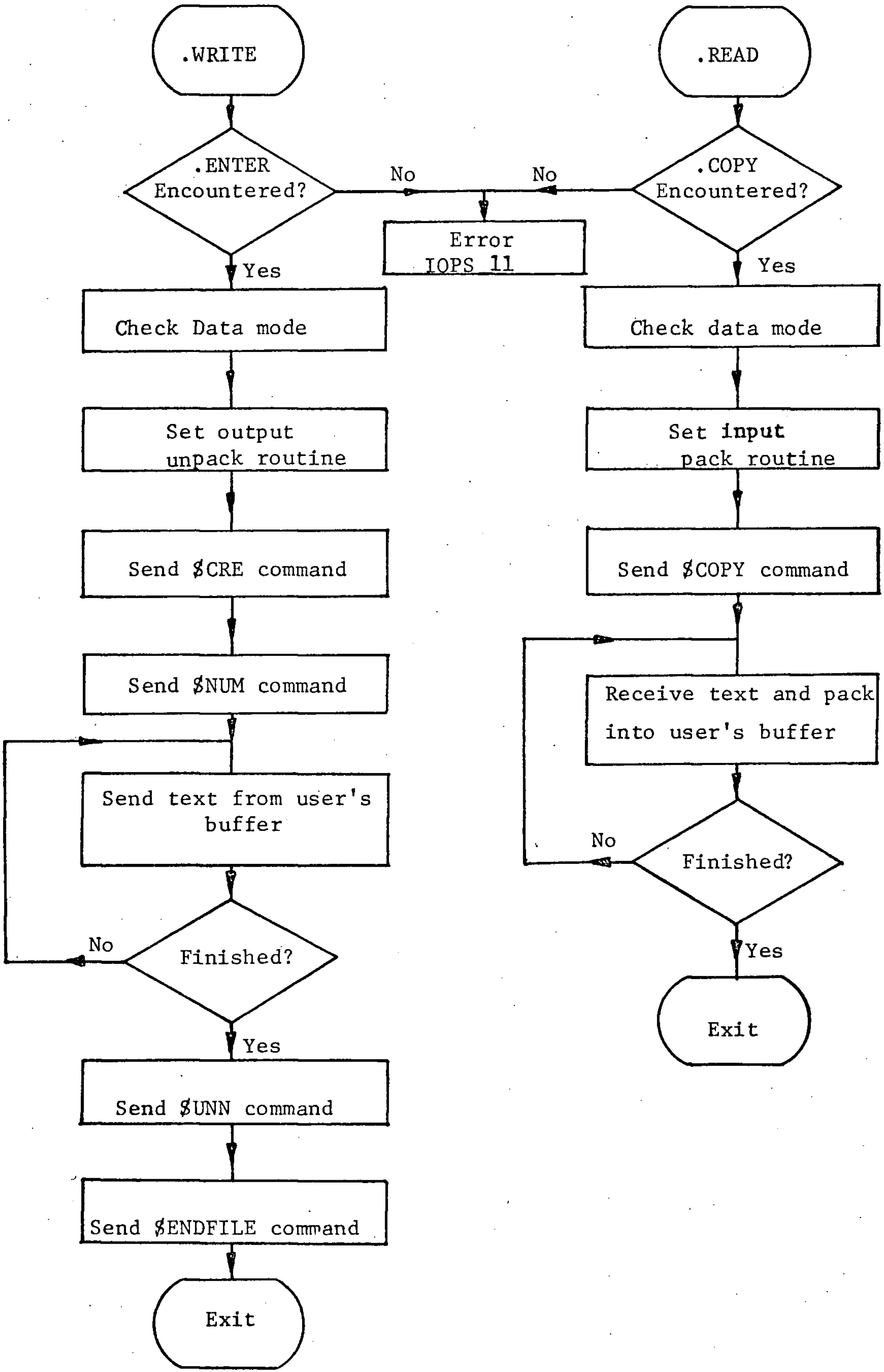
(I) The Handler Initialization Routines



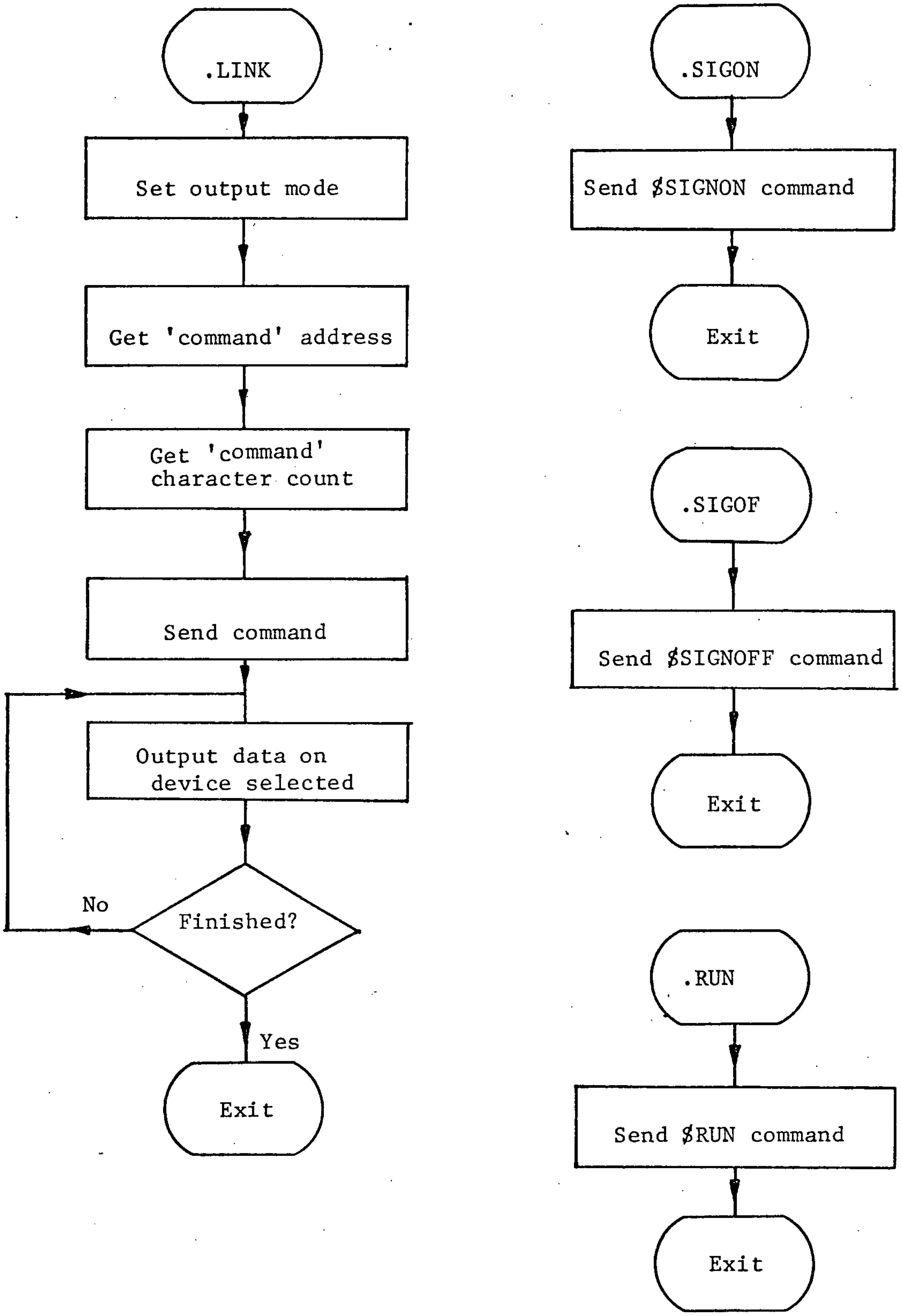
(II) Text Transmission Routine



(III) Text Reception Routine



(IV) The .WRITE and .READ Macros



(V) The .LINK, .SIGON, .SIGOF and .RUN Macros

APPENDIX VII: The Program TRSFER

7.1 Loading procedure and commands

The program TRSFER is loaded by the Linking Loader. When it is loaded, the data-phone handler (DPH.), Dectape Handler (DTA.), paper tape punch handler (PPA.) and paper tape reader handler (PRA.) are all loaded. Before loading, the user's .DAT slot devices should be assigned as follows:-

DTA1	1
DTA2	2
DPH	5
PRA	6
PPA	7
TTA	10

When TRSFER starts execution, it prints the following on the teletype

```
360-LINK
COMMAND MODE
#
```

This indicates that TRSFER is ready to accept command from the keyboard to be typed on the same line of '#' sign. The user can now use the keyboard as an ordinary remote terminal teletype by entering MTS commands. Output is printed on the teletype as well. Successful completion and readiness for the next command is again indicated by the '#' sign.

To enter file transfer mode, the command TRANSFER followed by a carriage return (ALT mode is illegal) should be entered at the keyboard. The short form T followed by carriage return is also valid. On detecting T↓ or TRANSFER↓, the following message is typed,

```
TRANSFER MODE
>
```

At this point, the user should prepare the input/output devices which are required to be ready and then type the data file transfer command.

When in command mode, if a command beginning with T but not terminated in one of the two allowed formats is encountered, the message 'WHAT?' is output on the teletype and the '#' sign is again printed. The user will then have to re-type the command.

To revert back to command mode when in transfer mode, the command C or COMMAND is typed. The messages

```
COMMAND MODE
#
```

will again be printed to indicate that it is ready to accept further commands.

In Transfer Mode, the general format of a command string is as follows:-

```
T DEV1 : FILE1 ← DEV2 : FILE2 ; EXT↓
```

where T indicates file transfers

DEV1 is the output device

DEV2 is the input device

FILE1 is the output filename

FILE2 is the input filename

and EXT is the file extension which can be source (SRC) or binary (BIN)

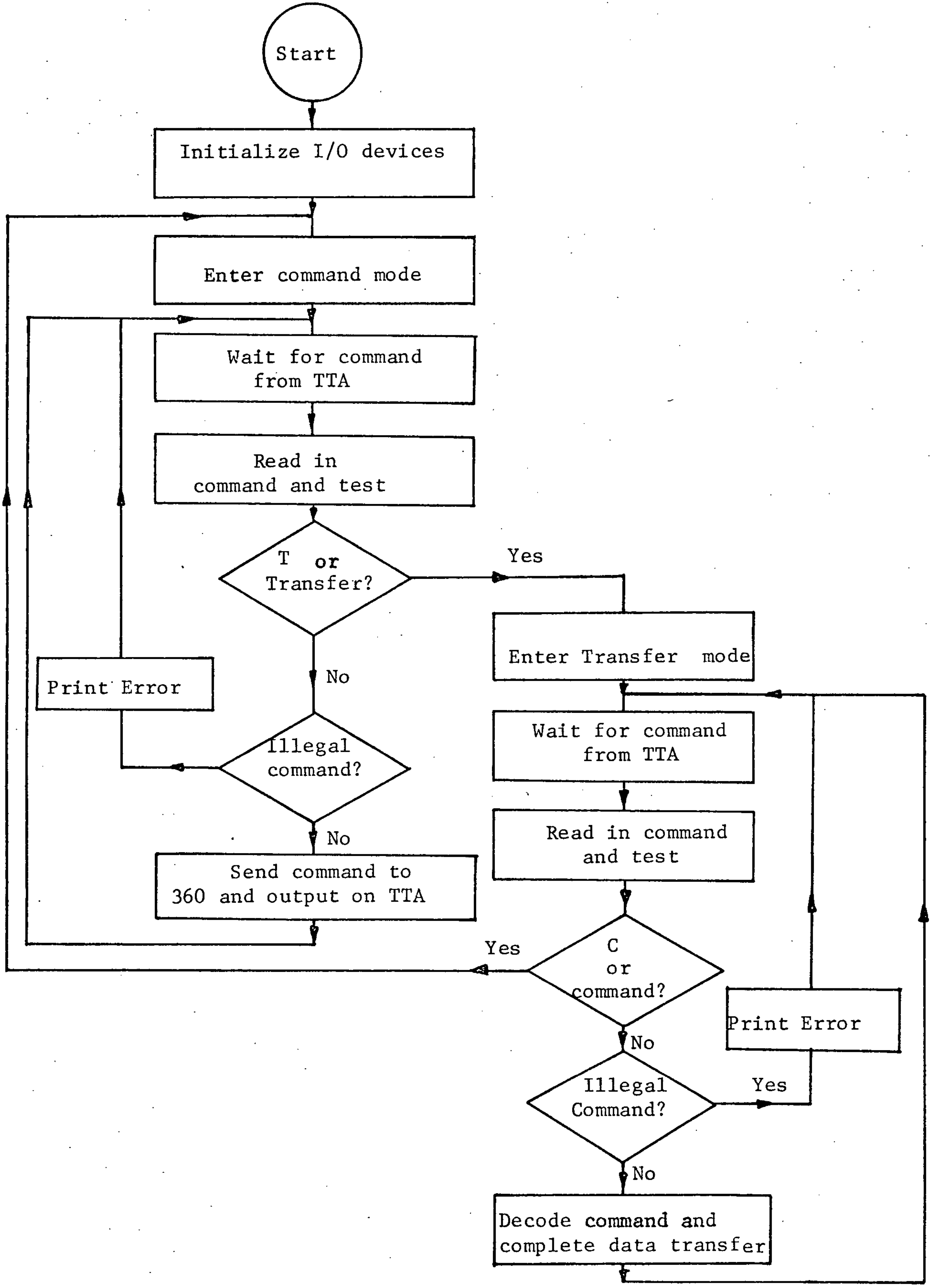
Either the output or input device has to be the IBM 360/67, which is designated as 360. It is also illegal to have both DEV1 and DEV2 as I/O devices of the PDP-9. Filename can be omitted if the device is not file-oriented. PDP-9 filenames are limited to six characters whereas 360 filenames cannot be longer than 16 characters. Spaces in between filenames and device names are ignored when encountered.

7.2 Error Messages

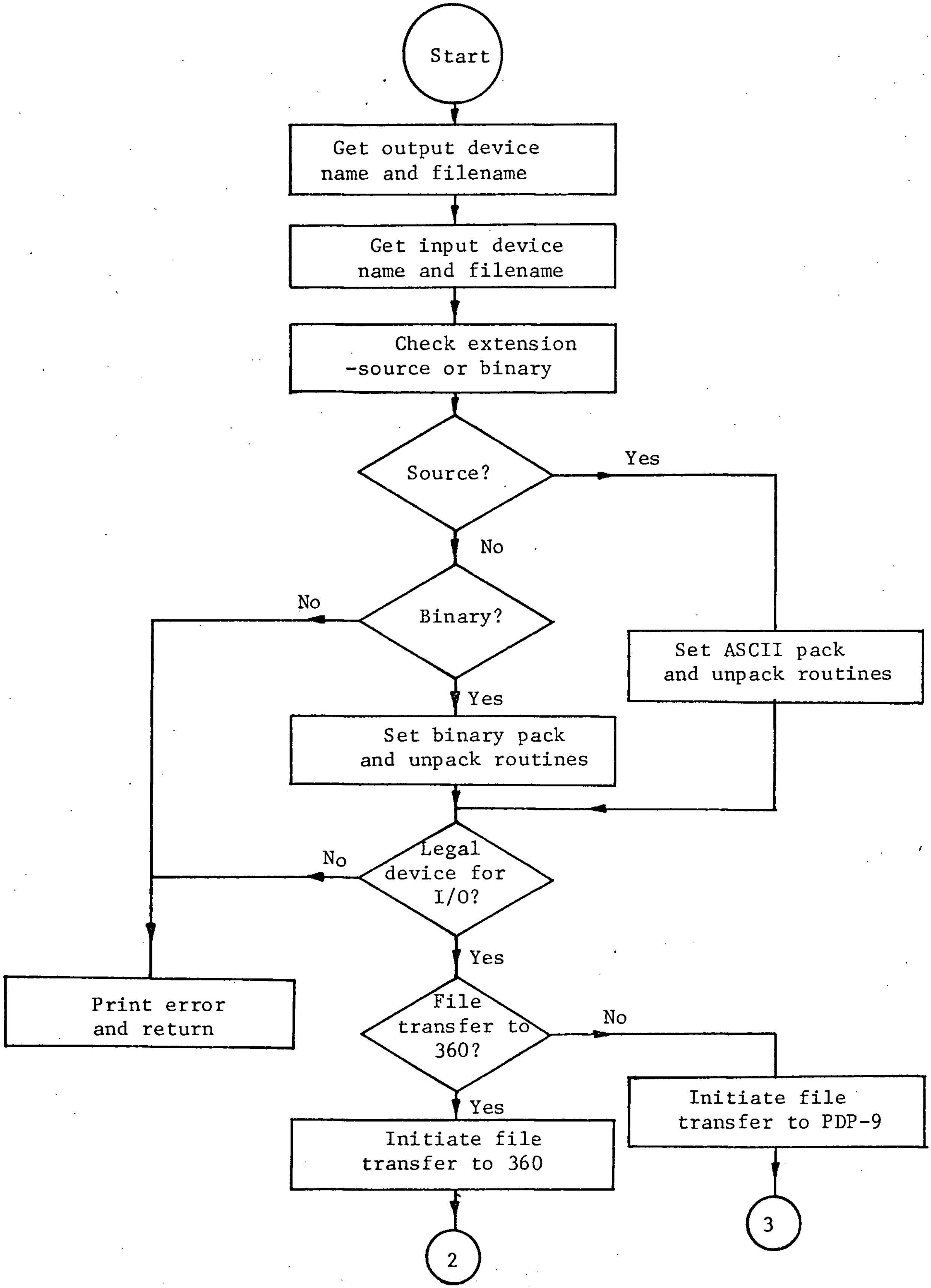
1. 'WHAT?' is printed when in command mode, a command is encountered which begins with T but is not in the form T↓ or TRANSFER ↓. It is also printed when in transfer mode, the command begins with C but is not in the form C↓ or COMMAND↓.
2. 'ILLEGAL COMMAND' is printed when a transfer mode command does not conform to the standard format.
3. 'NO OUTPUT DEVICE UNIT?' is printed if no output device unit is specified in the transfer mode command.
4. 'NO OUTPUT FILENAME?' is printed if no output filename is specified.
5. 'SAME DEVICE FOR INPUT AND OUTPUT' is printed when both input and output device units are 360.
6. 'NO INPUT DEVICE UNIT?' is printed if no input device unit is specified.
7. 'NO INPUT FILENAME?' is printed if no input filename is specified.
8. 'DATA MODE REQUIRED!' is printed if no file extension is given.
9. 'ILLEGAL DATA MODE' is printed if the file extension is neither SRC nor BIN.
10. 'DEVICE NAME TOO LONG' is printed if device name is more than 3 characters.
11. 'FILE NAME TOO LONG' is printed if filename is more than the allowed length.
12. 'ILL DEVICE UNIT' is printed when the device name is illegal or when neither the input or output device is 360. Legal device names include 360, DT1, DT2, PPA (PP), PRA(PR) and TTA (TT).

After printing the error message, the right bracket sign (or the # sign when in command mode) will again be printed. The command has to be re-entered entirely.

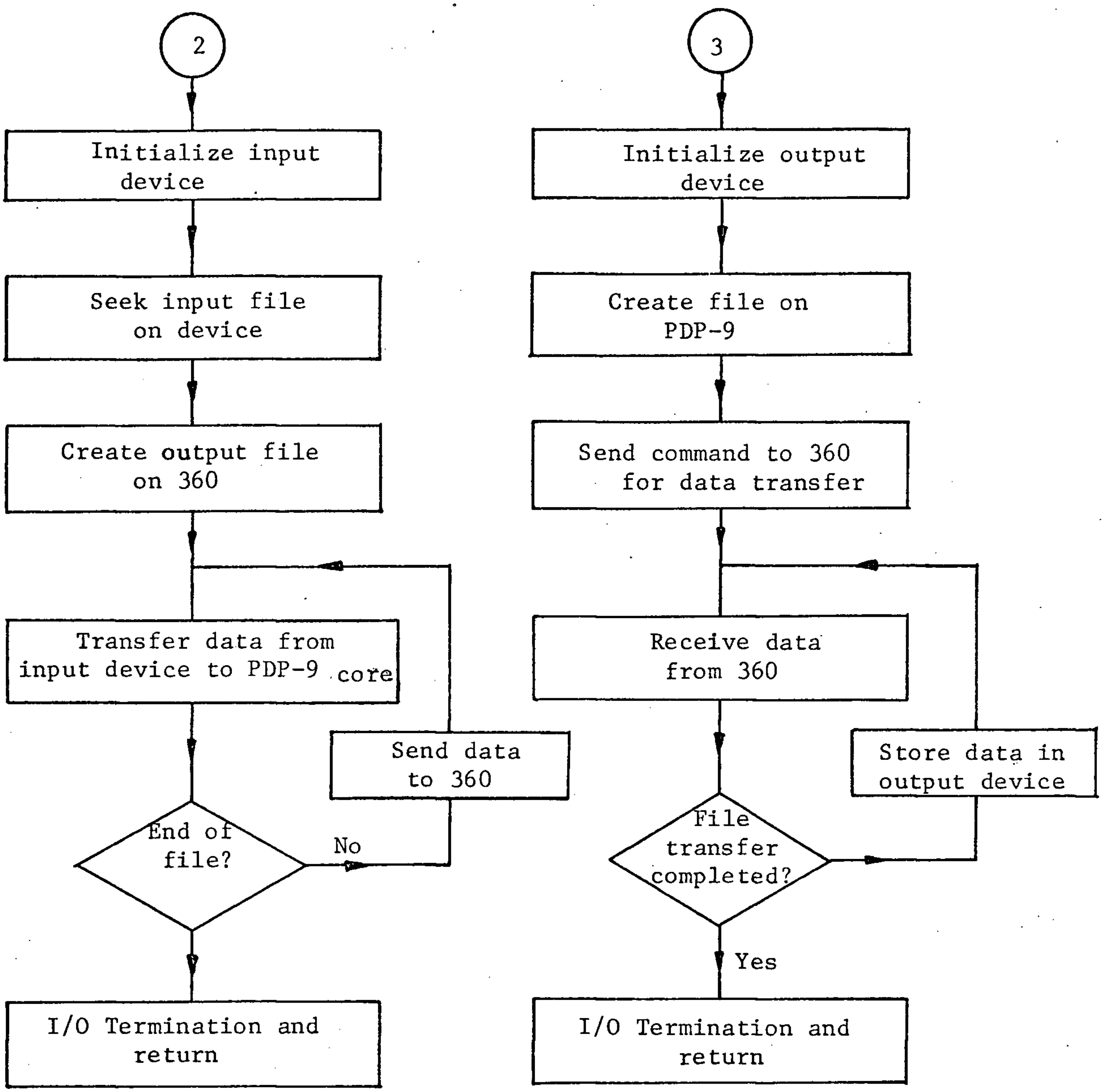
APPENDIX VIII



(I) Overall Flow-Chart for Program TRSFER


Start
Get output device name and filename
Get input device name and filename
Check extension -source or binary
Source?
Yes
No
Set ASCII pack and unpack routines
Binary?
No
Yes
Set binary pack and unpack routines
Legal device for I/O?
No
Yes
Print error and return
File transfer to 360?
No
Yes
Initiate file transfer to PDP-9
Initiate file transfer to 360
2
3

(II) Decoding file Transfer Commands


2
Initialize input device
Seek input file on device
Create output file on 360
Transfer data from input device to PDP-9 core
Send data to 360
End of file?
No
I/O Termination and return
3
Initialize output device
Create file on PDP-9
Send command to 360 for data transfer
Receive data from 360
Store data in output device
File transfer completed?
Yes
I/O Termination and return

(III) File Transfer to 360

(IV) File Transfer to PDP-9

REFERENCES

1. Proposed USA Standard Data Communication Control Procedures for the USA Standard Code for Information Exchange. Communication of the ACM. Vol. 12, No. 3, March 1969, p. 166-178.

2. EIA STANDARD RS-232-B Interface between Data Processing Terminal Equipment and Data Communication Equipment, October 1965.

3. Eisenbies, J.L., Conventions for Digital Data Communication Link Design. IBM Systems Journal, Vol. 6, No. 4, 1967, p. 267-302.

4. DPØ9A Bit-Synchronous Data Communication Channel Instruction Manual. DEC-09-H8AA-D, Digital Equipment Corporation, Maynard, Mass., U.S.A.

5. Proposed USA Standard Character Structure and Character Parity Sense for Serial-by-bit Communication in the American Standard Code for Information Interchange. Communication of the ACM. Vol. 8, No. 9, Sept. 1965, p. 553-556.

6. Lynch, W.C., Reliable Full-Duplex File Transmission over Half Duplex Telephone Lines. Comm. of the ACM, Vol. 11, No. 6, June 1968, p. 407-410.

7. PDP-9 Advanced Software System Monitors Manual DEC-9A-MAAO-D, Digital Equipment Corporation, Maynard, Mass., U.S.A.

8. Bhushan, A.K. and Stotz, R.H., Procedures and Standards for Inter-Computer Communications. AFIPS Conference Proceedings, SJCC, 1968, Vol. 32, p. 95-104.