DPV-11

DPV-11 DCLT
CVCLHB0

AH-F584B-MC
FICHE 1 OF 1

B 1

```
2222                    .TITLE  CVCLHB DPV-11 DATA COMM. LINK TEST
                        .REM &
```

IDENTIFICATION

PRODUCT CODE: AC-F582B-MC

PRODUCT NAME: CVCLHB0 DPV-11 DATA COMM. LINK TEST

PRODUCT DATE: 26-OCT-81

MAINTAINER: MERRIMACK DIAGNOSTIC ENGINEERING

AUTHOR: BRUCE RIBOLINI-BRUCE LUHRS

C 1

REVISION HISTORY:

REV	DATE	AUTHOR	REASON
A	20-AUG-80	BRUCE RIBOLINI BRUCE LUHRS	ORIGINAL ISSUE, DCLT FOR THE DPV-11
B	21-SEPT-81	ERNIE COOPER	ADD ''SET EXPECT=TRANSMIT COMMAND ADD ''EXIT'' COMMAND ADD ''RPT>'' LEVEL ADD CHECK TO INSURE TRANSMIT LIST TOTAL = EXPECT LIST TOTAL UPDATE DOCUMENTATION

TABLE OF CONTENTS

F 1

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS DCLT (DATA COMMUNICATION LINK TEST) PROGRAM IS MEANT TO PROVIDE FIELD SERVICE WITH A TOOL TO MAINTAIN DPV-11 TO DPV-11 COMMUNICATION LINKS. THIS DCLT PROGRAM WILL PROVIDE THE COVERAGE NECESSARY TO DETECT FAILURES TO THE COMPUTER EQUIPMENT, THE COMMUNICATION LINK, OR THE MODEM.

THIS DIAGNOSTIC HAS BEEN WRITTEN FOR USE WITH THE DIAGNOSTIC RUNTIME SERVICES SOFTWARE (SUPERVISOR). THESE SERVICES PROVIDE THE INTERFACE TO THE OPERATOR AND TO THE SOFTWARE ENVIRONMENT. THIS PROGRAM CAN BE USED WITH XXDP+, ACT, APT, SLIDE AND PAPER TAPE. FOR A COMPLETE DESCRIPTION OF THE RUNTIME SERVICES, REFER TO THE XXDP+ USER'S MANUAL (CHQUS?.SEQ WHERE ? IS REV. LEVEL OF THE MANUAL). THERE IS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES IN SECTION 2 OF THIS DOCUMENT.

1.2 SYSTEM REQUIREMENTS

IN ORDER TO RUN THE DPV DCLT PROGRAM, THE FOLLOWING MINIMUM HARDWARE IS REQUIRED:

- A LSI-11 CPU
- MINIMUM OF 24K WORDS OF MEMORY
- A WORKING CLOCK
- A CONSOLE TERMINAL
- ANY XXDP+ SUPPORTED LOAD MEDIA
- ONE OF THESE DPV-11 CONFIGURATIONS:

  DPV11-DB
  DPV11-DA

1.3 RELATED DOCUMENTS AND STANDARDS

- XXDP+ USER'S MANUAL (CHQUS?.SEQ WHERE ? IS THE REV. LEVEL OF THE MANUAL - "C" IS THE CURRENT REV.).

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

THE GOAL OF THE DATA COMM. LINK TEST PROGRAM IS TO TEST THE COMMUNICATION LINK AND THEREFORE ASSUMES THAT THE CPU'S, CLOCKS, AND DVP-11'S AT EACH END OF THE LINK HAVE ALREADY BEEN TESTED.

IF A WORKING CLOCK IS NOT FOUND, THE PROGRAM WILL CONTINUE BUT ANY OF THE PROGRAM THAT TIMES THE DEVICE WILL HANG IF THE DEVICE TIMES OUT. ALSO, THE EVENT LOG WILL CONTAIN A ZERO EVENT TIME FOR ALL EVENTS LOGGED.

IT IS NOT THE INTENTION OF A DATA COMM. LINK TEST PROGRAM TO TEST THE DPV-11'S, BUT TO TEST THE COMMUNICATION LINK TO WHICH THEY ARE CONNECTED.

SOME OF THE DIAGNOSTICS THAT COULD BE RUN IF EITHER OF THE DPV-11'S LOCK BAD:

CVDPVXX DPV-11 FCTNL DIAG
CXDPVXX DPV-11 DECX MODULE
XX= LATEST REVISION

1.5 ASSUMPTIONS - RESTRICTIONS

IT IS ASSUMED THAT THE COMMUNICATIONS DEVICE (A DPV-11) HAS BEEN TESTED USING THE PREREQUISTE DIAGNOSTICS. THE OPERATOR SHOULD HAVE READ THE USER DOCUMENTATION PORTION OF THE LISTING TO FAMILIARIZE HIMSELF WITH THE COMMANDS AND CAPABILITIES AVAILABLE UNDER THE DIAGNOSTIC SUPERVISOR AND DCLT.

THIS DIAGNOSTIC DOES NOT RUN THE DPV IN BIT STUFF MODE IT IS ASSUMED THAT IF THE LINK WORKS IN CHAR MODE THE LINK WILL WORK IN BIT STUFF MODE.

THE DPV11 IS NOT A DMA DEVICE AND THUS MUST RELY ON THE SOFTWARE FOR SERVICE. THEREFORE THIS DCLT WILL NOT RUN WITH THE DPV AT ITS HIGH CLOCK SPEED OF 50KHZ.

2.0 OPERATING INSTRUCTIONS

THIS SECTION CONTAINS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES.
FOR DETAILED INFORMATION, REFER TO THE XXDP+ USER'S MANUAL (CHQUS).

2.1 COMMANDS

THERE ARE ELEVEN LEGAL COMMANDS FOR THE DIAGNOSTIC RUNTIME SERVICES
(SUPERVISOR). THIS SECTION LISTS THE COMMANDS AND GIVES A VERY
BRIEF DESCRIPTION OF THEM. THE XXDP+ USER'S MANUAL HAS MORE DETAILS.

COMMAND	EFFECT
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER ^C)
PROCEED	CONTINUE FROM AN ERROR HALT
EXIT	RETURN TO XXDP+ MONITOR (XXDP+ OPERATION ONLY!)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION (IF IMPLEMENTED BY THE DIAGNOSTIC - SECTION 4.0)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE SECTION 2.3)
ZFLAGS	CLEAR ALL FLAGS (SEE SECTION 2.3)

A COMMAND CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. SO
YOU MAY, FOR EXAMPLE, TYPE "STA" INSTEAD OF "START".

2.2 SWITCHES

THERE ARE SEVERAL SWITCHES WHICH ARE USED TO MODIFY SUPERVISOR OPERATION.
THESE SWITCHES ARE APPENDED TO THE LEGAL COMMANDS. ALL OF THE LEGAL
SWITCHES ARE TABULATED BELOW WITH A BRIEF DESCRIPTION OF EACH.
IN THE DESCRIPTIONS BELOW, A DECIMAL NUMBER IS DESIGNATED BY "DDDDD".

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO BE RUN. ALL OTHER TESTS WILL NOT BE RUN.
/PASS:DDDDD	EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN SECTION 2.3.
/EOP:DDDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDDD = 1 TO 64000)
/UNITS:LIST	TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10-12 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

START/TESTS:1-5/PASS:1000/EOP:100

THE EFFECT OF THIS COMMAND WILL BE: 1) TESTS 1 THROUGH 5 WILL BE EXECUTED, 2) ALL UNITS WILL TESTED 1000 TIMES AND 3) THE END OF PASS MESSAGES WILL BE PRINTED AFTER EACH 100 PASSES ONLY. A SWITCH CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. YOU MAY, FOR EXAMPLE, TYPE "/TES:1-5" INSTEAD OF "/TESTS:1-5".

BELOW IS A TABLE THAT SPECIFIES WHICH SWITCHES CAN BE USED BY EACH COMMAND.

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBE*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXE*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
PNT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	"BELL" ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT

APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)

IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST
EVL	EXECUTE EVALUATION (ON DIAGNOSTICS WHICH HAVE EVALUATION SUPPORT)

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

SEE THE XXDP+ USER'S MANUAL FOR MORE DETAILS ON FLAGS. YOU MAY SPECIFY MORE THAN ONE FLAG WITH THE FLAG SWITCH. FOR EXAMPLE, TO CAUSE THE PROGRAM TO LOOP ON ERROR, INHIBIT ERROR REPORTS AND TYPE A 'BELL' ON ERROR, YOU MAY USE THE FOLLOWING STRING:

```
/FLAGS:LOE:IER:BOE
```

2.4 HARDWARE QUESTIONS

WHEN A DIAGNOSTIC IS STARTED, THE RUNTIME SERVICES WILL PROMPT THE USER FOR HARDWARE INFORMATION BY TYPING 'CHANGE HW (L) ?' YOU MUST ANSWER 'Y' AFTER A START COMMAND UNLESS THE HARDWARE INFORMATION HAS BEEN 'PRELOADED' USING THE SETUP UTILITY (SEE CHAPTER 6 OF THE XXDP+ USER'S MANUAL). WHEN YOU ANSWER THIS QUESTION WITH A 'Y', THE RUNTIME SERVICES WILL ASK FOR THE NUMBER OF UNITS (IN DECIMAL).

THE DPV-11 DATA COMM. LINK TEST PROGRAM WILL NOT USE MORE THAN ONE UNIT. FOR THE DPV-11 , THE HARDWARE INFORMATION REQUESTED WILL BE:

```
# UNITS (D) ? 1<CR>

UNIT 0
FULL DUPLEX OPERATION :  (L) Y ?
DPV CSR ADDRESS : (O)  160170 ?
INTERRUPT VECTOR ADDRESS:  (O)  300  ?
REMOTE NODE "ITEP" :  (L) N ?
```

THE FULL DUPLEX QUESTION SHOULD BE ANSWERED 'Y' WHEN USING FULL DUPLEX MODEMS,OR NULL MODEM, OR MODEM ELIMINATORS. ANSWER 'N' FOR HALF DUPLEX MODEMS.

REMOTE NODE ITEP SHOULD BE ANSWERED 'Y' IF OTHER NODE IS RUNNING SOFTWARE THAT IS USING 'ITEP' FORMATS(I.E. COMM TURNARROUND SYSTEM OR PDP-11 RUNNING ITEP).

2.5 DATA COMM. LINK TEST COMMANDS

THE ''DCLT>'' COMMAND LEVEL FOLLOWS THE ANSWERING OF THE HARDWARE P-TABLE QUESTIONS. THESE COMMANDS CAN BE TYPED WHEN THE ''DCLT> (A) ?'' PROMPT IS PRINTED.

MESSAGE COMMANDS AVAILABLE:

YOU ONLY HAVE TO TYPE ENOUGH CHARACTERS TO UNIQUELY SPECIFY A COMMAND.

THE COMMAND LINE IS INTERPRETED FROM LEFT TO RIGHT. THEREFORE, IF A QUALIFIER ON THE COMMAND LINE IS RELATFD OR EFFECTS A QUALIFIER TO THE LEFT ON THE COMMAND LINE, THE QUALIFIER FARTHEREST TO THE RIGHT TAKES PRECEDENCE SINCE IT IS INTERPRETED LAST. (I.E. IF /CHECK..... .../NOCHECK APPEAR ON THE SAME LINE, NOCHECK WILL BE INDICATED IN THE PARAMETERS WORD.)

REFER TO SECTION 6.0 FOR A DESCRIPTION OF THE DIFFERENT MODES OF OPERATION AND THE TYPES OF MESSAGES AVAILABLE.

2.5.1 MESSAGE COMMANDS

COMMAND		DESCRIPTION
CLEAR	EXPECTLIST	ZEROES THE EXPECTLIST (00'S) AND THEN PUTS DEFAULT ITEP MSG IN SO NOT REALLY EMPTY
CLEAR	TRANSMITLIST	FILLS TRANSMITLIST (000'S) AND THEN PUTS DEFAULT ITEP MSG IN SO NOT REALLY EMPTY
SET	EXPECTMSG=TYPE/QUAL	DEFINE A MESSAGE TO BE PUT ON THE EXPECTED LIST

WHERE: ''TYPE'' IS:
=ONES
=ZEROES
=1ALT
=0ALT
=ITEP
=CCITT
=ALPHA
=''A-Z,0-9,SPACES OR TABS IN QUOTES''

WHERE THE OPTIONAL ''QUAL'' IS:

	/SIZE=NNN	MAKE THE MESSAGE ''NNN'' BYTES LONG. (DEFAULT VALUE IS SIZE OF MESSAGE SPEC'D BY OPERATOR OR DEFAULTS.)
	/COPY=NN	COPY THIS MESSAGE INTO THE BUFFER ''NN'' TIMES (DEFAULT IS 0 = PUT THE MESSAGE IN ONLY ONCE)

NOTE: SET'S ADD MESSAGES TO THE LIST IN THE ORDER THEY'RE DEFINED. ''NNN'' IS A DECIMAL NUMBER. THE FIRST SET OVERWRITES THE DEFAULT ITEP MESSAGE PLACED THERE BY INITIALIZATION OR A ''CLEAR'' COMMAND.

SEE SECTION 6.2 FOR A DESCRIPTION OF THE PRE-DEFINED MESSAGES THAT ARE AVAILABLE. (ZEROS,ONES ...)

SET	TRANSMITMSG=TYPE/QUAL	DEFINE A MESSAGE TO BE PUT ON THE TRANSMIT LIST (SEE DESCRIPT FOR SET EXP)
SET	EXPECT=TRANSMIT	MAKES A COPY OF THE TRANSMIT LIST IN THE EXPECT LIST.
SHOW	EXPECTLIST	LISTS THE MESSAGE SIZE AND TYPE FOR THE MESSAGES IN THE EXPECT LIST
SHOW	TRANSMITLIST	LISTS THE MESSAGE SIZE AND TYPE FOR THE MESSAGES IN THE TRANSMIT LIST

2.5.2 STATISTICAL COMMANDS

COMMAND		DESCRIPTION
PRINT		TAKES THE OPERATOR TO THE REPORT LEVEL 'RPT>'. FROM HERE YOU CAN EXAMINE THE EVENT LOG.
DUMP	SSSSSS-EEEEEE/B	PRINTS THE CONTENTS OF THE MEMORY LOCATIONS BETWEEN OCTAL ADDRESSES ''SSSSSS'' AND ''EEEEEE'' WHERE ''SSSSSS'' IS THE START ADDRESS AND ''-EEEEE'' IS THE END ADDRESS.
	WHERE ''/B'' IS OPTIONAL: DEFAULT IS PRINT WORDS ''/B'' CAUSES PRINT BYTES	IF ''-EEEEEE'' IS NOT SPECIFIED THEN THE CONTENTS OF ''SSSSSS'' IS PRINTED IN WORD FORMAT.

IS PRINTED IN WORD FORMAT.

NOTE: THE DUMP COMMAND IS USEFUL FOR EXAMINING MESSAGE DATA. STARTING ADDRESSES CAN BE FOUND BY LOOKING IN THE EVENT LOG.

2.5.3 RUN COMMAND

COMMAND	DESCRIPTION
RUN MODE=MTYPE/QUAL	STARTS DCLT EXECUTING IN THE MODE SPECIFIED

NOTE: MODE=ACTIVE IS NOT DEFAULT, A MODE=MTYPE MUST BE TYPED EACH TIME A RUN IS TYPED

WHERE THE 'MTYPE' IS ANY ONE OF THE FOLLOWING:

=ACTIVE	(FORCES /NOECHO ,NO LOOPING)
=PASSIVE	(FORCES NO LOOPING)
=RECEIVE	(FORCES /NOECHO ,NO LOOPING)
=LISTEN	(FORCES /NOECHO ,NO LOOPING, /NOCHECK)
=TRANSMIT	(FORCES /NOECHO ,NO LOOPING, /NOCHECK)
=TALK	(FORCES /NOECHO ,NO LOOPING, /NOCHECK)
=DOWNLINELOAD	(DOWN-LINE-LOADING IS NOT SUPPORTED FOR DPV-11 TO DPV-11 LINKS).

(FORCING NO LOOPING MEANS IT MUST BE SPECIFIED AS A QUALIFIER ANY TIME ITS DESIRED, THERE IS NO DEFAULT)

AND OPTIONAL 'QUAL' IS ANY COMBINATION OF THE FOLLOWING:

/CHECK/NOCHECK	ENABLES/DISABLES CHECKING OF RECEIVED DATA AGAINST THE EXPECTED DATA

NOTE: IF BOTH NODES IN ACTIVE AND '/NOCHECK' IS USED, END-OF-PASS IS DEFINED AS RECEIVING 1 MESSAGE AND COMPLETING THE TRANSMIT LIST. WITH NO DATA CHECKING, THERE IS NO WAY FOR DCLT TO KNOW HOW MANY MESSAGES IT SHOULD EXPECT TO RECEIVE.

/STATUS/NOSTATUS	ENABLES/DISABLES PRINTING OF PROGRAM STATUS MESSAGES TO THE OPERATOR
/ECHO/NOECHO	ENABLES/DISABLES THE RETRANSMISSION OF THE DATA RECEIVED IN PASSIVE MODE. (IGNORED IN MODES OTHER THAN PASSIVE)
/MODEM/NOMODEM/	ENABLES/DISABLES THE REPORTING OF MODEM STATUS INTERRUPT CHANGES.
/LOOP=LTYPE	SPECIFIES WHICH, IF ANY, TYPE OF MAINTENANCE LOOPBACK IS BEING USED.

(IGNORED IN MODES OTHER THAN ACTIVE)
MUST BE SPECIFIED EACH TIME ELSE NO
LOOP IS USED.

"LTYPE" IS:

=INTERNALTTL LOOPS DATA INTERNAL TO USYNRT

=CABLE USE THIS FOR TESTING WITH H3260
OR H3259 TURNARROUND CONNECTOR

NOTE: THIS SKIPS OVER THE CHECK
FOR MODEM READY WHEN DTR IS SET.

=LOCALMODEM NOT USED BY DPV
=REMOTEMODEM " "

/PASS=NN SPECIFIES NUMBER OF ITERATIONS TO MAKE BEFORE
END-OF-PASS. DEFAULT VALUE OF 1
WILL BE USED ON ANY RUN THAT A /PASS=N
IS NOT ADDED TO THE "RUN ..." COMMAND.
IF A "-1" IS TYPED, THEN THE PROGRAM
RUN UNTIL A ^C IS TYPED.

NOTE: SEE SECTION 6.1 FOR A DESCRIPTION
----- OF THE "RUN MODES" AND "LOOP MODES"

2.5.4 DEFAULTS

IF NO "SET'S" THEN THE DEFAULT IS SAME AS IF TYPED:
SET TRANSMITMSG=ITEP/SIZE=58/COPY=0
SET EXPECTMSG=ITEP/SIZE=58/COPY=0

THE DEFAULT COPY AND SIZE FOR EACH OF THE MESSAGE TYPES:
ONES - /SIZE=64/COPY=0
ZEROES - /SIZE=64/COPY=0
0ALT - /SIZE=64/COPY=0
1ALT - /SIZE=64/COPY=0
CCITT - /SIZE=64/COPY=0
ALPHA - /SIZE=65/COPY=0
ITEP - /SIZE=58/COPY=0
OPER. SPEC'D - /SIZE=LENGTH-OF-TEXT-TYPED-BETWEEN-QUOTES/COPY=0

FOR THE RUN COMMAND THE DEFAULTS ARE:

RUN MODE=ACTIVE/NOSTATUS/CHECK/NOECHO/NOMODEM/PASS=1

NOTE: MODE=ACTIVE IS NOT DEFAULT, A MODE=MTYPE MUST BE TYPED
----- EACH TIME A RUN IS TYPED

IF THE DCLT PROGRAM IS RUN IN UNATTENDED MODE (UAM FLAG=1 OR CHAINED),
THE DEFAULTS ARE AS IF THESE SETUP AND RUN COMMANDS WERE TYPED:

```
SET TRANS=ITEP
SET EXPECT=ITEP
RUN MODE=ACTIVE/LOOP=INTERNAL/NOSTAT/NOECHO/NOMODEM/CHECK/PASS=1
```

OTHER NOTES:

^C	ALWAYS RETURNS YOU TO 'DR>'' (THE SUPERVISOR)
<CR>	IS SEEN AS A COMMAND TERMINATOR
'RUBOUT''	DELETE LAST CHAR. TYPED IN COMMAND STRING

2.5.5 PRINT COMMAND

THE PRINT COMMAND TAKES YOU TO THE REPORT LEVEL 'RPT>'.
THE COMMANDS AVAILABLE IN RPT> ARE ...

COMMAND	DESCRIPTION
HELP OR ?	PRINTS HELP INFORMATION FOR RPT>
EXIT	RETURNS YOU TO THE LEVEL THAT YOU ENTERED FROM. (DCLT> OR DR>)
LOG	PRINTS THE DCLT EVENT LOG

NOTE: AT SOME FUTURE DATE WHEN DDCMP PROTOCOL IS SUPPORTED BY THIS DCLT PROGRAM, A COMMAND WILL BE ADDED TO EXAMINE ERROR COUNTERS.

2.5.6 MISC COMMANDS

COMMANDS	DESCRIPTION
EXIT	FROM THE DCLT> LEVEL RETURNS YOU TO DR>
HELP OR ?	PRINTS HELP INFORMATION

2.6 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

1. BOOT XXDP+
2. GIVE THE DATE AND ANSWER THE LSI AND 50HZ (IF THERE IS A CLOCK) QUESTIONS
3. TYPE 'R NAME', WHERE NAME IS THE NAME OF THE BIN OR BIC FILE FOR THIS PROGRAM
4. TYPE 'START'
5. ANSWER THE 'CHANGE HW' QUESTION WITH 'Y'
6. ANSWER ALL THE HARDWARE QUESTIONS. THE NUMBER OF UNITS THAT CAN DCLT CAN USE IS ALWAYS '1'.

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING ONLY THE DEFAULTS FOR FLAGS. THESE DEFAULTS ARE DESCRIBED IN SECTION 2.3.

7. AFTER THE 'DCLT> (A) ?' PROMPT, TYPE 'RUN MODE=ACTIVE<CR>'

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING THE DEFAULT TRANSMIT AND EXPECTED MESSAGES. THE DEFAULT PASS COUNT AND 'RUN' QUALIFIERS ARE ALSO BEING USED. THESE DEFAULTS ARE DESCRIBED IN SECTION 2.5.3.

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

THERE ARE THREE LEVELS OF ERROR MESSAGES THAT MAY BE ISSUED BY A DIAGNOSTIC: GENERAL, BASIC AND EXTENDED. GENERAL ERROR MESSAGES ARE ALWAYS PRINTED UNLESS THE "IER" FLAG IS SET (SECTION 2.3). THE GENERAL ERROR MESSAGE IS OF THE FORM:

```
NAME  TYPE  NUMBER  ON UNIT NUMBER  TST NUMBER PC:XXXXXX
ERROR MESSAGE
```

,WHERE; NAME = DIAGNOSTIC NAME
TYPE = ERROR TYPE (SYS FATAL, DEV FATAL, HARD OR SOFT)
NUMBER = ERROR NUMBER
UNIT NUMBER = 0 - N (N IS LAST UNIT IN PTABLE)
TST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED
PC:XXXXXX = ADDRESS OF ERROR MESSAGE CALL

BASIC ERROR MESSAGES ARE MESSAGES THAT CONTAIN SOME ADDITIONAL INFORMATION ABOUT THE ERROR. THESE ARE ALWAYS PRINTED UNLESS THE "IER" OR "IBE" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL MESSAGE.

EXTENDED ERROR MESSAGES CONTAIN SUPPLEMENTARY ERROR INFORMATION SUCH AS REGISTER CONTENTS OR GOOD/BAD DATA. THESE ARE ALWAYS PRINTED UNLESS THE "IER", "IBE" OR "IXE" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

3.2 SPECIFIC ERROR MESSAGES

3.2.1 COMMAND LINE INTERPRETER ERRORS

ERROR MESSAGE:	MEANING
?ILL CMD-BAD SYNTX?	A COMMAND WITH AN ILLEGAL CHAR WAS TYPED - RETYPE THE COMMAND. THE VALID COMMANDS AND THEIR SYNTAX ARE SHOWN IN SECTION 2.5.
?INCMPLTE CMD?	A REQUIRD PART OF A COMMAND WAS LEFT OUT.
?NUM TOO BIG?	THE VALUE OF A NUMERIC STRING IN THE COMMAND LINE WAS LARGER THAN 65535 OR 177777 OCTAL. (> 16 BITS).
?BAD RADIX?	A "8" OR "9" WAS TYPED WHEN AN OCTAL STRING WAS EXPECTED. PROBABLY OCCURRED WHEN TYPING A "DUMP" COMMAND WHERE OCTAL ADDRESSES ARE EXPECTED.

?'LOOP'' VALID ONLY IN ACTIVE? THE ''/LOOP=..'' SWITCH WAS TYPED IN A RUN COMMAND BUT THE MODE WAS NOT SET TO ACTIVE. MAINTENANCE LOOP IS ONLY POSSIBLE IF THE MODE OF OPERATION IS ACTIVE.

?''ECHO'' VALID ONLY IN PASSIVE? THE ''/ECHO'' SWITCH WAS TYPED IN A RUN COMMAND BUT THE MODE WAS NOT SET TO PASSIVE. ECHOING OF RECEIVED DATA IS ONLY POSSIBLE IF THE MODE OF OPERATION IS PASSIVE.

?ILL CHR- ''A-Z,0-9,SP,TAB'' ONLY? A CHARACTER TYPED WITHIN QUOTES WHEN TRYING TO DEFINE THE CONTENTS OF A TRANSMIT OR EXPECT MESSAGE WAS NOT A ''A-Z,0-9,SPACE OR TAB''. RETYPE THE COMMAND WITH ONLY THESE CHARACTERS BETWEEN QUOTES.

?''SIZE=0'' NOT VALID? A MESSAGE ZERO BYTES LONG CAN NOT BE BUILT. RETYPE THE COMMAND WITH A ''/SIZE=NNN''. IF NO ''/SIZE='' IS TYPED A DEFAULT SIZE WILL BE USED.

?TRANSMIT AND EXPECT LIST MUST BE IDENTICAL FOR LOOP?

IF RUN COMMAND WITH ''/LOOP/CH'' IS TYPED THE TRANSMIT LIST AND EXPECT LIST MUST BE EQUAL. IF THEY ARE NOT THIS ERROR WILL BE DISPLAYED. USE 'SE E=T' COMMAND.

3.2.2 DCLT OR DEVICE ERROR MESSAGES:

CLOCK NOT FOUND THIS MEANS THAT NO CLOCK WAS FOUND ON THE SYSTEM THE DIAG-NOSTIC WILL STILL RUN BUT NONE OF THE TIME OUT CONDITIONS WILL OCCUR

BAD CLOCK - PROGRAM WILL HANG ON ''TIMEOUT''!!
THIS MEANS THAT THE CLOCK FOUND ON THE SYSTEM DID NOT INTERRUPT WHEN ASKED TO DO A ''TICK''.

THE PROGRAM WILL STILL RUN, BUT ANY OF THE PROGRAM THAT TIMES THE DEVICE WILL HANG IF THE DEVICE TIMES OUT. ALSO, THE EVENT LOG WILL CONTAIN A ZERO EVENT TIME FOR ALL EVENTS LOGGED.

MAX. CHAR. MSG COUNT EXCEEDED - MSG. NOT BUILT !!

THIS MEANS THAT THE TRANSMIT OR EXPECT

BUFFER IS FULL. NO MORE MESSAGES CAN BE ADDED TO THAT BUFFER.

BUFFER FULL - MSG. NOT BUILT !!

THIS MEANS THAT THE LAST MESSAGE YOU TRIED TO ADD TO EITHER THE TRANSMIT OR EXPECT BUFFER CAUSED THE TOTAL NUMBER OF MESSAGES TO BE EXCEEDED. NO MORE MESSAGES CAN BE ADDED TO THAT BUFFER. THE LIMIT IS DETERMINED BY THE SIZE OF THE MESSAGE POINTER TABLE.

CHAR. COUNT EXCEEDS BUFF LIMIT - MSG TRUNCATED

THIS MEANS THAT THE LAST MESSAGE YOU TRIED TO ADD TO THE TRANSMIT OR EXPECT BUFFER CAUSED THE TOTAL CHAR. COUNT FOR THAT BUFFER TO EXCEED THE LIMIT. THE MESSAGE WAS TRUNCATED TO COMPLETELY FILL THE BUFFER. NO MORE MESSAGES CAN BE ADDED TO THAT BUFFER.

3.2.3 DEVICE ERROR MESSAGE

DATA COMPARISON DATA ERROR
BYTE # IN MSG=XXX EXPTD=YYY RECVD=ZZZ

XXX= OFFSET OF THAT BYTE FROM THE START OF THE COMPARE OR EXPECT MESSAGE.
YYY= THE CONTENTS OF THAT BYTE IN THE EXPECTED MESSAGE
ZZZ= THE CONTENTS OF THAT BYTE IN THE RECEIVED MESSAGE

UP TO FIVE OF THESE ERRORS WILL BE PRINTED PER MESSAGE COMPARED. ONLY THE FIRST FIVE MISMATCHES WILL BE INDIVIDUALLY REPORTED, BUT TOTAL NUMBER OF MISTMATCHES IS REPORTED BY ANOTHER ERROR.

PRINTING THE EVENT LOG AND USING THE DCLT 'DUMP'' COMMAND WILL ALLOW YOU TO FIND THE ADDRESS OF THE MESSAGE AND EXAMINE IT.

DATA COMPARISON DATA ERROR
TOTAL MISMATCHES IN MSG = NNN

THIS MEANS THAT WHEN THE MESSAGE RECEIVED WAS COMPARED AGAINST THE MESSAGE THAT WAS EXPECTED, SOME OF THE CHARS. WERE NOT THE SAME.

DATA COMPARISON LENGTH ERROR

G 2

COMPARE COUNT= XXX RECEIVE COUNT= ZZZ

XXX= NUMBER OF BYTES IN THE COMPARE MESSAGE
ZZZ= NUMBER OF BYTES IN THE RECEIVED MESSAGE
THIS MEANS THAT THE MESSAGE RECEIVED WAS A DIFFENT LENGTH THEN THE MESSAGE THAT WAS EXPECTED.

MODEM STATUS CHANGES FOR THIS PASS WERE..
HARD CHANGES=XXXXX GLITCHES=XXXXX

WHERE XXXXX IS A 5 DIGIT DECIMAL NUMBER THIS MSG IS ONLY PRINTED IF NUMBER OF EITHER HARD CHANGES OR GLITCHES IS GREATER THAN 0. A HARD CHANGE IS ONE WHERE THE DPV WAS ABLE TO LATCH UP A DIFFERENCE IN THE MODEM STATUS. A GLITCH IS WHEN A MODEM STATUS INTERRUPT OCCURS BUT THE DPV CANNOT FIND A DIFFERENCE IN STATUS BIT.

****
* NOTE * - IN THE FOLLOWING ERROR DESCRIPTIONS XXXXX
******** REFERS TO THE OCTAL CONTENTS OF THE DEVICE REGISTERS SPECIFIED.

MASTER RESET DID NOT WORK		THIS MEANS THAT AFTER A MASTER
RXCSR	TXCSR	RESET WAS ISSUED TO DPV THE
XXXXXX	XXXXXXX	RXCSR REGSITER WAS NON ZERO.
NO CLEAR TO SEND FROM MODEM		WHEN REQUEST TO SEND SIGNAL
RXCSR	TXCSR	IS SET MODEM DOES NOT RESPOND
XXXXXXX	XXXXXXX	WITH CLEAR TO SEND
TIME OUT WAITING FOR RX OR TX TO COMPLETE		
RXCSR	TXCSR	DEVICE TIMED OUT AFTER
XXXXXXX	XXXXXXX	WAITNG FOR A RECIEVER OR TRANSMITTER TO COMPLETE
MODEM DID NOT RETURN MODEM READY		
RXCSR	TXCSR	MODEM DID NOT RESPOND WITH MR.
XXXXXXX	XXXXXXX	
CRC IN ERROR		THE CRC ERROR BIT IS CLEAR
RDSR	RXCSR	AT TIME THAT IT SHOULD BE SET
XXXXXX	XXXXXX	
RECEIVER OVERRUN		RECIEVER DETECTED AN OVERRUN
RDSR	RXCSR	
XXXXXXX	XXXXXXX	
TIMED OUT IN START,STACK ACK SEQ		THIS ERROR OCCURS WHEN IN THE

RDATA	SDATA	
XXXXXXX	XXXXXXX	DEVICE INIT ROUTINE A TIME OUT OCCURS WHILE TRYING TO DO START STACK ACK START UP SEQUENCE. THE VALUES IN RDATA AND SDATA INDICATE THE LAST CONTROL CHAR RECIEVED(RDATA) AND TRANSMITTED (SDATA).

4.0 PERFORMANCE AND PROGRESS REPORTS

DCLT USES IT'S OWN METHOD FOR DETERMINING AN "END OF PASS" WHICH IS CALLED A "DCLT END OF PASS". THE NUMBER OF "DCLT PASSES" TO BE RUN IS SPECIFIED BY THE "/PASS=XXX" SWITCH ON THE DCLT RUN COMMAND. THE TOTAL NUMBER OF "DCLT ERRORS" IS REPORTED WHEN " X NUMBER OF DCLT PASSES" ARE COMPLETED.

4.1 PRINTING OF EVENT LOG

SIGNIFICANT EVENTS OR CHECK-POINTS WILL BE LOGGED IN A "CIRCULAR QUEUE" STORAGE AREA CALLED THE EVENT LOG. THE LAST "N" EVENTS ARE KEPT LOGGED AND CAN BE LISTED ON THE OPERATORS CONSOLE BY GIVING A "PRINT" COMMAND AT THE "DR>"(DIAGNOSTIC SUPERVISOR) OR "DCLT>" (DCLT) LEVEL. THIS WILL TAKE YOU TO THE RPT> LEVEL. NOW INPUT THE "LOG" COMMAND. THE EVENTS ARE PRINTED IN A "LAST-IN FIRST-OUT" ORDER.

EVENT TIME IS TYPED OUT AS MMM:SS:TT (LIKE 254:36:07) WHERE MMM,SS,TT REPRESENT THE NUMBER OF MINUTES, SECONDS, CLOCK TICKS SINCE THE LAST START OR RESTART. IT SHOULD BE NOTED THAT THE TIMES ARE RELATIVE SINCE WHILE THE PROCESSOR IS RUNNING AT PRIORITY 7 THE CLOCK CAN'T INTERRUPT TO KEEP TIME. THIS IS THE CASE WHILE THE PROGRAM IS FETCHING DCLT COMMANDS FROM THE OPERATOR. IT SHOULD ALSO BE NOTED THAT THERE ARE ONLY 8 BITS AVALIABLE TO STORE RELATIVE MINUTES SO "TIME" WILL WRAP TO 000:00:00 AFTER 256:59:59.

A START OR RESTART COMMAND AT THE "DR>" LEVEL INITIALIZES THE EVENT LOG. THEREFORE IT IS WISE TO DO A "PRINT" AT THE "DR>" LEVEL BEFORE GIVING A "START" OR "RESTART".

THE TYPES OF EVENTS KEPT IN THE EVENT LOG ARE:

TRANSMIT MESSAGE QUEUED:
        EVENT TIME, ADDRESS OF 1ST BYTE OF MESSAGE,
        TOTAL NO. OF BYTES, MODEM STATUS AT THAT TIME.
TRANSMIT MESSAGE COMPLETED:
        EVENT TIME, ADDRESS OF 1ST BYTE OF MESSAGE,
        TOTAL NO. OF BYTES, MODEM STATUS AT THAT TIME.
RECEIVE SPACE QUEUED:
        EVENT TIME, ADDRESS OF 1ST BYTE OF MESSAGE,
        TOTAL NO. OF BYTES, MODEM STATUS AT THAT TIME.
RECEIVE MESSAGE COMPLETED:
        EVENT TIME, ADDRESS OF 1ST BYTE OF MESSAGE,

TOTAL NO. OF BYTES, MODEM STATUS AT THAT TIME.

DATA COMPARISON STARTED:
EVENT TIME, ADDRESS OF 1ST BYTE OF RECEIVED MSG.,
TOTAL NO. OF BYTES IN RCV. MSG., TOTAL NO. OF BYTES
IN EXPECT MSG.

DATA COMPARISON DATA ERROR:
EVENT TIME, ADDRESS OF 1ST BYTE OF RECEIVED MSG.,
TOTAL NO. OF BYTES IN RCV. MSG., TOTAL NO. OF
COMPARISON FAILURES

DATA COMPARISON LENGTH ERROR:
EVENT TIME, ADDRESS OF 1ST BYTE OF RECEIVED MSG.,
TOTAL NO. OF BYTES IN RCV. MSG., TOTAL NO. OF BYTES
IN EXPECT MSG.

DEVICE INIT AND SETUP:
EVENT TIME, MODE OF OPERATION, TYPE OF MAINTENANCE
LOOP, "DCLT" PASS COUNT, "RUN" PARAMETERS

DEVICE ERROR:
EVENT TIME, DEVICE ERROR MESSAGE, CONTENTS OF TWO
REGISTERS RELATING TO THE ERROR.

END OF PASS:
EVENT TIME, "DCLT" PASS COUNT, "DCLT" ERROR COUNT,
AND THE "STRT-TO"(COUNT OF START TIME OUTS).

4.2 OPERATOR STATUS MESSAGES

THE "/STATUS, /NOSTATUS" QULAIFIERS FOR THE DCLT "RUN" COMMAND ENABLES/DISABLES THE PRINTING OF PROGRAM STATUS MESSAGES TO THE OPERATOR. THESE MESSAGES ARE INTENDED TO TELL THE OPERATOR WHAT THE DCLT PROGRAM IS CURRENTLY DOING. BELOW ARE THE MESSAGES THAT MIGHT BE PRINTED AND THEIR MEANING:

MESSAGE	MEANING
TXQ	DEVICE IS ABOUT START TRANSMITING A MESSAGE
TXC	TRANSMISSION OF MESSAGE COMPLETED
RXQ	DEVICE HAS QUEUED SPACE TO RECEIVE/ COMPLETED RECEIVE
ERR	DEVICE ERROR HAS OCCURRED
INI	DEVICE ABOUT TO BE INITIALIZED
MSC	ABNORMAL MODEM STATUS CHANGE
CMP	ABOUT TO DO DATA CHECKING OF RECVD VS. EXPTD DATA
CML	LENGTH ERROR OCCURRED DURING DATA COMPARISON
CMD	DATA ERROR OCCURRED DURING DATA COMPARISON
EOP	END OF PASS

5.0 DEVICE INFORMATION TABLES

THIS IS THE DEFAULT HARDWARE P-TABLE. THE VALUES AND
SIZE ARE USED AS A ''TEMPLATE'' FOR CREATING ACTUAL P-TABLE
ENTRIES AND THE DEFAULT VALUES PROVIDED FOR THE OPERATOR.
SEE SECTION 2.4 FOR AN EXAMPLE OF THE HARDWARE QUESTIONS.

THE NUMBERS IN BRACKETS (I.E. [10]) INDICATES THE OFFSET OF THE
WORD INTO THE HARDWARE P-TABLE. THE OFFSETS MUST MATCH THE P-TABLE
OFFSETS USED IN THE HARDWARE PARAMETER CODING SECTION WHERE THE
''GET PARAMETER'' CALLS ARE USED TO FILL THE P-TABLE.

```
.WORD   1               ;[0] FULL OR HALF DUPLEX FLAG (BIT0=1 IF FULL)
.WORD   160170          ;[2] CSR ADDRESS
.WORD   300             ;[4] INTERRUPT VECTOR
.WORD   240             ;[6]  SPARE
.WORD   0               ;[10] SPARE
.WORD   0               ;[12] SPARE
.WORD   0               ;[14] REMOTE NODE ''ITEP''
```

6.0 MODE AND MESSAGE DESCRIPTIONS

6.1 MODE DESCRIPTIONS

THE FOLLOWING MODE DESCRIPTIONS REFER TO MESSAGE LISTS BEING
TRANSMITTED AND RECIEVED BE AWARE THAT OTHER DATA IS ALSO
SENT WITH THE MESSAGE. MESSAGE FORMATS ARE..... 177(OCTAL)SYNC
CHARS;SOH CHAR(201),BYTE COUNT OF MSG.(2 BYTES);THREE BYTES OF
OCTAL 001;2 BYTES OF CRC;THE MESSAGE BODY;2BYTES OF CRC.
IF REMOTE NODE ITEP QUESTION IS ANSWERED Y THEN SYNC CHAR WILL
BE OCTAL 26 INSTEAD OF 226;NO SOH,BYTE COUNT OR HEADER DATA WILL
BE SENT. ON FULL DUPLEX LINKS A START STACK ACK SEQ WILL BE SENT
ONCE AT INIT TIME FOR EACH MODE. THIS SEQUENCE CONTAINS CONTROL
MESSAGES WHOSE FORMAT IS ..8 SYNC CHAR, 1BYTE ENQ(005),START
STACK OR ACK,3 BYTES OF 001, AND 2BYTES OF CRC.

6.1.1 TRANSMIT MODE

A LIST OF MESSAGES IS TRANSMITTED WITHOUT EXPECTING ANY DATA
TO BE RECEIVED.

6.1.2 RECEIVE MODE

SPACE IS QUEUED FOR THE DEVICE TO RECEIVE MESSAGES.
AFTER RECEIVING AN 'EXPECTED'' NUMBER OF MESSAGES, THE DATA RECEIVED
CAN BE COMPARED AGAINST A LIST OF 'EXPECT TO RECEIVE'' MESSAGES
IF DATA-CHECKING IS ENABLED.

6.1.3 PASSIVE MODE

THEN EVERY TIME A MESSAGE IS RECEIVED, A MESSAGE IS TRANSMITTED.
DATA CHECKING CAN BE DONE ON THE RECEIVED DATA.THE ''/ECHO, /NOECHO''
ENABLES/DISABLES THE RETRANSMISSION OF THE DATA RECEIVED.

6.1.4 ACTIVE MODE

A LIST OF MESSAGES IS TRANSMITTED AND MESSAGES ARE RECEIVED.
AFTER RECEIVING AN 'EXPECTED'' NUMBER OF MESSAGES, THE DATA RECEIVED
CAN BE COMPARED AGAINST A LIST OF ''EXPECT TO RECEIVE'' MESSAGES
IF DATA-CHECKING IS ENABLED.

NOTE: IF BOTH ENDS OF THE LINK ARE IN ACTIVE MODE, THEN THE
LINK MUST BE A FULL DUPLEX LINK!

6.1.5 DOWN-LINE-LOAD

DOWN-LINE-LOADING IS NOT SUPPORTED FOR DPV-11 TO DPV-11 LINKS.

6.1.6 TALK MODE

THE ''TALK'' END OF THE LINK TRANSMITS OPERATOR-TYPED MESSAGES
UNTIL A ''EXIT'' MESSAGE IS TYPED. AT THAT POINT, THE NODE GOES
INTO ''LISTEN'' MODE. AN ''EXIT MESSAGE'' IS A MESSAGE WHOSE FIRST
FOUR CHARACTERS ARE ''EXIT''. SINCE ONLY THE FIRST FOUR CHARACTERS
NEED TO BE ''EXIT'', MORE CHARACTERS CAN BE ADDED SO THAT A MESSAGE
MAY BE SENT AND THE MODE SWITCHED ALL AT ONCE. FOR EXAMPLE:

TLK> EXIT ALL OF THIS LINE IS SENT THEN MODE SWTICHED

6.1.7 LISTEN MODE

THE ''LISTEN'' END OF THE LINK PRINTS ALL OF THE MESSAGES
RECEIVED BY THE DEVICE ON THE OPERATOR'S CONSOLE. IF THE MESSAGE
RECEIVED IS AN ''EXIT'' MESSAGE, THEN THE NODE ENTERS ''TALK'' MODE.
AN ''EXIT MESSAGE'' IS A MESSAGE WHOSE FIRST FOUR CHARACTERS ARE ''EXIT''.

6.1.8 MAINTENANCE ''LOOP'' MODES

REMEMBER THAT THE WHENEVER A ''RUN'' COMMAND IS TYPED, THE DEFAULT IS NO LOOPBACK AND THAT A LOOP MODE MUST BE SPECIFIED BY A ''/LOOP=..'' IF A LOOP MODE IS DESIRED.
LOOP MODES ARE ONLY VALID IF THE MODE TO RUN IS ACTIVE !

INTERNALTTL LOOPS DATA INTERNAL TO THE USYNRT

THE FOLLOWING TABLE SUMMARIZES THE MODES THAT CAN BE RUN TOGETHER WHEN THE DCLT PROGRAM IS RUNNING ON TWO PROCESSORS (ONE AT EACH END OF THE LINK):

HALF DUPLEX START	STATION A ''HOST'' NODE	''/LOOP'' ALLOWED?	STATION B ''REMOTE'' NODE	DUPLEX
B	TALK	NO	LISTEN*, RECEIVE	HALF OR FULL
A	LISTEN	NO	TALK*, TRANSMIT	HALF OR FULL
B	TRANSMIT	NO	RECEIVE*, LISTEN	HALF OR FULL
A	RECEIVE	NO	TRANSMIT*, TALK	HALF OR FULL
A	PASSIVE	NO	ACTIVE*	HALF OR FULL
-NA-	ACTIVE	YES	ACTIVE*	FULL
B	ACTIVE	YES	PASSIVE*	HALF OR FULL
-NA-	DOWNLINELOAD	** DOWN-LINE-LOADING IS NOT SUPPORTED FOR DPV-11 TO DPV-11 LINKS.		

*= MOST LIKELY TO BE IN THAT MODE

NOTE: H/D START COLUMN INDICATES WHICH NODE TO START FIRST ON A HALF DUPLEX LINK

6.2 MESSAGE DESCRIPTIONS

NAME	DESCRIPTION
ZEROES	MESSAGE OF ALL 0'S (00000000,00000000,00000000,....)
ONES	MESSAGE OF ALL 1'S (11111111,11111111,11111111,....)
1ALT	MESSAGE OF ALTERNATING 1'S (10101010,10101010,.....)
0ALT	MESSAGE OF ALTERNATING 0'S (01010101,01010101,.....)
CCITT	''CCITT'' 512-BIT (VS. 511 BITS) TEST PATTERN
ITEP	''INTERPROCESSOR TEST PROGRAM'S (ITEP)'' MESSAGE 1(DP1:) (<177><177>/$A THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.<15><12><001><177><177><177><177>)
ALPHA	ALPHA-NUMERICS (OR FUTURE COMM TURNAROUND MSG) (#$!'' (AMPERSAND)'()*+,-.0123456789:;<=>?@ABCDEFGHIJK LMNOPQRSTUVWXYZ/[\]^_%)

OPERATOR-SPECIFIED ''A-Z,0-9,SPACES,TABS''
THESE ARE THAT THE CHARACTERS THAT CAN
BE TYPED BETWEEN QUOTATION MARKS (''..'')
TO SPECIFIY A UNIQUE MESSAGE.

7.0 OTHER INFORMATION

7.1 INTERFACING TO AN ''ITEP'' NODE

THESE ARE THE RULES WHEN USING ITEP/WITH A DUP TO TALK
TO A DPV USING DCLT.

ITEP NODE	DCLT NODE
ANSWER ALL QUESTION TO THE SET SWITCHES PROMPT.	ANSWER ALL QUESTIONS TO THE DCLT> PROMPT.
FOR ONE WAY OUT... SET SWITCHES TO 1221	CLEAR EXPECTED SET E=ITEP/S=56 RUN MODE=REC/STATUS/CHECK
NOTE: DUP ITEP SENDS ONLY 56 CHARS	
FOR ONE WAY IN.....	
SET SWITCHES TO 1222	RUN MODE=TRA/STATUS
FOR EXTERNAL LOOPBACK....	
SET SWITCHES.....1224	CLEAR EXPECTED SET EXP=ITEP/S=56 RUN MODE=ACTIVE/STATUS/CHECK
FOR INTERNAL LOOPBACK.....	
SET SWTICHES.....1260	CLEAR EXPECTED SET EXP=ITEP/S=56 RUN MODE=ACTIVE/STATUS/CHECK

NOTE: DO NOT USE SWITCH 8 WITH ITEP GOING TO DCLT
THE ONLY MESG. DCLT SUPPORTS IS MSG 1.

DCLT IGNORES CRC ERRORS WHEN REC DATA FROM ITEP
BECAUSE ITPE SENDS NO CRC.

7.2 TROUBLESHOOTING HINTS

LISTED BELOW ARE SOME SETUPS THAT COULD BE USED FOR ISOLATING FAULTS.
THESE ARE BY NO MEANS THE ONLY WAYS DCLT CAN BE USED !!!!!!!
DCLT IS MEANT TO BE A VERY FLEXIBLE TOOL! THIS SECTION IS MEANT TO
GIVE SOMEONE NOT TOO FAMILIAR WITH DCLT A PLACE TO START.

REMEMBER THAT THE PRINTING OF STATUS MESSAGES AND PRINTING OF THE
EVENT LOG CAN PROVIDE A LOT OF INFORMATION ABOUT THE SEQUENCE OF
EVENTS AND HOW THE DEVICE AND LINK ARE BEHAVING.

NOTE: IF BOTH NODES IN ACTIVE AND "/NOCHECK" IS USED,
----- END-OF-PASS IS DEFINED AS RECEIVING 1 MESSAGE
AND COMPLETING THE TRANSMIT LIST. WITH NO DATA
CHECKING, THERE IS NO WAY FOR DCLT TO KNOW HOW
MANY MESSAGES IT SHOULD EXPECT TO RECEIVE.

7.2.1 INTERNAL LOOP AT EACH NODE

RUN EACH END OF THE LINK IN ACTIVE MODE WITH LOOP=INTERNAL.
TRANSMIT TWO OR THREE MESSAGES WITH NO DATA CHECKING.
STATUS PRINTING COULD BE TURNED OFF IF ON, BUT SEEING THE SEQUENCE
OF EVENTS MIGHT BE INFORMATIVE.

A POSSIBLE COMMMAND SEQUENCE IS:

```
C E
C T
SE T=ONES/S=20/C=2
R M=A/LO=I/NOCH/STAT
```

WHAT THE ABOVE COMMAND SEQUENCE MEANS:

THE "C E" AND THE "C T" INITIALIZES THE "EXPECT"
LIST AND THE "TRANSMIT LIST". THE "SE T=ONES/S=20/C=2"
SETS THE TRANSMIT LIST TO CONTAIN 3 MESSAGES. THE MESSAGES
CONTAIN DATA OF ALL ONES AND EACH ONE IS 20 BYTES IN LENGTH.
THE "R M=A/LO=I/NOCH/STAT" SETS THE MODE TO RUN IN TO BE
ACTIVE AND LOOP TYPE TO BE INTERNAL TTL. THE PROGRAM WILL
NOT BE CHECKING DATA SO THERE WAS NO NEED TO SET UP AN
EXPECT LIST. THE PROGRAM WILL BE PRINTING STATUS MESSAGES.

WHAT YOU SHOULD SEE AFTER ENTERING THE RUN COMMAND
IF THINGS ARE RUNNING CORRECTLY :

```
INI RXQ TXQ RXQ TXC TXQ RXQ TXC
TXQ RXQ TXC EOP
     MODE=ACTIVE/LOOP=INTERNAL/PASS=00000
                 /STATUS/NOCHECK/NOECHO/NOMODEM
DCLT> (A)  ?
```

THIS GIVES YOU A IDEA IF THE COMM. DEVICE CAN EVEN TRANSMIT AND
RECEIVE. ANY ERRORS REPORTED WILL PROBABLY BE DUE TO INCORRECT
DEVICE ADDRESSES BEING USED OR A FAULTY DEVICE. CHECK ADDRESSES
WITH "DISPLAY" AND RUN THE PREREQUISITE DIAGNOSTICS FOR THE COMM.
DEVICE.

NOW TRY RUNNING EACH NODE THE SAME WAY WITH DATA CHECKING ENABLED.
A POSSIBLE COMMMAND SEQUENCE IS:

```
SE E=T
R M=A/LO=I/CH/PAS=3
```

WHAT THIS SEQUENCE MEANS:

THIS SEQUENCE IS SIMILAR TO THE ONE ABOVE . THE "SE E=T"
MAKES A COPY OF THE TRANSMIT LIST IN THE EXPECT LIST.
THE EXPECT LIST NOW CONTAINS 3 MESSAGES. THE MESSAGES WILL
HAVE ALL ONES FOR DATA AND BE 20 BYTES EACH IN LENGTH.
THE RUN COMMAND IS THE SAME WITH THE ADDITION OF TWO
SWITCHES "/CH/PAS=3". THE "CH" SWITCH TELLS THE PROGRAM
TO CHECK THE RECEIVED DATA AGAINST THE "EXPECTED LIST".
THE "PAS=3" SWITCH TELLS THE PROGRAM TO RUN 3 PASSES BEFORE
RETURNING TO THE DCLT> PROMPT.

WHAT YOU SHOULD SEE AFTER ENTERING THE RUN COMMAND
IF THINGS ARE RUNNING CORRECTLY :

```
INI RXQ TXQ TXC RXQ TXQ TXC RXQ
TXQ TXC CMP CMP CMP EOP RXQ TXQ
TXC RXQ TXQ TXC RXQ TXQ TXC CMP
CMP CMP EOP RXQ TXQ TXC RXQ TXQ
TXC RXQ TXQ TXC CMP CMP CMP EOP
     MODE=ACTIVE/LOOP=INTERNAL/PASS=00000
                 /STATUS/CHECK/NOECHO/NOMODEM
```

IF A CABLE TURNAROUND CONNECTOR IS AVAILABLE, PUT IT ON THE END OF
THE CABLE JUST BEFORE THE MODEM OR IF A H3260(RS-423) ON BOARD
CONNECTOR IS AVAILABLE INSTALL IT AND RUN IN ACTIVE MODE WITH THE
"/LOOP=CABLE" SWITCH.
POSSIBLE COMMAND SEQUENCE IS:

```
R M=A/L=C/CH/PAS=3
```

WHAT THIS SEQUENCE MEANS:

THIS SEQUENCE HAS THE "/LO=C". THIS INFORMS THE SOFTWARE
NOT TO CHECK FOR DATA SET READY SIGNAL FROM THE MODEM.

WHAT YOU SHOULD SEE AFTER ENTERING THE RUN COMMAND

IF THINGS ARE RUNNING CORRECTLY :

```
INI RXQ TXQ TXC RXQ TXQ TXC RXQ
TXQ TXC CMP CMP CMP EOP RXQ TXQ
TXC RXQ TXQ TXC RXQ TXQ TXC CMP
CMP CMP EOP RXQ TXQ TXC RXQ TXQ
TXC RXQ TXQ TXC CMP CMP CMP EOP
     MODE=ACTIVE/LOOP=CABLE/PASS=00000
                 /STATUS/CHECK/NOECHO/NOMODEM
DCLT> (A)  ?
```

7.2.2 TRANSMIT ON ONE NODE RECEIVE ON THE OTHER

NOW TRY TRANSMIITING FROM ONE END AND RECEIVING ON THE OTHER. MAYBE WITH NO DATA CHECKING AT FIRST TO ESTABLISH IF THE LINK IS WORKING. POSSIBLE COMMAND SEQUENCES ARE:

NODE A	NODE B
C E	C E
C T	C T
SE T=1ALT/S=250	R M=R/NOCH/PAS=3
R M=TR/PAS=3	

WHAT THIS SEQUENCE MEANS:

THE ''C E '' AND ''C T'' INITIALIZE BOTH THE TRANSMIT AND EXPECT LISTS. THE ''SE T=1ALT/S=250'' SETS THE TRANSMIT LIST ON NODE A TO BE 1 MESSAGE WITH A LENGTH OF 250 BYTES AND DATA OF ALTERNATING ONES AND ZEROS. THE ''R M=TR/PAS=3'' SETS THE RUN MODE OF NODE A TO BE TRANSMIT AND THE PASS COUNT IS SET TO 3. THE ''R M=R/NOCH/PAS=3'' SETS THE RUN MODE OF NODE B TO BE RECEIVE, NO DATA CHECKING IS TO BE DONE,AND THE PASS COUNT IS SET TO THREE.

WHAT YOU SHOULD SEE AFTER ENTERING THE RUN COMMAND
IF THINGS ARE RUNNING CORRECTLY :

FOR NODE A:

```
INI TXQ TXC EOP TXQ TXC EOP TXQ
TXC EOP
     MODE=TRANSMIT/PASS=00000
                 /STATUS/NOCHECK/NOECHO/NOMODEM
DCLT> (A)  ?
```

FOR NODE B:

```
INI RXQ EOP RXQ EOP RXQ EOP
     MODE=RECEIVE/PASS=00000
                 /STATUS/NOCHECK/NOECHO/NOMODEM
DCLT> (A)  ?
```

NOW TRY DOING DATA CHECKING ON THE MESSAGE(S) BEING
TRANSMITTED. POSSIBLE COMMAND SEQUENCES ARE:

```
                                        SE E=1ALT/S=250
        R M=TR/PAS=3                    R M=R/CH/PAS=3
```

WHAT THIS SEQUENCE MEANS:
THE ''SE E=1ALT/S=250'' LINE MUST BE ADDED HERE
TO SET UP THE ''EXPECT LIST'' ON THE RECEIVE NODE
SO IT WILL KNOW WHAT TO COMPARE AGAINST.
THE CHANGE IN THE RUN COMMAND IS FROM ''NOCH'' TO
''CH''. THE ''CH'' ENABLES DATA CHECKING.

WHAT YOU SHOULD SEE AFTER ENTERING THE RUN COMMAND IF THINGS
ARE RUNNING CORRECTLY:

NODE A: IS THE SAME AS ABOVE.

NODE B:

```
        INI RXQ CMP EOP RXQ CMP EOP RXQ CMP EOP
         MODE=RECEIVE/PASS=00000
                /STATUS/CHECK/NOECHO/NOMODEM
        DCLT> (A)?
```

NOW RUN THRU THE SEQUENCE AGAIN WITH NODE A RECEIVING
AND NODE B TRANMITTING TO CHECK OUT THE OPPOSITE
DIRECTION OF DATA FLOW.

7.2.3 ONE NODE ACTIVE THE OTHER NODE PASSIVE

NOW TRY RUNNING ONE NODE IN ACTIVE MODE WHILE THE OTHER
END RUNS IN PASSIVE. DATA CHECKING SHOULD BE TURNED OFF
IF THE MESSAGE LISTS ARE NOT THE SAME.
POSSIBLE COMMAND SEQUENCES ARE:

NODE A	NODE B
C E	C E
C T	C T
SE T=CCITT/S=10/C=2	SE T=1ALT/S=20/C=2
R M=ACT/NOCH/PAS=3	R M=P/NOCH/PAS=3

WHAT THIS SEQUENCE MEANS:
THE EXECUTION OF THIS SEQUENCE CAUSES THE FOLLOWING
THINGS TO HAPPEN ON NODE A. THE TRANSMIT AND EXPECT
LISTS ARE INITIALIZED THEN THE TRANSMIT LIST IS SET
TO 3 MESSAGES OF 10 BYTES EACH. THE DATA USED IN THE
TRANSMIT MESSAGES IS THE CCITT PATTERN. THEN NODE A
IS RUN IN ACTIVE MODE WITH DATA CHECKING DISABLED AND
THE PASS COUNT SET TO THREE. NOTE STATUS WOULD STILL BE
PRINTED IF THE PREVIOUS SEQUENCES HAD BEEN RUN.
IF YOU ARE RUNNING FROM LOAD TIME YOU WOULD HAVE
TO ADD A ''/STA TO THE RUN COMMAND LINE.

NODE B: THE TRANSMIT AND EXPECT LISTS ARE INTIALIZED
THEN THE TRANSMIT LIST IS SET TO 3 MESSAGES OF
20 BYTES EACH. THE DATA FOR EACH MESSAGE IS ALTERNATING
1'S AND 0'S. THE NODE IS THEN RUN IN PASSIVE MODE WITH
DATA CHECKING DISABLED AND THE PASS COUNT SET TO 3.

WHAT YOU SHOULD SEE AFTER ENTERING THE RUN COMMAND
IF THINGS ARE RUNNING CORRECTLY :

FOR NODE A:

```
INI RXQ TXQ TXC RXQ TXQ TXC RXQ
TXQ TXC EOP RXQ TXQ TXC RXQ TXQ
TXC RXQ TXQ TXC EOP RXQ TXQ TXC
RXQ TXQ TXC RXQ TXQ TXC EOP
     MODE=ACTIVE/PASS=00000
                /STATUS/NOCHECK/NOECHO/NOMODEM
DCLT> (A)  ?
```

FOR NODE B:

```
INI RXQ TXQ TXC RXQ TXQ TXC RXQ
TXQ TXC EOP RXQ TXQ TXC RXQ TXQ
TXC EOP RXQ TXQ TXC RXQ TXQ TXC
RXQ TXQ TXC EOP
     MODE=PASSIVE/PASS=00000
                /STATUS/NOCHECK/NOECHO/NOMODEM
DCLT> (A)  ?
```

NOW USE DATA CHECKING WITH THE "EXPECT MESSAGE LISTS" SET
UP APPROPRIATELY. ANOTHER VARIATION IS TO HAVE LARGE SIZE
MESSAGES ON ONE SIDE WITH SMALL MESSAGES ON THE OTHER.

THEN REVERSE THE SETUP SO THAT THE NODE RUNNING IN ACTIVE
IS RUNNING IN PASSIVE AND VICE VERSA.

7.2.4 BOTH NODES ACTIVE

NOW BOTH NODES CAN BE RUN IN ACTIVE WITH DATA CHECKING ON.
STATUS PRINTING COULD BE TURNED OFF IF YOU'RE NOT INTERESTED
IN THEM.

NODE A	NODE B
C E	C E
C T	C T
SE T=0ALT/S=10	SE E=0ALT/S=10
SE T=CCITT/S=20	SE E=CCITT/S=20
SE T=ALPHA/S=30	SE E=ALPHA/S=30
SE E=ZERO/S=11	SE T=ZERO/S=11
SE E=ONES/S=21	SE T=ONES/S=21
SE E=ITEP/S=31	SE T=ITEP/S=31
R M=A/CH/NOST/PAS=3	R M=A/CH/NOST/PAS=3

WHAT THIS SEQUENCE MEANS:
NODE A SETS UP IS TRANSMIT LIST TO BE
3 MESSAGES. MESSAGE 1 IS 10 BYTES LONG AND
CONTAINS DATA OF ALTENATING 0'S AND 1'S
MESSAGE 2 IS 20 BYTES LONG AND CONTAINS
DATA OF THE CCITT PATTERN. MESSAGE THREE
IS 30 BYTES LONG AND CONTAINS ALPHANUMERICS
FOR DATA. THE EXPECT LIST ALSO CONTAINS
3 MESSAGES. MESSAGE 1 IS 11 BYTES LONG AND
CONTAINS 0'S FOR DATA. MESSAGE TWO IS 21
BYTES LONG AND CONTAINS 1'S FOR DATA. MESSAGE
3 IS 31 BYTES LONG AND CONTAINS THE ITEP DATA.
NODE B HAS THE SAME MESSAGES EXCEPT THAT THE
TRANSMIT MESSAGE LIST IS THE EXPECT MESSAGE LIST
AND VICE VERSA.
BOTH NODES ARE RUN IN THE ACTIVE MODE WITH
DATA CHECKING AND PASS COUNT EQUAL TO THREE.

WHAT YOU SHOULD SEE AFTER ENTERING THE RUN COMMAND
IF THINGS ARE RUNNING CORRECTLY :
ON BOTH NODES A AND B:
MODE=ACTIVE/PASS=00000
/NOSTATUS/CHECK/NOECHO/NOMODEM
DCLT> (A) ?

A VARIATION THAT CAN BE USED IS FOR ONE END TO SEND A LOT OF
SMALL MESSAGES AND THE OTHER TO SEND A FEW LARGE MESSAGES.
THE ''END-OF-PASS'' POINT WILL BE OUT OF SYNC BUT THIS IS NOT
A PROBLEM.

7.2.5 TALK AND LISTEN MODES FOR COMMUNICATING

TALK AND LISTEN MODES ARE USEFUL IF THE OPERATORS WISH TO COMMUNICATE
WITH EACH OTHER. JUST SETUP A TIME THAT EACH WILL GO TO THEIR MODE,
TALK OR LISTEN, AND SEND MESSAGES OVER THE LINK. POSSIBLE COMMAND
SEQUENCES ARE.

```
R M=LIS/NOST          R M=TA/NOST
LIS>                  TLK>
```

7.3 EXAMPLES OF COMMANDS

THIS SECTION WILL SHOW A SAMPLING OF COMMANDS AND
EXACTLY WHAT TO EXPECT FROM THEM.

7.3.1 EXAMPLES OF MESSAGES COMMANDS

THE CLEAR COMMANDS .

```
C E
C T
```

THIS WILL INITIALIZE THE TRANSMIT AND EXPECT LIST

```
TO 1 MESSAGE OF 58 BYTES. THE DATA OF THE MESSAGE WILL
BE THE ITEP MESSAGE.

IF THESE COMMANDS ARE FOLLOWED BY A SHOW COMMAND
        SH E
SUCH AS THE SHOW EXPECT LIST. WHAT YOU WOULD SEE IS
        MSG: TYPE=ITEP/SIZE=58
             MODE=ACTIVE/PASS=00001
                        /NOSTATUS/CHECK/NOECHO/NOMODEM
        DCLT> (A) ?
NOW IF YOU DID A SET EXPECT LIST COMMAND SUCH AS:
        SE E=A/S=35/C=3
AND FOLLOWED IT WITH A SHOW EXPECT LIST COMMAND
        SH E
WHAT YOU WOULD SEE IS
        MSG: TYPE=ALPHA/SIZE=35
        MSG: TYPE=ALPHA/SIZE=35
        MSG: TYPE=ALPHA/SIZE=35
        MSG: TYPE=ALPHA/SIZE=35
             MODE=ACTIVE/PASS=00001
                        /NOSTATUS/CHECK/NOECHO/NOMODEM
        DCLT> (A) ?


7.3.2 EXAMPLES STATISTICAL COMMANDS

IF YOU TYPE A HELP COMMMAND
        HELP
WHAT YOU WILL SEE IS
DCLT CMDS:
 CLEAR OR SHOW  EXPECTLIST OR TRANSMITLIST
 PRINT
 EXIT
 DUMP START-END/B
 SET EXPECTMSG OR TRANSMITMSG=TYPE/SIZE=N OR /COPY=N
 SET EXPECT=TRANSMIT
   TYPE=ONES,ZEROES,1ALT,0ALT,ITEP,CCITT,ALPHA
        OR 'DPR SPCD=A-Z,SP,TAB,0-9 IN QUOTES''
 RUN MODE=MTYP/LOOP=LTYP/CHECK,STATUS,ECHO,MODEM,PASS=N
   MTYP=TRAN,REC,ACT,PAS,TAL,LIS,DOWN
   LTYP=INT,CAB,LOC,REM/

DCLT> (A)  ?

THE SAME WILL HAPPEN IF YOU USE THE ?

THE DUMP COMMAND WORKS LIKE THIS
        DUM 41260-41300
THIS WILL DUMP THE DATA FROM ADDRESSES 41260 TO
41300 IN THE FOLLOWING MANNER

41260   104423  000167 177772 021122 012112 006312 006312 006312
41300   006312
IF YOU HAD USED THE /B SWITCH
        DUM 41260-41300/B
WHAT YOU WOULD SEE IS
```

```
                41260   023 211 167 000 372 377 122 024
                41270   112 024 312 014 312 014 312 014
                41300   312
```

7.3.3 EXAMPLES RUN COMMMANDS

YOU CAN FIND SEVERAL EXAMPLES OF THE RUN COMMMAND IN THE
TROUBLE SHOOTING HINTS SECTION BUT HERE ARE SOME OTHERS.

IF YOU WERE TO EXECUTE THE RUN COMMAND

```
        R M=TR/NOST/CH/PAS=4
```

WHAT WOULD HAPPEN IS AFTER 4 PASSES THE PROGRAM WOULD RETURN
TO THE DCLT PROMPT AND PRINT

```
        MODE=TRANSMIT/PASS=00000
                    /NOSTATUS/CHECK/NOECHO/NOMODEM
  DCLT> (A)  ?
```

IF YOU WERE TO EXECUTE THE RUN COMMAND

```
        C E
        C T
        R M=A/LO=I/ST/CH/PAS=3
```

WHAT YOU WOULD SEE (IF USING DEFAULT TRANSMIT AND EXPECT
MESSAGES) IS

```
        INI RXQ TXQ TXC CMP EOP RXQ TXQ
        TXC CMP FOP RXQ TXQ TXC CMP EOP
        MODE=ACTIVE/LOOP=INTERNAL/PASS=0000
                    /STATUS/CHECK/NOECHO/NOMODEM
 DCLT> (A)  ?
```

IF YOU USE THE EXIT COMMAND

```
        EXIT
```

WHAT YOU WOULD SEE IS

```
        CZCLK EOP
             0 CUMLATIVE ERRORS

        DR>
```

7.3.4 EXAMPLES PRINT COMMANDS

THE PRINT COMMMAND CAN BE USED FROM THE SUPERVISOR (DR>)
LEVEL OR THE DCLT (DCLT>) LEVEL. ONCE YOU ARE AT THE
REPORT LEVEL YOU WILL KNOW IT BY THE PROMPT 'RPT>''.
AFTER TYPING PRI FOR EITHER THE THE DLCT> OR DR> THE
FOLLOWING IS DISPLAYED.

```
        TYPE 'H'' OR  ''?'' FOR HELP!
        RPT> (A)  ?
```

HERE ARE SOME EXAMPLES OF RPT> LEVEL COMMANDS:

THE HELP OR ? COMMAND

```
        HELP
```

OR

```
        ?
```

PRODUCES THE FOLLOWING:

```
                    DCLT REPORT CMDS:
                    LOG - PRINT DCLT EVENT LOG
                    EXIT - EXIT REPORT LEVEL
                    HELP - PRINT THIS MESSAGE


                    RPT> (A)  ? LOG

                    THE LOG COMMAND PRODUCES THE FOLLOWING

                    >>> DCLT EVENT LOG ENTRY >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
                        0:01:16   RECEIVE SPACE QUEUED
                       ADDR OF MSG=000455   BYTE COUNT= 512
                            MODEM STATUS: CTS DSR DCD RTS RI SQD TM
                                          0   1   0   0   1   1   1
                    >>> DCLT EVENT LOG ENTRY >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
                        0:01:16   RECEIVE SPACE QUEUED
                       ADDR OF MSG=000455   BYTE COUNT= 512
                            MODEM STATUS: CTS DSR DCD RTS RI SQD TM
                                          0   1   0   0   1   1   1
                                         .....
                                         .....
                                         .....

                            THIS GOES ON FOR 45 EVENTS IF THE MODE
                            PREVIOUSLY EXECUTED HAD THAT MANY
                            YOU EXIT FROM EVENT LOG PRINTING BY
                            TYPING A CONTROL C.

                    TO LEAVE THE REPORT LEVEL INPUT THE "EXIT" COMMAND.


                    7.3.5  EXAMPLE EXIT COMMAND

                    THE EXIT COMMAND WORKS LIKE THIS. IF YOU
                    ENTERED THE REPORT LEVEL FROM THE SUPERVISOR
                    (DR>)THEN TYPING

                            EXIT

                    WILL RETURN YOU TO THE SUPERVISOR.

                            DR>

                    IF YOU ENTERED REPORT FROM THE DCLT LEVEL
                    THEN TYPING

                            EXIT

                    WILL RETURN YOU TO THE DCLT LEVEL.

                            DCLT>


                    7.4 THINGS TO WATCH OUT FOR
```

IF YOU ARE RUNNING DCLT ON SYSTEMS THAT HAVE CONSOLES
WITH DIFFERENT SPEEDS YOU WILL BE UNABLE TO USE THE
PRINT STATUS FEATURE IN CERTAIN MODES. THE RULE IS
IF IT DOESNT WORK WITH STATUS PRINTING RUN THE MODE
WITH NOSTATUS.

IF YOU ARE USING PASSIVE MODE WITH THE ECHO SWITCH
THEN YOU WILL PROBABLY HAVE TO RE-ENTER THE TRANSMIT
LIST ON THE SIDE WITH THE ECHO SWITCH. THE REASON IS
THAT THE TRANSMIT LIST GETS OVER WRITTEN WITH THE
RECEIVE LIST WHEN USING THE ECHO SWITCH.

BEWARE THAT THIS DCLT WILL NOT RUN THE DPV11 AT ITS
HIGH CLOCK SPEED OF 50KHZ SINCE THE SOFTWARE IS NOT
ABLE TO KEEP UP WITH THIS SPEED.

&

```
                                .SBTTL  PROGRAM HEADER

002000                                  BGNMOD

                                ;++
                                ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
                                ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
                                ;--

002000                                  POINTER BGNRPT,BGNAU,BGNDU

002000                                  HEADER  CVCLH,B,0,1800.,0,#PRI07
(4) 002000                                                              L$NAME::
(4) 002000      103                             .ASCII  /C/
(4) 002001      126                             .ASCII  /V/
(4) 002002      103                             .ASCII  /C/
(4) 002003      114                             .ASCII  /L/
(4) 002004      110                             .ASCII  /H/
(6) 002005      000                             .BYTE   0
(6) 002006      000                             .BYTE   0
(5) 002007      000                             .BYTE   0
(5) 002010                                                              L$REV::
(4) 002010      102                             .ASCII  /B/
(5) 002011                                                              L$DEPO::
(4) 002011      060                             .ASCII  /0/
(5) 002012                                                              L$UNIT::
(4) 002012  000000                              .WORD   0
(5) 002014                                                              L$TIML::
(4) 002014  003410                              .WORD   1800.
(5) 002016                                                              L$HPCP::
(4) 002016  035346                              .WORD   L$HARD
(5) 002020                                                              L$SPCP::
(4) 002020  000000                              .WORD   0
(5) 002022                                                              L$HPTP::
(4) 002022  002130                              .WORD   L$HW
(5) 002024                                                              L$SPTP::
(4) 002024  000000                              .WORD   0
(5) 002026                                                              L$LADP::
(4) 002026  035624                              .WORD   L$LAST
(5) 002030                                                              L$STA::
(4) 002030  000000                              .WORD   0
(5) 002032                                                              L$CO::
(4) 002032  000000                              .WORD   0
(5) 002034                                                              L$DTYP::
```

```
(4)  002034  000000                                        .WORD   0
(5)  002036                                         L$APT::
(4)  002036  000000                                         .WORD   0
(5)  002040                                         L$DTP::
(4)  002040  002124                                         .WORD   L$DISPATCH
(5)  002042                                         L$PRIO::
(4)  002042  000340                                         .WORD   #PRIO7
(5)  002044                                         L$ENVI::
(4)  002044  000000                                         .WORD   0
(5)  002046                                         L$EXP1::
(4)  002046  000000                                         .WORD   0
(5)  002050                                         L$MREV::
(4)  002050      003                                        .BYTE   C$REVISION
(3)  002051      003                                        .BYTE   C$EDIT
(5)  002052                                         L$EF::
(4)  002052  000000                                         .WORD   0
(5)  002054  000000                                         .WORD   0
(5)  002056                                         L$SPC::
(4)  002056  000000                                         .WORD   0
(5)  002060                                         L$DEVP::
(4)  002060  011474                                         .WORD   L$DVTYP
(5)  002062                                         L$REPP::
(4)  002062  024156                                         .WORD   L$RPT
(5)  002064                                         L$EXP4::
(4)  002064  000000                                         .WORD   0
(5)  002066                                         L$EXP5::
(4)  002066  000000                                         .WORD   0
(5)  002070                                         L$AUT::
(4)  002070  025064                                         .WORD   L$AU
(5)  002072                                         L$DUT::
(4)  002072  025056                                         .WORD   L$DU
(5)  002074                                         L$LUN::
(4)  002074  000000                                         .WORD   0
(5)  002076                                         L$DESP::
(4)  002076  011504                                         .WORD   L$DESC
(5)  002100                                         L$LOAD::
(4)  002100  104035                                         EMT     E$LOAD
(5)  002102                                         L$ETP::
(4)  002102  000000                                         .WORD   0
(5)  002104                                         L$ICP::
(4)  002104  024172                                         .WORD   L$INIT
(5)  002106                                         L$CCP::
(4)  002106  025034                                         .WORD   L$CLEAN
(5)  002110                                         L$ACP::
(4)  002110  025032                                         .WORD   L$AUTO
(5)  002112                                         L$PRT::
(4)  002112  024164                                         .WORD   L$PROT
(5)  002114                                         L$TEST::
(4)  002114  000000                                         .WORD   0
(5)  002116                                         L$DLY::
(4)  002116  000000                                         .WORD   0
(5)  002120                                         L$HIME::
(4)  002120  000000                                         .WORD   0
4117
```

```
 4129                                   .SBTTL  DISPATCH TABLE
 4130
 4131                                   ;++
 4132                                   ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
 4133                                   ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
 4134                                   ;--
 4135
 4136  002122                                   DISPATCH 1
                                                                          .WORD   1
   (4)  002122  000001                                          L$DISPATCH::
   (3)  002124                                                            .WORD   T1
   (6)  002124  025072
 4137
```

```
                                .SBTTL  DEFAULT HARDWARE P-TABLE

                                ;++
                                ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
                                ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
                                ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
                                ; AND IS USED AS A ''TEMPLATE'' FOR BUILDING THE P-TABLES.
                                ;--

002126                                  BGNHW   DFPTBL
(3)  002126  000010                                                     .WORD   L10000-L$HW/2
(3)  002130                                                     L$HW::
(3)  002130                                                     DFPTBL::

                                ;INDEPENDENT SECTION
                                ;       THE NUMBERS IN BRACKETS ARE THE OFFSET VALUES USED IN THE PARAMETER
                                ;       CODING SECTION.

002130  000001                          .WORD   1               ;[0] FULL OR HALF DUPLEX FLAG (BIT0=1 IF FULL)

                                :DEVICE DEPENDENT SECTION
                                ;       ADDING OR REMOVING WORDS FROM THIS TABLE EFFECTS THE ''GET'' CALLS IN
                                ;       THE HARDWARE PARAMTER CODING SECTION BY CHANGING ''OFFSETS''

002132  160170                          .WORD   160170          ;[2] CSR ADDRESS
002134  000300                          .WORD   300             ;[4] INTERRUPT VECTOR
002136  000240                          .WORD   240             ;[6] INTERRUPT PRIORITY (5)
002140  000000                          .WORD   0               ;[10] SPARE
002142  000000                          .WORD   0               ;[12] SPARE
002144  000000                          .WORD   0               ;[14] OTHER NODE ''ITEP''
002146  000000                          .WORD   0               ;[16] SPARE

002150                                  ENDHW
(3)  002150                                                     L10000:
```

```
 4202
 4203
 4220
 4243
 4244                                   .SBTTL  GLOBAL EQUATES SECTION
 4245
 4246
 4256
 4257
 4258                                   ;++
 4259                                   ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
 4260                                   ; ARE USED IN MORE THAN ONE TEST.
 4261                                   ;--
 4262
 4277
 4278  002150                                   EQUALS
  (1)                                   ;
  (1)                                   ; BIT DIFINITIONS
  (1)                                   ;
  (1)           100000                  BIT15== 100000
  (1)           040000                  BIT14== 40000
  (1)           020000                  BIT13== 20000
  (1)           010000                  BIT12== 10000
  (1)           004000                  BIT11== 4000
  (1)           002000                  BIT10== 2000
  (1)           001000                  BIT09== 1000
  (1)           000400                  BIT08== 400
  (1)           000200                  BIT07== 200
  (1)           000100                  BIT06== 100
  (1)           000040                  BIT05== 40
  (1)           000020                  BIT04== 20
  (1)           000010                  BIT03== 10
  (1)           000004                  BIT02== 4
  (1)           000002                  BIT01== 2
  (1)           000001                  BIT00== 1
  (1)                                   ;
  (1)           001000                  BIT9==  BIT09
  (1)           000400                  BIT8==  BIT08
  (1)           000200                  BIT7==  BIT07
  (1)           000100                  BIT6==  BIT06
  (1)           000040                  BIT5==  BIT05
  (1)           000020                  BIT4==  BIT04
  (1)           000010                  BIT3==  BIT03
  (1)           000004                  BIT2==  BIT02
  (1)           000002                  BIT1==  BIT01
  (1)           000001                  BIT0==  BIT00
  (1)                                   ;
  (1)                                   ; EVENT FLAG DEFINITIONS
  (1)                                   ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
  (1)                                   ;
  (1)           000040                  EF.START==      32.                     ; START COMMAND WAS ISSUED
  (1)           000037                  EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
  (1)           000036                  EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
  (1)           000035                  EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
  (1)           000034                  EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
  (1)                                   ;
```

```
(1)                                 ;
(1)                                 ; PRIORITY LEVEL DEFINITIONS
(1)                                 ;
(1)             000340              PRI07== 340
(1)             000300              PRI06== 300
(1)             000240              PRI05== 240
(1)             000200              PRI04== 200
(1)             000140              PRI03== 140
(1)             000100              PRI02== 100
(1)             000040              PRI01== 40
(1)             000000              PRI00== 0
(1)                                 ;
(1)                                 ;OPERATOR FLAG BITS
(1)                                 ;
(1)             000004              EVL==         4
(1)             000010              LOT==        10
(1)             000020              ADR==        20
(1)             000040              IDU==        40
(1)             000100              ISR==       100
(1)             000200              UAM==       200
(1)             000400              BOE==       400
(1)             001000              PNT==      1000
(1)             002000              PRI==      2000
(1)             004000              IXE==      4000
(1)             010000              IBE==     10000
(1)             020000              IER==     20000
(1)             040000              LOE==     40000
(1)             100000              HOE==    100000
4279
```

```
                                ;***** INDEPENDENT EQUATES
                                
001000                                  BUFLIM=512.             ;MAX BUFFER SIZE IN BYTES
                                                                ;  APPLIES TO TX,RX AND CMP BUFFS
000017                                  MSGLIM=15.              ;MAX NO. OF MESSAGES PER BUFFER
                                                                ;  (FOR EACH INCREMENT (+1) TO MSGLIM,
                                                                ;   ADD 6 WORDS TO THE POINTER TABLE
                                                                ;   (PTRTAB:) SINCE THIS MEANS 2 MORE
                                                                ;   'POINTER'' WORDS PER BUFFER.

                                ;MODE OF OPERATION EQUATES
000000                                  REC=0                   ;RECEIVE MODE
000001                                  TRA=1                   ;TRANSMIT MODE
000002                                  PAS=2                   ;PASSIVE MODE
000003                                  ACT=3                   ;ACTIVE MODE
000004                                  DOW=4                   ;DOWN-LINE-LOAD MODE
000005                                  TAL=5                   ;TALK MODE
000006                                  LIS=6                   ;LISTEN MODE
                                ;MAINT LOOP TYPE EQUATES
000000                                  NONE=   0               ;NO LOOP
000001                                  TTL=    1               ;INTERNAL TTL
000002                                  CABLE=  2               ;CABLE LOOP
000003                                  MODLOC= 3               ;MODEM LOCAL
000004                                  MODREM= 4               ;MODEM REMOTE
000005                                  MOP=    5               ;MOP


                                ;CLOCK ENABLE VALUES TO BE LOADED IN CLK'S CSR
000100                                  LCLKEN= 100             ;L-CLOCK CSR VALUE TO ENABLE THE CLOCK
000111                                  PCLKEN= 111             ;P-CLOCK CSR VALUE TO ENABLE THE CLOCK
001600                                  PCLKCT= 1600            ;P-CLOCK COUNT SET REGISTER FOR COUNTER

                                ;PARAM WORD EQUATES

000001                                  STATB=  BIT0            ;OPERATOR AWAKE ASKED FOR
000002                                  DATCKB= BIT1            ;DATA CHECK BIT
000004                                  ECHOB=  BIT2            ;ECHO BIT
000010                                  MOCHK=  BIT3            ;MODEM STATUS CHECK BIT
000020                                  CRCB=   BIT4            ;CRC CALCUALTE ASKED FOR
000040                                  PROTOB= BIT5            ;PROTOCOL PROCESSING ASKED FOR

                                ;OPTION TYPE EQUATES


000000                                  DPV=    0       ;CODE FOR DPV CHAR MODE

                                ;EVENT LOG MESSAGE TYPES (USED TO LOCATE EVENT DESCRIPTION IN EVENT TABLE
                                ;  AND DISPATCHING TO SEPERATE SECTIONS OF THE EVENT REPORTING SECTION)
000000                                  TXQ=    0               ;TRANSMIT MESSAGE QUEUED
000002                                  TXC=    2               ;TRANSMIT COMPLETE
000004                                  RXQ=    4               ;RECEIVE BUFFER QUEUED
000006                                  RXC=    6               ;RECEIVE COMPLETE
000010                                  DER=    10              ;DEVICE INFORMATION
000012                                  DVI=    12              ;DEVICE ABOUT TO INIT
000014                                  DCK=    14              ;DATA COMPARISON RESULTS
```

```
4356    000016                  MSC=    16              ;MODEM STATUS CHANGE
4357
4358    000020                  DLE=    20              ;DATA COMPARISON LENGH ERROR
4359    000022                  DDE=    22              ;DATA COMPARISON DATA ERROR
4360    000024                  EOP=    24              ;END OF PASS
4361
4362                    ;EQUATES FOR FLAG WORD
4363
4364    000001                  ININT=  BIT0            ;INPUT INT. REC.
4365    000002                  OTINT=  BIT1            ;OUTPUT INT REC
4366    000004                  QRX=    BIT2            ;RX QUED /COMPL
4367    000010                  QTX=    BIT3            ;TX QUED/COMPL
4368    000100                  ERX=    BIT6            ;EXPECT TO GET A RX COMPLED
4369    000200                  ETX=    BIT7            ;EXPECT TO GET A TX COMPLETED
4370
4381
4382    000020                  TXM=    BIT4            ;INDICATES TO TX INTERRUPT ROUTINE
4383                                                    ;THAT IT IS TIME TO TRANSMIT BODY OF MSG.
4384    000040                  RXM=    BIT5            ;INDICATES TO RX INTERUPPT ROUTINE
4385                                                    ;THAT IT IS TIME TO REC MSG BODY
4386    000400                  BCC=    BIT8            ;TIME FOR CRC CHECK.
4387
4388    001000                  PAD=    BIT9            ;INDICATES THAT PAD MUST BE SENT
4389
4390    002000                  INOVR=  BIT10           ;INIT OVER
4391
4392    004000                  FIRST=  BIT11           ;FIRST  TIME FOR CTS
4393
4394                    ; SPECIAL CLI CODES FOR "CHAR" ARGUMENT IN CLI CALLS
4395                    ;  (COMMAND LINE INTERPRETER DEFINITIONS)
4396    000000                  CLIERR= 0
4397    000001                  CLIEXI= 1
4398    000002                  CLIBR=  2
4399    000003                  CLIBIF= 3
4400    000004                  CLISPA= 4
4401    000005                  CLINUM= 5
4402    000006                  CLIALP= 6
4403    000007                  CLIALN= 7
4404    000010                  CLIOCT= 8.
4405    000011                  CLIDEC= 9.
4406    000012                  CLISTR= 10.
4407
4408                    ; DEFS FOR COMMAND LINE INTERPRETATION ACTION VALUES
4409    000000                  NULL=0
4410    000001                  CLEAR=1
4411    000002                  SHOW=2
4412    000003                  CHECK=3
4413    000004                  RUN=4
4414    000005                  HLP=5
4415    000006                  CSHEXP=6
4416    000007                  CSHTRN=7
4417    000010                  SETEXP=10
4418    000011                  SETTRN=11
4419    000012                  SIZE=12
4420    000013                  QCOPY=13
4421    000014                  NUM=14
```

```
4422    000015                  OPRMSG=15
4423    000016                  STATUS=16
4424    000017                  ENDQO=17
4425    000020                  CMSG0=20
4426    000021                  CMSG1=21
4427    000022                  CMSG2=22
4428    000023                  CMSG3=23
4429    000024                  CMSG4=24
4430    000025                  CMSG5=25
4431    000026                  CMSG6=26
4432    000027                  ATVMOD=27
4433    000030                  PASMOD=30
4434    000031                  RECMOD=31
4435    000032                  LISMOD=32
4436    000033                  DLLMOD=33
4437    000034                  TRAMOD=34
4438    000035                  TALMOD=35
4439    000036                  NO=36
4440    000037                  ECHO=37
4441    000040                  CRC=40
4442    000041                  PROTO=41
4443    000042                  PASC=42
4444    000043                  MOP=43
4445    000044                  TTLLOP=44
4446    000045                  CBLLOP=45
4447    000046                  LMDLOP=46
4448    000047                  RMDLOP=47
4449    000050                  NOTNUF=50
4450    000051                  BADCHR=51
4451    000052                  DMPS=52
4452    000053                  DMPE=53
4453    000054                  DMPQ=54
4454    000055                  PRNT=55
4455    000056                  MOSC=56
4456    000057                  EXIT=57         ;REV B BY EC
4457    000060                  SETET=60        ;REV B BY EC
4458                    ;FOLLOWING EQUATES USED IN REPORT CLI ; REV B EC
4459    000001                  RPHLP=1
4460    000002                  RPEXT=2
4461    000003                  RPLOG=3
4473
4474                    ;***** DEVICE DEPENDENT EQUATES
4475                    ; MODEM SIGNAL BIT DEFINITONS
4476                    ;       IF SIGNAL AVAILABLE IN DEVICE, EQUATE NAME TO BIT POSITION,
4477                    ;       ELSE EQUATE IT TO = 0
4478
4479    020000                  CTS=    BIT13           ;CLEAR TO SEND (CIRCUIT CB)
4480    001000                  DSR=    BIT9            ;DATA SET READY (CIRCUIT CC)
4481    010000                  DCD=    BIT12           ;DATA CARRIER DETECT (CIRCUIT CF)
4482    000004                  RTS=    BIT2            ;REQUEST TO SEND (CIRCUIT CA)
4483    040000                  RI=     BIT14           ;RING INDICATOR (CIRCUIT CE)
4484    000040                  SQD=    BIT5            ;SIGNAL QUALITY DETECT (CIRCUIT CG)
4485    000040                  TM=     BIT5            ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)
4486
4494
4495
```

GLOBAL EQUATES SECTION

```
                        ; DEVICE SIGNALS
000002                          DTR=    BIT1            ;DATA TERMINAL READY
000020                          RXENA=  BIT4            ;RECEIVER ENABLE
000040                          DSITEN= BIT5            ;DATA SET CHANGE ENABLE
000100                          RINTEN= BIT6            ;REC INT. ENABLE
000200                          RDATRY= BIT7            ;REC DATA READY
002000                          RSTARY= BIT10           ;REC STATUS READY
004000                          RXACT=  BIT11           ;REC ACTIVE
000001                          RESET=  BIT0            ;MASTER RESET
000002                          TXACT=  BIT1            ;TX ACTIVE
000004                          TBMT=   BIT2            ;TX BUFFER EMPTY
000010                          TTLL=   BIT3            ;TTL LOOP BIT
000020                          TXENA=  BIT4            ;TX ENABLE
000100                          TINTEN= BIT6            ;TX INT ENABLE
000400                          TSOM=   BIT8            ;TX START OF MSG.
001000                          TEOM=   BIT9            ;TX END OF MSG.
100000                          TERR=   BIT15           ;TX ERROR
100000                          RERR=   BIT15           ;REC OVER RUN
000226                          SYN=    226             ;SYNC WORD
```

```
                                                                H 4
 4517                                   .SBTTL  GLOBAL DATA SECTION
 4518                                   .SBTTL          DEFAULT MESSAGE DEFINITIONS AND TABLES
 4519
 4520                                   ;++
 4521                                   ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
 4522                                   ; IN MORE THAN ONE TEST.
 4523                                   ;--
 4524
 4525                                   ;MESSAGE BYTE COUNT TABLE
 4526
 4527  002150                           DMSGCT:
 4528  002150  000001                  MSG0C:  .WORD   EMSG0-MSG0      ;BYTE COUNT OF MESSAGE #0
 4529  002152  000001                  MSG1C:  .WORD   EMSG1-MSG1      ;BYTE COUNT OF MESSAGE #1
 4530  002154  000001                  MSG2C:  .WORD   EMSG2-MSG2      ;BYTE COUNT OF MESSAGE #2
 4531  002156  000001                  MSG3C:  .WORD   EMSG3-MSG3      ;BYTE COUNT OF MESSAGE #3
 4532  002160  000100                  MSG4C:  .WORD   EMSG4-MSG4      ;BYTE COUNT OF MESSAGE #4
 4533  002162  000072                  MSG5C:  .WORD   EMSG5-MSG5      ;BYTE COUNT OF MESSAGE #5
 4534  002164  000101                  MSG6C:  .WORD   EMSG6-MSG6      ;BYTE COUNT OF MESSAGE #6
 4535  002166  000000                  OPCNT:  .WORD   0               ;BYTE COUNT FOR OPERATOR SPEC'D MSG.
 4536  002170  000001                  MSG8C:  .WORD   EMSG8-MSG8      ;BYTE COUNT OF RECEIVE BUFFER FILL PATTERN
 4537
 4538                                   ;MESSAGE ADDRESS TABLE
 4539
 4540  002172                           DMSGAD:
 4541  002172  002214                          MSG0                    ;ADDRESS OF MESSAGE #0
 4542  002174  002215                          MSG1                    ;ADDRESS OF MESSAGE #1
 4543  002176  002216                          MSG2                    ;ADDRESS OF MESSAGE #2
 4544  002200  002217                          MSG3                    ;ADDRESS OF MESSAGE #3
 4545  002202  002220                          MSG4                    ;ADDRESS OF MESSAGE #4
 4546  002204  002320                          MSG5                    ;ADDRESS OF MESSAGE #5
 4547  002206  002412                          MSG6                    ;ADDRESS OF MESSAGE #6
 4548  002210  002520                          OPBUF                   ;ADDRESS OF OPERATOR SPEC'D MSG.
 4549  002212  002642                          MSG8                    ;ADDRESS OF RECEIVE BUFFER FILL PATTERN
 4550
 4551  002214     000                  MSG0:   .BYTE   000             ;MESSAGE OF ALL 0'S
 4552  002215                          EMSG0:
 4553  002215     377                  MSG1:   .BYTE   377             ;MESSAGE OF ALL 1'S
 4554  002216                          EMSG1:
 4555  002216     252                  MSG2:   .BYTE   252             ;MESSAGE OF ALTERNATING 1'S
 4556  002217                          EMSG2:
 4557  002217     125                  MSG3:   .BYTE   125             ;MESSAGE OF ALTERNATING 0'S
 4558  002220                          EMSG3:
 4559  002220                          MSG4:                           ;"CCITT" 512-BIT (VS. 511 BITS) TEST PATTERN
 4560  002220  177603  157427  031011          .WORD   177603,157427,031011,047321,163715,105221,143325,142304
       002226  047321  163715  105221
       002234  143325  142304
 4561  002240  040041  014116  052606          .WORD   040041,014116,052606,172334,105025,123754,111337,111523
       002246  172334  105025  123754
       002254  111337  111523
 4562  002260  030030  145064  137642          .WORD   030030,145064,137642,143531,063617,135075,066730,026575
       002266  143531  063617  135075
       002274  066730  026575
 4563  002300  052012  053627  070071          .WORD   052012,053627,070071,151172,165044,031605,166632,016741
       002306  151172  165044  031605
       002314  166632  016741
 4564  002320                          EMSG4:
```

```
 4565  002320                                MSG5:                           ;''INTERPROCESSOR TEST PROGRAM'S (ITEP)'' MESSAGE
 4566                                                                        ; #1, (DP1:)
 4567  002320  077577  040444  052040              .ASCII  <177><177>/$A THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG./
       002326  042510  050440  044525
       002334  045503  041040  047522
       002342  047127  043040  054117
       002350  045040  046525  042520
       002356  020104  053117  051105
       002364  052040  042510  046040
       002372  055101  020131  047504
       002400  027107
 4568  002402  005015  077401  077577              .ASCIZ  <15><12><001><177><177><177><177>
       002410  000177
 4569  002412                                EMSG5:
 4570  002412                                MSG6:                           ;ALPHA-NUMERICS (OR FUTURE COMM TURNAROUND MSG)
 4571  002412  022043  021041  023040              .ASCII  /#$!'' &'()**,-.0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ/
       002420  024047  025051  026053
       002426  027055  030460  031462
       002434  032464  033466  034470
       002442  035472  036474  037476
       002450  040500  041502  042504
       002456  043506  044510  045512
       002464  046514  047516  050520
       002472  051522  052524  053526
       002500  054530     132
 4572  002503     057  056133  057135              .ASCIZ  ?/[\]^_%?
       002510  022537     000
 4573  002513                                EMSG6:
 4574          002514                              .EVEN
 4575
 4576                                        ; ********************************************************************************
 4577                                        ;THESE THREE STORAGE AREAS MUST NOT BE SEPERATED !!!!
 4578
 4579  002514  047045  040445               OPBFPT: .ASCII  /%N%A/
 4580  002520  000122                       OPBUF:  .BLKB   82.             ;BUFFER FOR OPERATOR SPEC'D MESSAGES
 4581  002642                               OPEND:
 4582
 4583                                        ; THE ABOVE THREE LINES MUST BE KEPT TOGETHER
 4584                                        ; ********************************************************************************
 4585
 4586  002642     033                       MSG8:   .BYTE   33              ;RECEIVE BUFFER FILL PATTERN
 4587  002643                               EMSG8:
 4588          002644                              .EVEN
```

```
                           ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                           ; THE FOLLWING IS THE AREA USED TO TRANSMIT AND REC THE ;
                           ; HEADER MSGS. AND THE START,STACK ACK SEQUENCES.       ;
                           ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
002644     000             HDMSG:  .BYTE 0            ;FILLER
002645     201                     .BYTE 201          ;SOH CHAR
002646  000000             HDMCC:  .WORD   0          ;CHAR COUNT GOES HERE
002650     001                     .BYTE   1          ;RESPONSE NUMBER
002651     001                     .BYTE   1          ;MSG. NUMBER
002652     001                     .BYTE   1          ;ADDR TO.
002653                     HSMSE:
        002654                     .EVEN
002654  000006             HDMC:   .WORD   6

002656     000             RHDMSG: .BYTE 0
002657     201                     .BYTE 201          ;SOH GOES IN HERE
002660  000000             RHDMCC: .WORD              ;BYTE COUNT GOES HERE
002662     001                     .BYTE   1          ;RESP NUM
002663     001                     .BYTE   1          ;MSG NUM
002664     001                     .BYTE   1          ;ADDR TO.
        002666                     .EVEN
```

```
                                ;COMMAND LINE BUFFER, DATA LOCATIONS AND MESSAGES FOR ACTION ROUTINES

003666  000122                  CMDBUF: .BLKB   82.             ;BUFFER FOR OPERATOR COMMANDS
003010  000000                  KEYWD1: .WORD   0               ;THIS LOC WILL =1 IF CLEAR TYPED, 2 FOR SHOW,
                                                                ; A 4 IF RUN WAS TYPED, 5 IF HELP WAS TYPED
003012  000000                  QUALFG: .WORD   0               ;THIS LOC HOLDS QUALIFIER VALUE (SIZE OR COPY)
003014  000000                  QUALVL: .WORD   0
003016  012246                  HLPTAB: .WORD   HLP1
003020  012261                          .WORD   HLP2
003022  012376                          .WORD   HLP3
003024  012463                          .WORD   HLP3A
003026  012510                          .WORD   HLP4
003030  012567                          .WORD   HLP4A
003032  012645                          .WORD   HLP5
003034  012735                          .WORD   HLP6
003036                          HLPEND:
                                ;INDEX TABLE FOR REPORT 'RPT>' HELP MESSAGES REV B EC
003036  013072                  RHLPTB: .WORD   RHLP1
003040  013114                          .WORD   RHLP2
003042  013147                          .WORD   RHLP3
003044  013200                          .WORD   RHLP4
003046                          RHLPEN:

003046  013266  013275  013302  SHTYTB: .WORD   SHTYP0,SHTYP1,SHTYP2,SHTYP3,SHTYP4,SHTYP5,SHTYP6,SHTYP7
003054  013307  013314  013322
003062  013327  013335

                                ; THE LIST OF BYTES BELOW ARE THE FIRST BYTES OF THE PREDEFINED MESSAGES
                                ;  USED TO ''SHOW'' THE TRANSMIT AND COMPARE BUFFER CONTENTS.

003066    000     377     252   SHTAB:  .BYTE   0,377,252,125,203,177,043
003071    125     203     177
003074    043
003075                          SHTEND:
          003076                        .EVEN

003076  013346                  MODES:  .WORD   M00      ;ADDRESSES OF MODE TYPES IN ASCII
003100  013356                          .WORD   M01
003102  013367                          .WORD   M02
003104  013377                          .WORD   M03
003106  013406                          .WORD   M04
003110  013423                          .WORD   M05
003112  013430                          .WORD   M06

003114  013437                  LOOPS:  .WORD   LP0      ;ADDRESSES OF LOOP TYPES IN ASCII
003116  013447                          .WORD   LP1
003120  013460                          .WORD   LP2
003122  013466                          .WORD   LP3
003124  013501                          .WORD   LP4

                                ;COMMAND LINE TRAVERSE LOCATIONS (USED BY 'P$TRV')

003126  000000                  P$BUFA: .WORD   0                ;LOC. TO HOLD ADDR. OF CMD LINE BUFFER
003130  000000                  P$TREE: .WORD   0                ;LOC. TO HOLD ADDR. OF PARSING TREE
003132  000000                  P$ACT:  .WORD   0                ;LOC. TO HOLD ADDR. OF ACTION ROUTINE
003134  000000                  P$CNT:  .WORD   0                ;LOC. TO BE A COUNTER LOCATION
```

```
 4665  003136  000000                P$NUM:  .WORD   0               ;LOC. TO HOLD NUMERIC VALUE FROM PARSE
 4666  003140  000000                P$RADX: .WORD   0               ;LOC. TO HOLD RADIX USED(LO) AND +/-(HI BYTE)
 4667  003142     000                P$NNUF: .BYTE   0               ;RETURN =0 IF ENOUGH OF COMMAND FOUND
 4668  003143     000                P$GDBD: .BYTE   0               ;RETURN CODE 0 IF NO ERROR FOUND
 4669
```

```
 4671                                   .SBTTL          MESSAGE BUFFERS AND POINTER TABLES
 4672
 4673  003144  001000                   TXBUF:  .BLKB   BUFLIM  ;TRANSMITTER BUFFERS
 4674  004144  001000                   RXBUF:  .BLKB   BUFLIM  ;RECEIVER BUFFERS
 4675  005144  001000                   CMPBUF: .BLKB   BUFLIM  ;COMPARISON BUFFERS
 4676  006144  000036                   PTRTAB: .BLKW   MSGLIM*2        ;TABLE FOR MESSAGE ADDRS. & BYTE COUNTS
 4677  006240  000036                   PTR13:  .BLKW   MSGLIM*2
 4678  006334  000036                   PTR23:  .BLKW   MSGLIM*2
 4679  006430                           PTREND:                 ; END OF MSG. PTR. TABLE
 4680
 4681  006430  000002                           .BLKW   2       ;FILLER FOR OVERFLOW OF RX POINTER TABLE
 4682
 4683  006434  000000                   RXPTR:  .WORD   0       ;RECEIVER MESSAGE POINTER
 4684  006436  000000                   TXPTR:  .WORD   0       ;TRANMITTER BUFFER POINTER
 4685  006440  000000                   CMPPTR: .WORD   0       ;COMPARISON BUFFER POINTER
 4686  006442  000000                   CMPTOT: .WORD   0       ;CMP MSG TOTAL
 4687  006444  000000                   CTOTCC: .WORD   0       ;COMPARE BUFFER CHAR. COUNT
 4688  006446  000000                   CCURAD: .WORD   0       ;CURRENT ADDR OF CMP BUFF TO ADD AT
 4689
 4690  006450  000000                   DVTXA:  .WORD   0       ;DEVICE TX ADDR
 4691  006452  000000                   DVTCC:  .WORD   0       ;DEVICE TX CHAR COUNT
 4692  006454  000000                   DVTCT:  .WORD   0       ;DEVICE TX MESSAGE COUNT
 4693  006456  000000                   TXMTOT: .WORD   0       ;TX MSG TOTAL
 4694  006460  000000                   TTOTCC: .WORD   0       ;TX BUFFER CHAR. COUNT
 4695  006462  000000                   TCURAD: .WORD   0       ;CURRENT ADDR. OF TX BUFF TO ADD AT
 4696
 4697  006464  000000                   DVRXA:  .WORD   0       ;DEVICE RX ADDR
 4698  006466  000000                   DVRCC:  .WORD   0       ;DEVICE RX CHAR COUNT
 4699  006470  000000                   DVRCT:  .WORD   0       ;DEVICE RX MESSAGE COUNT
 4700  006472  000000                   RXMTOT: .WORD   0       ;RX MSG TOTAL
 4701
 4702  006474  000000                   LNCNT:  .WORD   0       ;NUMBER OF OPERATOR AWAKE MSGS
 4703  006476  000000                   OPVAR:  .WORD   0       ;OPTTIONAL VARIABLE LOCATION
 4704  006500  000000                   PSCNT:  .WORD   0       ;PASS COUNTER
 4705  006502  000000                   ERRCNT: .WORD   0       ;ERROR COUNTER
 4706  006504  000000                   STADD:  .WORD   0       ;START ADDR.
 4707  006506  000000                   ENADD:  .WORD   0       ;END ADDR. FOR DUMP
 4708  006510  000000                   BYTBIT: .WORD   0       ;BYTE BIT FOR DUMP ROUTINE
 4709
 4710                                   ;OTHER MESSSAGE RELATED STORAGE LOCATIONS
 4711
 4712  006512  000000                   MSGTYP: .WORD   0       ;TYPE OF DATA 0=0'S,1=1'S,2=10'S,3=01'S
 4713                                                           ;4=CCITT,5=QUICK FOX,6=ALPHA/NUM,7=OPER
 4714  006514  000000                   CURCC:  .WORD   0       ;TX/RX/CMP CHAR COUNT
 4715  006516  000000                   CPTRR:  .WORD   0       ;CURRENT RX POINTER
 4716  006520  000000                   CPTR:   .WORD   0       ;CURRENT POINTER
 4717  006522  000000                   CURADD: .WORD   0       ;CURRENT TX/RX/CMP START ADDD
 4718  006524  000000                   TOTCC:  .WORD   0       ;TOTAL CHAR COUNT NOT MORE THEN 'BUFLIM'
 4719  006526  000000                   OFSET:  .WORD   0       ;OFFSET COUNT
 4720  006530  000000                   TEMP:   .WORD   0       ;TEMPORARY LOCATIONS (USED A LOT)
 4721  006532  000000                   TEMP1:  .WORD   0
 4722  006534  000000                   TEMP2:  .WORD   0
 4723  006536  000000                   TEMP3:  .WORD   0
 4724  006540  000000                   TEMP4:  .WORD   0
 4725  006542  000000                   TEMP5:  .WORD   0
 4726  006544  000000                   CONOTM: .WORD   0       ;CONTROL OUT ERROR MSG. ADDRESS
```

```
 4727  006546  000000                CONTIN: .WORD   0       ;WORD FOR CONTROL IN
 4728  006550     000                GOOD:   .BYTE   0       ;BYTE TO HOLD EXPECTED MESSAGE DATA BYTE FOR ERR REPORT
 4729  006551     000                BAD:    .BYTE   0       ;BYTE TO HOLD RECEIVED MESSAGE DATA BYTE FOR ERR REPORT
 4730
```

```
                                ;MORE INDEPENDENT CODE STORAGE LOCATIONS

006552  000000                  LOGUNT: .WORD   0       ;LOC. TO HOLD LOGICAL UNIT NUMBER
006554  000000                  PCADD:  .WORD   0       ;LOC. HOLD PC OF CALLING ROUTINE
006556  000000                  DCLFLG: .WORD   0       ;CLEANUP & EXIT FLAG -1 = EXIT TEST
006560  000000                  RESFLG: .WORD   0       ;LOC TO HOLD FLAG (-1) THAT A RESTART WAS GIVEN
006562  000000                  MODTYP: .WORD   0       ;DCLT MODE OF OPERATION TYPE
                                                        ;  (0=REC-ONLY, 1=TX-ONLY, 2=PASSIVE-LOOPBK,
                                                        ;   3=ACTIVE-LOOPBK, 4=DOWN L.L., 5=TALK, 6=LISTEN)
006564  000000                  MLTYP:  .WORD   0       ;MAINTENANCE LOOP TYPE (0=NONE, 1=INTERNAL TTL,
                                                        ;  2=CABLE, 3=MODEM-ANALOG LOOPBK (LOCAL),
                                                        ;  4=MODEM-DIGITAL LOOPBK (REMOTE), 5=MOP)
006566  000000                  FHDPLX: .WORD   0       ;FULL OR HALF DUPLEX FLAG (1=FULL FROM P-TABLE)
006570  000002                  PARAM:  .WORD   2       ;PROGRAM PARAMETERS
                                                        ; BIT0= STATUS MSGS TO OPR PRINTED (1=YES)
                                                        ; BIT1= DATA CHECKING DONE ON RCVD MSGS (1=YES)
                                                        ; BIT2= ECHO (TRANSMIT) RCV'D MSG.(PASSIVE)(1=YES)
                                                        ; BIT3= MODEM STATUS CHECK (1=YES)
                                                        ; BIT4= CRC CALC./CHECK DONE (1=YES)
                                                        ; BIT5= PROTOCOL EMULATION (1=YES)
                                                        ; BIT6= SPARE
006572  000000                  RPASS:  .WORD   0       ;PASS NUMBER FROM RUN COMMAND
006574  000000                  FLAG:   .WORD   0       ;DEVICE FLAG WORD

                                :MODE DISPATCH TABLE
006576  030704                  MODE:   .WORD   RXONLY  ;RX ONLY DISPATCH
006600  030736                          .WORD   TXONLY  ;TX ONLY DISPATCH
006602  030776                          .WORD   PLCK    ;PASSIVE LOOP BACK DISP
006604  031032                          .WORD   ALCK    ;ACTIVE LOOP BACK DISP
006606  032216                          .WORD   DLL     ;DOWN LINE LOAD DISP
006610  032242                          .WORD   TALCK   ;TALK MODE DISPATCH
006612  032462                          .WORD   LISCK   ;LISTEN MODE DISPATCH


                                .SBTTL          CLOCK TABLES, EVENT LOG AND POINTERS
006614  000000                  CLKCSR: .WORD   0       ;CLOCK CSR ADDRESS
006616  000000                  CLKBR:  .WORD   0       ;CLOCK INTERRUPT LEVEL
006620  000000                  CLKVEC: .WORD   0       ;CLOCK INTERRUPT VECTOR
006622  000074                  CLKHZ:  .WORD   60.     ;CLOCK'S HERTZ RATE
006624  000000                  CLKEN:  .WORD   0       ;CLOCK'S CSR VALUE TO INTRPT. ENABLE IT

006626  000000                  TIMMIN: .WORD   0       ;PLACE TO KEEP TIME-SINCE-START
006630  000000                  TIMSEC: .WORD   0
006632  000000                  TIMTCK: .WORD   0       ;PLACE TO KEEP # OF TICKS/SEC

006634  000000                  TIMER1: .WORD   0       ;EVENT TIMER #1 (TICKS)
006636  000000                  TIMER2: .WORD   0       ;EVENT TIMER #2 (TICKS)
006640  000000                  TIMERS: .WORD   0       ;EVENT TIMER #3 (SECONDS)
```

```
 4782
 4783  006642  006644                ;EVENT LOG TABLE AND ITS NEXT ENTRY POINTER
 4784  006644  000341                EVTPTR: .WORD   EVTLOG  ;POINTER TO NEXT FREE SPACE IN EVENT LOG
 4785  007546  000001                EVTLOG: .BLKW   225.    ;EVENT LOG BUFFER
 4786                                EVTEND: .BLKW   1.      ;APPROXIMATE END OF EVENT TABLE (ALLOWS CIRCULAR QUE)
 4787
 4788                                .SBTTL          MODEM DATA SECTION
 4789  007550  000000                MODS:   .WORD   0       ;MODEM STATUS
 4790
 4791
```

```
                                ;TABLE OF MODEM SIGNAL BIT DEFINITIONS

007552  020000  MOBITS: .WORD   CTS             ;CLEAR TO SEND (CIRCUIT CB)
007554  001000          .WORD   DSR             ;DATA SET READY (CIRCUIT CC)
007556  010000          .WORD   DCD             ;DATA CARRIER DETECT (CIRCUIT CF)
007560  000004          .WORD   RTS             ;REQUEST TO SEND (CIRCUIT CA)
007562  040000          .WORD   RI              ;RING INDICATOR (CIRCUIT CE)
007564  000040          .WORD   SQD             ;SIGNAL QUALITY DETECT (CIRCUIT CG)
007566  000040          .WORD   TM              ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)
007570          MOBITE:

                                ;TABLE OF ADDRESSES OF MODEM SIGNAL MESSAGE POSITIONS

007570  016171  MOMSGS: .WORD   EVMCTS          ;CLEAR TO SEND (CIRCUIT CB)
007572  016175          .WORD   EVMDSR          ;DATA SET READY (CIRCUIT CC)
007574  016201          .WORD   EVMDCD          ;DATA CARRIER DETECT (CIRCUIT CF)
007576  016205          .WORD   EVMRTS          ;REQUEST TO SEND (CIRCUIT CA)
007600  016211          .WORD   EVMRI           ;RING INDICATOR (CIRCUIT CE)
007602  016215          .WORD   EVMSQD          ;SIGNAL QUALITY DETECT (CIRCUIT CG)
007604  016221          .WORD   EVMTM           ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)

                                ;TABLE OF ADDRESSES OF EVENT DESCRIPTION MESSAGES
                                ;       ORDER CORRESPONDS TO MESSAGE TYPE VALUES

007606  014604  EVTLST: .WORD   EDTXQ   ;TRANSMIT MESSAGE QUEUED
007610  014630          .WORD   EDTXC   ;TRANSMIT OF MESSAGE COMPLETE
007612  014657          .WORD   EDRXQ   ;RECEIVE MESSAGE SPACE QUEUED
007614  014704          .WORD   EDRXC   ;MESSAGE RECEIVED - RECEIVE COMPLETE
007616  014732          .WORD   EDDER   ;DEVICE INFORMATION
007620  014777          .WORD   EDDVI   ;DEVICE INITIALIZE STARTED
007622  014747          .WORD   EDDCK   ;DATA COMPARISON DONE
007624  015131          .WORD   EDMOS   ;MODEM STATUS CHANGE
007626  015025          .WORD   EDDLE   ;DATA COMPARE LENGTH ERROR
007630  015062          .WORD   EDDDE   ;DATA COMPARE DATA ERROR
007632  015115          .WORD   EDEOP   ;END OF PASS

                                ;LOCATIONS USED DURING EVENT REPORTING

007634  000000  EVTSEC: .WORD   0       ;TEMPORARY LOCS TO KEEP EVENT TIME WHILE REPORTING
007636  000000  EVTMIN: .WORD   0
007640  000000  EVTTCK: .WORD   0
007642  000000  EVTADD: .WORD   0       ;TEMP. LOC. TO HOLD ADDRESS DURING EVENT REPORTING
007644  000000  EVTBCT: .WORD   0       ;       ''      ''  BYTE COUNT   ''      ''      ''
007646  000000  EVTTMP: .WORD   0       ;       ''      ''  OTHER DATA   ''      ''      ''

                                ;REPORT CODING DISPATCH TABLE

007650  021150  RPTDSP: .WORD   RPTTXQ  ;TRANSMIT QUEUED ENTRY DECODING
007652  021150          .WORD   RPTTXQ  ;TRANSMIT COMPLETE ENTRY DECODING
007654  021150          .WORD   RPTTXQ  ;RECEIVER QUEUED ENTRY DECODING
007656  021150          .WORD   RPTTXQ  ;RECEIVER COMPLETE ENTRY DECODING
007660  021222          .WORD   RPTDER  ;DEVICE ERROR ENTRY DECODING
007662  021316          .WORD   RPTDVI  ;DEVICE INIT ENTRY DECODING
007664  021512          .WORD   RPTDCK  ;DATA COMPARISON ENTRY DECODING
007666  021566          .WORD   RPTMSC  ;REPORT MODEM STATUS CHANGE
```

```
 4849  007670  021512                    .WORD   RPTDLE  ;DATA COMPARISON LENGH ERROR
 4850  007672  021436                    .WORD   RPTDDE  ;DATA COMPARISON DATA ERROR
 4851  007674  021362                    .WORD   RPTEOP  ;END OF PASS
 4852
 4853
 4854  007676  000000            DEV1:   .WORD   0               ;TEMP LOCS TO HOLD DATA FOR EVENT REPORTING
 4855  007700  000000            DEV2:   .WORD   0               ;  AND SHOW MODE,... SUBROUTINE
 4856  007702  000000            DEV3:   .WORD   0
 4857  007704  000000            DEV4:   .WORD   0
 4858
```

```
                        .SBTTL          COMMAND LINE ACTION TREE

                        ;SAMPLE CLI TREE NODE   (ALWAYS AT LEAST 1 WORD)
                        ; ----------------------
                        ; ! ACTION  ! CHAR CODE !
                        ; ----------------------
                        ; !  MISS DISPLACEMENT  !       ONLY IF 'MISS' ARGUMENT DEFINED
                        ; ----------------------
                        ; ! NEXT NODE DISPLMNT  !       ONLY IF 'ASCII' ARGUMENT DEFINED
                        ; ----------------------
                        ; ! ASCIZ MATCH STRING  !       ONLY IF 'ASCII' ARGUMENT DEFINED
                        ; !             (.EVEN) !
                        ; ----------------------

007706                  CLITRE:

                        ;FIRST KEYWORD
007706                          CLI     CLISPA,0,N10$                   ;SKIP ANY LEADING SPACES
007712                  N10$:   CLI     <'?>,HLP,N42$                   ;IS THE FIRST NON-SP CHAR A '?'
007716                          CLI     CLIEXI,0                        ;  IF YES DO 'HLP' AND EXIT
007720                  N42$:   CLI     CLISTR,HLP,N43$,<'HELP'>        ;ELSE, IS FIRST WORD A 'HELP'
007734                          CLI     CLIEXI,0                        ;  IF YES DO 'HLP' AND EXIT
007736                  N43$:   CLI     CLISTR,PRNT,N44$,<'PRINT'>      ;ELSE, IS FIRST WORD A 'PRINT'
007752                          CLI     CLIEXI,0                        ;  IF YES DO 'PRINT' AND EXIT
007754                  N44$:   CLI     CLISTR,EXIT,N45$,<'EXIT'>       ;ELSE, IS FIRST WORD 'EXIT' ;REV B BY EC
007770                          CLI     CLIEXI,0                        ;  IF YES DO 'EXIT' AND EXIT
007772                  N45$:   CLI     CLISTR,RUN,N46$,<'RUN'>         ;ELSE, IS FIRST WORD A 'RUN'
010004                          CLI     CLIBR,0,N80$                    ;  IF YES DO 'RUN' & GOTO N80$
010010                  N46$:   CLI     CLISTR,NOTNUF,N40$,<'DUMP'>     ;ELSE, IS FIRST WORD A 'DUMP'
010024                          CLI     CLIBR,0,N50$                    ;  IF YES GOTO N80$
010030                  N40$:   CLI     CLISTR,CLEAR,N20$,<'CLEAR'>     ;ELSE, IS FIRST WORD A 'CLEAR'
010044                          CLI     CLIBR,NOTNUF,N100$              ;  IF YES DO 'CLR' & GOTO N100$
010050                  N20$:   CLI     <'S>,NOTNUF,N30$                ;ELSE, IS FIRST CHAR. A 'S'
010054                          CLI     CLISTR,SHOW,N25$,<'HOW'>        ;  IF YES IS REST OF WORD 'HOW'
010066                          CLI     CLIBR,0,N100$                   ;    IF YES, DO 'SHOW',BR N100$
010072                  N25$:   CLI     CLISTR,0,N30$,<'ET'>            ;  ELSE, IS REST OF WORD 'ET'
010104                          CLI     CLIBR,0,N110$                   ;    IF YES, DO 'SET', BR N110$
010110                  N30$:   CLI     CLIERR,0                        ;OTHERWISE 'ILL CMD' - EXIT

                        ;SECOND KEYWORD (MODE=) FOR RUN COMMAND

010112                  N80$:   CLI     CLISPA,0,N30$                   ;SKIP LEADING SPS, IF NONE-ERR
010116                  N81$:   CLI     CLISTR,NOTNUF,N30$,<'MODE'>     ;IS NEXT WORD 'MODE='
010132                          CLI     <'=>,0,N30$                     ;  IF NO, IT'S WRONG -ERR -EXIT
010136                          CLI     CLISTR,ATVMOD,N82$,<'ACTIVE'>   ;IS NEXT WORD 'ACTIVE'
010154                          CLI     CLIBR,0,N115$                   ;  IF YES, DO 'ACTIVE',BR N115$
010160                  N82$:   CLI     CLISTR,PASMOD,N83$,<'PASSIVE'>  ;IS NEXT WORD 'PASSIVE'
010176                          CLI     CLIBR,0,N115$                   ;  IF YES, DO 'PASSVE',BR N115$
010202                  N83$:   CLI     CLISTR,RECMOD,N84$,<'RECEIVE'>  ;IS NEXT WORD 'RECEIVE'
010220                          CLI     CLIBR,0,N115$                   ;  IF YES, DO 'RECVE',BR N115$
010224                  N84$:   CLI     CLISTR,LISMOD,N85$,<'LISTEN'>   ;IS NEXT WORD 'LISTEN'
010242                          CLI     CLIBR,0,N115$                   ;  IF YES, DO 'LISTEN',BR N115$
010246                  N85$:   CLI     CLISTR,DLLMOD,N86$,<'DOWNLINELOAD'>     ;IS NEXT WORD 'DOW...'
010272                          CLI     CLIBR,0,N115$                   ;  IF YES, DO 'DWNLL',BR N115$
010276                  N86$:   CLI     <'T>,0,N30$                     ;IS NEXT CHAR A 'T'
```

```
4916  010302                    CLI     CLISTR,TRAMOD,N87$,<'RANSMIT'>  ; IS REST OF WORD 'RANSMIT'
4917  010320                    CLI     CLIBR,0,N115$                   ;  IF YES, DO ''TRANSM'',BR N115$
4918  010324              N87$: CLI     CLISTR,TALMOD,N30$,<'ALK'>      ; IS REST OF WORD ''ALK''
4919  010336                    CLI     CLIBR,0,N115$                   ;  IF YES, DO ''TALK'',BR N115$
4920                                                                    ; IF NO, ERROR - EXIT
4921
4922                            ;SECOND KEYWORD (FOR CLEAR OR SHOW)
4923  010342              N100$: CLI    CLISPA,0,N30$                   ;SKIP LEADING SPACES, NONE=ERR
4924  010346              N102$: CLI    CLISTR,CSHEXP,N104$,<'EXPECTBUFF'>      ;IS NEXT WORD 'EXPE...''
4925  010370                    CLI     CLIEXI,0                        ; IF YES, DO CLR-EXP,EXIT
4926  010372              N104$: CLI    CLISTR,CSHTRN,N30$,<'TRANSMITBUFF'>     ;IS NEXT WORD 'TRANS..''
4927  010416                    CLI     CLIEXI,0                        ; IF YES, DO CLR-TRN,EXIT
4928                                                                    ;IF NO - ERROR - EXIT
4929
4930
4931                            ;SECOND KEYWORD (FOR SET)
4932  010420              N110$: CLI    CLISPA,0,N30$
4933  010424              N111$: CLI    CLISTR,SETEXP,N112$,<'EXPECT'>
4934  010442                    CLI     CLIBR,0,N120$
4935  010446              N112$: CLI    CLISTR,SETTRN,N30$,<'TRANSMIT'>
4936  010466                    CLI     CLIBR,0,N120$
4937
4938                            ;GET ADDRESSES FOR DUMP COMMAND
4939  010472              N50$:  CLI    CLIALP,0,N51$
4940  010476              N51$:  CLI    CLISPA,0,N52$
4941  010502              N52$:  CLI    CLIOCT,DMPS,N30$
4942  010506                     CLI    <'->,NOTNUF,N125$
4943  010512                     CLI    CLIOCT,DMPE,N30$
4944  010516                     CLI    <'/>,NOTNUF,N125$
4945  010522                     CLI    <'B>,DMPQ,N30$
4946  010526                     CLI    CLIBR,0,N125$
4947
4948                            ;QUALIFIERS FOR THE RUN COMMAND
4949  010532              N115$: CLI    CLIALP,0,N114$
4950  010536              N114$: CLI    <'/>,NOTNUF,N125$
4951  010542                     CLI    CLISTR,NO,N116$,<'NO'>
4952  010554              N116$: CLI    <'C>,0,N117$
4953  010560                     CLI    CLISTR,CHECK,N117$,<'HECK'>
4954  010574                     CLI    CLIBR,0,N115$
4955
4963
4964  010600              N117$: CLI    CLISTR,STATUS,N118$,<'STATUS'>
4965  010616                     CLI    CLIBR,0,N115$
4966  010622              N118$: CLI    CLISTR,ECHO,N130$,<'ECHO'>
4967  010636                     CLI    CLIBR,0,N115$
4968
4981
4982
4983  010642              N130$: CLI    CLISTR,0,N132$,<'PASS'>
4984  010656                     CLI    CLIBR,0,N150$
4985
4986
4987  010662              N132$: CLI    CLISTR,MOSC,N131$,<'MODEM'>
4988  010676                     CLI    CLIBR,0,N115$
4989
4990  010702              N131$: CLI    CLISTR,0,N30$,<'LOOP'>
```

```
 4991  010716                          CLI     CLIBR,0,N140$
 4992
 4993                                  ;GET MESSAGE TYPE FOR SET MESSAGE COMMANDS
 4994  010722                          N120$:  CLI     <'=>,0,N30$
 4995
 4996                                  ;   LOOK FOR DEFAULT MESSAGE NAME
 4997  010726                          N60$:   CLI     CLISTR,CMSG1,N61$,<'ONES'>
 4998  010742                                  CLI     CLIBR,0,N121$
 4999  010746                          N61$:   CLI     CLISTR,CMSG0,N62$,<'ZEROES'>
 5000  010764                                  CLI     CLIBR,0,N121$
 5001  010770                          N62$:   CLI     CLISTR,CMSG2,N63$,<'1ALT'>
 5002  011004                                  CLI     CLIBR,0,N121$
 5003  011010                          N63$:   CLI     CLISTR,CMSG3,N64$,<'0ALT'>
 5004  011024                                  CLI     CLIBR,0,N121$
 5005  011030                          N64$:   CLI     CLISTR,CMSG5,N65$,<'ITEP'>
 5006  011044                                  CLI     CLIBR,0,N121$
 5007  011050                          N65$:   CLI     CLISTR,CMSG4,N66$,<'CCITT'>
 5008  011064                                  CLI     CLIBR,0,N121$
 5009  011070                          N66$:   CLI     CLISTR,CMSG6,N67$,<'ALPHA'>
 5010  011104                                  CLI     CLIBR,0,N121$
 5011  011110                          N67$:   CLI     CLISTR,SETET,N68$,<'TRANSMIT'> ;REV B BY EC
 5012  011130                                  CLI     CLIBR,0,N125$
 5013                                  ;   LOOK FOR QUOTED MESSAGE
 5014  011134                          N68$:   CLI     <''>,QPRMSG,N30$
 5015  011140                          N70$:   CLI     <''>,ENDQQ,N71$
 5016  011144                                  CLI     CLIBR,0,N121$
 5017  011150                          N71$:   CLI     CLISPA,0,N72$
 5018  011154                          N72$:   CLI     CLIALN,0,N73$           ;ONLY A-Z,SP,TAB, OR 0-9 BETWEEN '"S
 5019  011160                                  CLI     CLIBR,0,N70$
 5020  011164                          N73$:   CLI     CLIERR,BADCHR           ;PRINT ERROR IF NONE LEGAL CHAR FOR '"S
 5021
 5022                                  ;GET QUALIFIERS (SIZE OR COPY) FOR SET MESSAGE COMMANDS
 5023  011166                          N121$:  CLI     CLIALP,0,N123$
 5024  011172                          N123$:  CLI     <'/>,NOINUF,N125$
 5025  011176                                  CLI     CLISTR,SIZE,N122$,<'SIZE'>
 5026  011212                                  CLI     CLIBR,0,N126$
 5027  011216                          N122$:  CLI     CLISTR,QCOPY,N30$,<'COPY'>
 5028  011232                                  CLI     CLIBR,0,N126$
 5029
 5030                                  ;NUMER FOR SIZE OR COPY
 5031  011236                          N126$:  CLI     <'=>,0,N30$
 5032  011242                                  CLI     CLIDEC,NUM,N30$
 5033  011246                                  CLI     CLIBR,0,N121$
 5034
 5035                                  ;GET MAINTENANCE LOOP TYPE FOR RUN 'LOOP'' QUIALIFIER
 5036  011252                          N140$:  CLI     <'=>,0,N30$
 5037
 5046
 5047  011256                          N141$:  CLI     CLISTR,TTLLOP,N142$,<'INTERNALTTL'>
 5048  011300                                  CLI     CLIBR,0,N115$
 5049  011304                          N142$:  CLI     CLISTR,CBLLOP,N143$,<'CABLE'>
 5050  011320                                  CLI     CLIBR,0,N115$
 5051  011324                          N143$:  CLI     CLISTR,LMDLOP,N144$,<'LOCALMODEM'>
 5052  011346                                  CLI     CLIBR,0,N115$
 5053  011352                          N144$:  CLI     CLISTR,RMDLOP,N30$,<'REMOTEMODEM'>
 5054  011374                                  CLI     CLIBR,0,N115$
```

```
                                                                    I  5
 5055
 5056                                   ;GET LINE NUMBER FOR "PASS" RUN QUALIFIER
 5057  011400                           N150$:  CLI     <'=>,0,N30$
 5058  011404                                   CLI     CLIDEC,PASC,N30$
 5059  011410                                   CLI     CLIBR,0,N115$
 5060
 5061
 5062
 5063                                   ;END-OF-LINE
 5064  011414                           N125$:  CLI     CLIEXI,0
 5065
```

```
5077
5078
5079                                    ;DEVICE DEPENDENT STORAGE LOCATIONS FOR
5080                                    ;  CURRENT DEVICE PARAMTERS
5081
5082
5083  011416  000000                    RXCSR:  .WORD   0               ;REC CONTROL AND STATUS
5084  011420  000000                    PCSAR:  .WORD   0               ;STATUS REGISTIER
5085  011422  000000                    RDSR:   .WORD   0               ;REC DATA AND STATUS REG
5086  011424  000000                    TXCSR:  .WORD   0               ;TRANSMIT AND REC. CONTROL
5087  011426  000000                    TDSR:   .WORD   0               ;TRANSMIT DATA AND STATUS REG
5088
5089
5090  011430  000000                    INVEC:  .WORD   0               ;INPUT INTERRUPT VECTOR ADDRESS
5091  011432  000000                    OUTVEC: .WORD   0               ;OUTPUT INTERRUPT VECTOR ADDRESS
5092  011434  000000                    INTPRI: .WORD   0               ;INTERRUPT PRIORITY
5093  011436  000000                    OPTYP:  .WORD   0               ;DEVICE OPTION TYPE(0=DMC,5=DMR-DMC MODE
5094
5095  011440  065626                    DPVP1:  .WORD   065626          ;THIS WORD IS BROKEN DOWN AS FOLLOWS
5096                                                                    ;BITS 0-7 =SYNC WORD
5097                                                                    ;BITS 8-10=ERR DET SELECTED
5098                                    ;THIS IS SET TO SYNC 262        ;BIT11 = IDLE
5099                                    ;CRC 16 INIT TO 1               ;BIT12 = SEC ADDR. MODE
5100                                    ;STRIP SYNC AND BCP MODE        ;BIT13 = STRIP SYNC
5101                                    ;IDLE SET TO MARK               ;BIT14 = PORTO TYPE SEL(1=BCP 0=BOP)
5102                                                                    ;BIT15 = ALL PARTIES ADDRESS..
5103
5104
5105  011442  000000                    CMODS:  .WORD   0               ;CURRENT MODEM
5106  011444  000000                    IRXCSR: .WORD   0               ;IMAGE OF RXCSR
5107  011446  000000                    IRDSR:  .WORD   0               ;IMAGE OR RDSR
5108  011450  000000                    MSGPTR: .WORD   0               ;MSG PTR.FOR HEADER OR CONTROL
5109  011452  000000                    MSGCC:  .WORD   0               ;MSG COUNTER OR CC
5110  011454  000000                    SYNCC:  .WORD   0               ;SYNC CHAR COUNT.
5111  011456  000000                    SYNCW:  .WORD   0               ;SYNC WORD.PLUS TSOM BIT.
5112  011460  000000                    RMSGPT: .WORD   0               ;MSG PTR FOR REC
5113  011462  000000                    RMSGCC: .WORD   0               ;CHAR COUNTER FOR REC
5114  011464  000000                    BCCW:   .WORD   0               ;CRC HOLDING LOC.
5115  011466  000000                    MGLCNT: .WORD   0               ;COUNT OF GLITCH ERRORS
5116  011470  000000                    MHRCNT: .WORD   0               ;COUNT OF HARD ERRORS
5117  011472  000000                    RNODE:  .WORD   0               ;1=REMOTE NODE ITEP,0=NON ITEP
5130
5140
5141                                    ;       ERRTBL
```

```
                                                    K 5
5143                            .SBTTL  GLOBAL TEXT SECTION
5144
5145                            ;++
5146                            ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
5147                            ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
5148                            ; MORE THAN ONE TEST.
5149                            ;--
5150
5151                            .SBTTL          DEVICE SUPPORTED
5152                            ;
5153                            ; NAMES OF DEVICES SUPPORTED BY PROGRAM
5154                            ;
5155
5164
5165  011474                            DEVTYP  <DPV-11>
 (4)  011474                                                        L$DVTYP::
 (3)  011474  050104  026526  030461                                        .ASCIZ  /DPV-11/
 (3)  011502     000
 (2)          011504                                                       .EVEN
5166
5172
5173                            .SBTTL          PROGRAM IDENTIFICATION
5174                            ; TEST DESCRIPTION
5175                            ;
5176  011504                            DESCRIPT        <DPV-11 DATA COMM LINK TEST >
 (4)  011504                                                        L$DESC::
 (3)  011504  050104  026526  030461                                        .ASCIZ  /DPV-11 DATA COM
 (3)  011512  042040  052101  020101
 (3)  011520  047503  046515  046040
 (3)  011526  047111  020113  042524
 (3)  011534  052123  000040
 (2)                                                                      .EVEN
5177                                    .EVEN
5178
5185
5186
5187
```

```
                                        .SBTTL          GLOBAL FORMAT STATEMENTS, MESSAGES, AND ASCII INFO

011540 041504 052114 000076  CLI$PM: .ASCIZ  /DCLT>/
011546 050122 037124    000  CLI$RP: .ASCIZ  /RPT>/          ;REV B BY EC
011553    045 022516 037501  CLIERM: .ASCIZ  /%N%A?ILL CMD-BAD SYNTX?/
011603    045 022516 037501  CLINUF: .ASCIZ  /%N%A?INCMPLTE CMD?/
011626 047045 040445 047077  CLINBG: .ASCIZ  /%N%A?NUM TOO BIG?/
011650 047045 040445 041077  CLIBRX: .ASCIZ  /%N%A?BAD RADIX?/
011670 047045 040445 021077  CLIBDL: .ASCIZ  /%N%A?''LOOP'' VALID ONLY IN ACTIVE?/
011732 047045 040445 021077  CLINPS: .ASCIZ  /%N%A?''ECHO'' VALID ONLY IN PASSIVE?/
011775    045 022516 037501  CLIBCR: .ASCIZ  /%N%A?ILL CHR- ''A-Z,0-9,SP,TAB'' ONLY?/
012042 047045 040445 021077  CLISEQ: .ASCIZ  /%N%A?''SIZE=0'' NOT VALID?/
012073    045 022516 037501  CLIPW:  .ASCIZ  /%N%A?TRANSMIT & EXPECT LIST MUST BE IDENTICAL FOR LOOP?/;REV B EC
012163    045 022516 052101  HLP0:   .ASCIZ  /%N%ATHIS IS DCLT. TYPE ''H'' OR ''?'' FOR DETAILS/
012241    045 022516 000124  HLPF:   .ASCIZ  /%N%T/
012246 041504 052114 041440  HLP1:   .ASCIZ  /DCLT CMDS:/
012261    040 046103 040505  HLP2:   .ASCII  / CLEAR OR SHOW  EXPECTLIST OR TRANSMITLIST/<15><12>
012335    040 051120 047111          .ASCII  / PRINT/<15><12>
012345    040 054105 052111          .ASCII  / EXIT/<15><12> ;REV B EC
012354 042040 046525 020120          .ASCIZ  ? DUMP START-END/B?
012376 051440 052105 042440  HLP3:   .ASCIZ  ? SET EXPECTMSG OR TRANSMITMSG=TYPE/SIZE=N OR /COPY=N?
012463    040 042523 020124  HLP3A:  .ASCIZ  / SET EXPECT=TRANSMIT/  ;REV B EC
012510 020040 052040 050131  HLP4:   .ASCIZ  ?   TYPE=ONES,ZEROES,1ALT,0ALT,ITEP,CCITT,ALPHA?
012567    040 020040 020040  HLP4A:  .ASCIZ  /       OR 'OPR SPCD=A-Z,SP,TAB,0-9 IN QUOTES''/
012645    040 052522 020116  HLP5:   .ASCIZ  ? RUN MODE=MTYP/LOOP=LTYP/CHECK,STATUS,ECHO,MODEM,PASS=N?
012735    040 020040 052115  HLP6:   .ASCII  /   MTYP=TRAN,REC,ACT,PAS,TAL,LIS,DOWN/<15><12>
013004 020040 046040 054524          .ASCIZ  /   LTYP=INT,CAB,LOC,REM/

013034 047045 040445 054524  RHLP0:  .ASCIZ  /%N%ATYPE ''H'' OR ''?'' FOR HELP!/ ;REV B EC
013072 041504 052114 051040  RHLP1:  .ASCIZ  /DCLT REPORT CMDS:/ ;REV B EC
013114 047514 020107 020055  RHLP2:  .ASCIZ  /LOG - PRINT DCLT EVENT LOG/  ;REV B EC
013147    105 044530 020124  RHLP3:  .ASCIZ  /EXIT - EXIT REPORT LEVEL/  ;REV B EC
013200 042510 050114 026440  RHLP4:  .ASCIZ  /HELP - PRINT THIS MESSAGE/  ;REV B EC

013232 047045 040445 051515  SHMSG:  .ASCIZ  ?%N%AMSG: TYPE=%T%A/SIZE=%D3?
013266 042532 047522 051505  SHTYP0: .ASCIZ  /ZEROES/
013275    117 042516 000123  SHTYP1: .ASCIZ  /ONES/
013302 040461 052114    000  SHTYP2: .ASCIZ  /1ALT/
013307    060 046101 000124  SHTYP3: .ASCIZ  /0ALT/
013314 041503 052111 000124  SHTYP4: .ASCIZ  /CCITT/
013322 052111 050105    000  SHTYP5: .ASCIZ  /ITEP/
013327    101 050114 040510  SHTYP6: .ASCIZ  /ALPHA/
013335    117 051120 051440  SHTYP7: .ASCIZ  /OPR SPEC/
013346 042522 042503 053111  MO0:    .ASCIZ  /RECEIVE/
013356 051124 047101 046523  MO1:    .ASCIZ  /TRANSMIT/
013367    120 051501 044523  MO2:    .ASCIZ  /PASSIVE/
013377    101 052103 053111  MO3:    .ASCIZ  /ACTIVE/
013406 047504 047127 044514  MO4:    .ASCIZ  /DOWNLINELOAD/
013423    124 046101 000113  MO5:    .ASCIZ  /TALK/
013430 044514 052123 047105  MO6:    .ASCIZ  /LISTEN/
013437    000              LP0:    .ASCIZ  //
013440 046057 047517 036520  LP00:   .ASCIZ  ?/LOOP=?
013447    111 052116 051105  LP1:    .ASCIZ  ?INTERNAL?
013460 040503 046102 000105  LP2:    .ASCIZ  ?CABLE?
013466 047514 040503 046514  LP3:    .ASCIZ  ?LOCALMODEM?
013501    122 046505 052117  LP4:    .ASCIZ  ?REMOTEMODEM?
```

```
5248  013515  116  117                PNST:   .ASCII  /NO/
5249  013517  123  040524  052524     PST:    .ASCIZ  /STATUS/
5250  013526  047516                   PNCK:   .ASCII  /NO/
5251  013530  044103  041505  000113   PCK:    .ASCIZ  /CHECK/
5252  013536  047516                   PNEC:   .ASCII  /NO/
5253  013540  041505  047510  000      PEC:    .ASCIZ  /ECHO/
5254  013545  116  117                PNMS:   .ASCII  /NO/
5255  013547  115  042117  046505     PMS:    .ASCIZ  /MODEM/
5256
5257
5268
5269  013555  045  022516  046101     LISP:   .ASCIZ  /%N%ALIS>/
5270  013566  046124  037113  000      OPRMM:  .ASCIZ  /TLK>/
5271  013573  124  044510  020123     L5060:  .ASCIZ  /THIS A 50. OR 60. HZ. LSI-11:/
5272          013632                   .EVEN
5273
5274
5275
5276                                   ;
5277                                   ; FORMAT STATEMENTS USED IN PRINT CALLS
5278                                   ;
5279
5293
5294  013632  047045  040445  047504  DLLCM:  .ASCIZ  /%N%ADOWN LINE LOAD NOT SUPPORTED BY THIS DEVICE/
5295
5296
5297  013712  047045  040445  046103  BDCLK:  .ASCIZ  /%N%ACLOCK NOT FOUND/
5298  013736  047045  040445  040502  NOCLK:  .ASCIZ  /%N%ABAD CLOCK - PROGRAM WILL HANG ON ''TIMEOUT''!!/
5299  014017  115  054101  020056     TABEX:  .ASCIZ  /MAX. CHAR. MSG COUNT EXCEEDED -/
5300  014057  102  043125  042506     BUFEX:  .ASCIZ  /BUFFER FULL -/
5301  014075  045  022516  022524     MSGTRN: .ASCIZ  /%N%T%A MSG. NOT BUILT !!/
5302  014126  047045  040445  044103  MSGTRU: .ASCIZ  /%N%ACHAR. COUNT EXCEEDS BUFF LIMIT - MSG TRUNCATED/
5303  014211  045  022516  032523     SHF0:   .ASCIZ  ?%N%S5%AMODE=%T%T%T%A/PASS=%Z5?
5309
5310  014247  045  022516  032523     SHF1:   .ASCIZ  ?%N%S5%S5%S5%A/%T%A/%T%A/%T%A/%T?
5311
5312  014307  045  032523  040445     EFM2:   .ASCIZ  /%S5%ATOTAL MISMATCHES IN MSG = %D5/
5313  014352  047045  051445  022463  PCPM:   .ASCIZ  /%N%S3%ACALLED FROM PC=%O6/
5314  014404  051445  022465  041501  EFM11:  .ASCIZ  /%S5%ACOMPARE COUNT=%D5%S3%ARECEIVE COUNT=%D5/
5315  014461  115  042117  046505     MSCMS:  .ASCIZ  /MODEM STATUS CHANGES FOR THIS PASS WERE../
5316  014533  045  032523  040445     EFM13:  .ASCIZ  /%S5%AHARD CHANGES=%D5%A%S3%AGLITCHES=%D5/
5317
5318
5319                                   ;EVENT DESCRIPTION MESSAGES
5320
5321  014604  051124  047101  046523  EDTXQ:  .ASCIZ  /TRANSMIT MSG QUEUED/
5322  014630  051124  047101  046523  EDTXC:  .ASCIZ  /TRANSMIT MSG COMPLETED/
5323  014657  122  041505  044505     EDRXQ:  .ASCIZ  /RECEIVE SPACE QUEUED/
5324  014704  042522  042503  053111  EDRXC:  .ASCIZ  /RECEIVE MSG COMPLETED/
5325  014732  042504  044526  042503  EDDER:  .ASCIZ  /DEVICE ERROR/
5326  014747  104  052101  020101     EDDCK:  .ASCIZ  /DATA COMPARISON STARTED/
5327  014777  104  053105  041511     EDDVI:  .ASCIZ  /DEVICE INIT AND SETUP/
5328  015025  104  052101  020101     EDDLE:  .ASCIZ  /DATA COMPARISON LENGTH ERROR/
5329  015062  040504  040524  041440  EDDDE:  .ASCIZ  /DATA COMPARISON DATA ERROR/
5330  015115  105  042116  047440     EDEOP:  .ASCIZ  /END OF PASS/
5331  015131  115  042117  046505     EDMOS:  .ASCIZ  /MODEM STATUS CHANGE/
```

```
                                          ;EVENT REPORTING MESSAGES
015155     045  031523  047445  BASM3:   .ASCIZ  /%S3%O3/
015164  051445  022463  033117  BASM2:   .ASCIZ  /%S3%O6/
015173     045  022516  033117  BASM1:   .ASCIZ  /%N%O6/
015201     045  022516  052101  NULEVT:  .ASCIZ  /%N%ATHE DCLT EVENT LOG IS EMPTY/
015241     045  022516  037101  EVTF0:   .ASCIZ  /%N%A>>> DCLT EVENT LOG ENTRY <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<</
015335     045  022516  032504  EVTF1:   .ASCIZ  /%N%D5%A:%Z2%A:%Z2%S3%T/
015364  047045  051445  022463  EVTF2:   .ASCIZ  /%N%S3%AADDR OF MSG=%O6%S3%ABYTE COUNT=%D5/
015436  047045  051445  022463  EVTF3:   .ASCIZ  /%N%S3%T%N/
015450  051445  022463  033117  EVTF3C:  .ASCIZ  /%S3%O6%S3%O6/
015465     045  031523  047445  EVTF3D:  .ASCIZ  /%S3%O6%S3%O6%S3%T/
015507     045  022516  031523  EVTF4:   .ASCIZ  /%N%S3%AADDR OF MSG=%O6%S3%ABYTE COUNT=%D5%S3%ANO. OF CMP ERRS=%D5/
015611     045  022516  031523  EVTF4A:  .ASCIZ  /%N%S3%AADDR OF MSG=%O6%S3%ARX BYTES=%D5%S3%ACOMPARE BYTES=%D5/

015707     045  022516  031523  EVTF4B:  .ASCIZ  /%N%S3%APASS=%D5%S3%AERRORS=%D5%S3%ASTRT-TO=%D5/

015766  051445  022465  041101  EVTF5A:  .ASCIZ  /%S5%ABYTE# IN MSG.=%D5%S3%AEXPTD=%O3%S3%ARECVD=%O3/

016051     045  022516  034523  EVMOCG:  .ASCIZ  /%N%S9%ACHANGED TO:/

                                          ; ***********************************************************************
                                          ;DO NOT SEPERATE THE NEXT LIST OF MESSAGES - MODEM SIGNAL HEADER AND REPORT

016074  047045  051445  022470  EVMOHD:  .ASCIZ  /%N%S8%AMODEM STATUS: CTS DSR DCD RTS RI  SQD TM/
016154  047045  051445  022471  EVMOST:  .ASCII  /%N%S9%S9%S5%A/
016171     130     040     040  EVMCTS:  .BYTE   'X,40,40,40
016175     130     040     040  EVMDSR:  .BYTE   'X,40,40,40
016201     130     040     040  EVMDCD:  .BYTE   'X,40,40,40
016205     130     040     040  EVMRTS:  .BYTE   'X,40,40,40
016211     130     040     040  EVMRI:   .BYTE   'X,40,40,40
016215     130     040     040  EVMSQD:  .BYTE   'X,40,40,40
016221     130     040     040  EVMTM:   .BYTE   'X,40,40,40
016225     000                           .BYTE   0
                                         .EVEN

                                          ;EXECUTION STATUS MESSAGES TO BE PRINTED TO KEEP OPERATOR AWAKE
016226  047045     000          CR:      .ASCIZ  /%N/           ;CR FOR LINES IN A ROW
016231     045  031523  040445  STXQ:    .ASCIZ  /%S3%ATXQ/     ;ABOUT TO TRANSMIT
016242  051445  022463  052101  STXC:    .ASCIZ  /%S3%ATXC/     ;TX COMPLETED
016253     045  031523  040445  SRXQ:    .ASCIZ  /%S3%ARXQ/     ;ABOUT TO RECEIVE
016264  051445  022463  042501  SDVE:    .ASCIZ  /%S3%AERR/     ;DEVICE ERROR
016275     045  031523  040445  SCM:     .ASCIZ  /%S3%ACMP/     ;ABOUT TO DO DATA CHECKING OF RECVD VS. EXPTD
016306  051445  022463  044501  SDVI:    .ASCIZ  /%S3%AINI/     ;DEVICE ABOUT TO BE INITIALIZED
016317     045  031523  040445  SCML:    .ASCIZ  /%S3%ACML/     ;COMPARE LENGTH ERROR
016330  051445  022463  041501  SCMD:    .ASCIZ  /%S3%ACMD/     ;COMPARE DATA ERROR
016341     045  031523  040445  SEOP:    .ASCIZ  /%S3%AEOP/     ;END  OF PASS
                                         .EVEN

016352  051445  022463  046501  SMSC:    .ASCIZ  /%S3%AMSC/     ;MODEM STATUS CHANGE.

016363     115  042117  046505  GLMSG:   .ASCIZ  /MODEM STATUS GLITCHED/
016411     115  042117  046505  HRDMSG:  .ASCIZ  /MODEM STATUS HARD ERROR/
```

```
 5402
 5421
 5422  016441     115  051501  042524  DVEM0:  .ASCII  /MASTER RESET DID NOT WORK/
 5423  016472  005015  020040  051040          .ASCIZ  <15><12>/   RXCSR     TXCSR  /
 5424  016520  047516  041440  042514  DVEM1:  .ASCII  /NO CLEAR TO SEND FROM MODEM /
 5425  016554  005015  020040  051040          .ASCIZ  <15><12>/   RXCSR     TXCSR  /
 5426  016602  044524  042515  047440  DVEM2:  .ASCII  /TIME OUT WAITING FOR RX OR TX TO COMPLETE/
 5427  016653     015  020012  020040          .ASCIZ  <15><12>/   RXCSR     TXCSR/
 5428  016677     103  041522  044440  DVEM3:  .ASCII  /CRC IN ERROR/
 5429  016713     015  020012  020040          .ASCIZ  <15><12>/   RDSR      RXCSR/
 5430  016737     122  041505  044505  DVEM4:  .ASCII  /RECEIVER OVERRUN/
 5431  016757     015  020012  020040          .ASCIZ  <15><12>/   RDSR      RXCSR/
 5432  017003     124  046511  042105  DVEM5:  .ASCII  /TIMED OUT IN START,STACK,ACK SEQ/
 5433  017043     015  020012  020040          .ASCIZ  <15><12>/   RDATA     SDATA/
 5434  017067     115  042117  046505  DVEM6:  .ASCII  /MODEM DID NOT RETURN MODEM READY/
 5435  017127     015  020012  020040          .ASCIZ  <15><12>/   RXCSR     TXCSR/
 5436
 5437
 5438
 5449
 5450                                          .EVEN
 5451
 5462
 5463
 5471
 5472
 5473
 5474
```

```
                        .SBTTL  GLOBAL ERROR REPORT SECTION

                        ;++
                        ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
                        ; USED BY MORE THAN TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
                        ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
                        ;--


5503  017154                    BGNMSG  ERR1
  (3) 017154                                                        ERR1::
5504  017154                    PRINTB  #EVTF5A,OFSET,<B,GOOD>,<B,BAD>  ;INDIVIDUAL DATA COMPARE ERROR
 (10) 017154  005046                                                        CLR     -(SP)
 (10) 017156  153716  006551                                                BISB    BAD,(SP)
  (9) 017162  005046                                                        CLR     -(SP)
  (9) 017164  153716  006550                                                BISB    GOOD,(SP)
  (8) 017170  013746  006526                                                MOV     OFSET,-(SP)
  (7) 017174  012746  015766                                                MOV     #EVTF5A,-(SP)
  (6) 017200  012746  000004                                                MOV     #4,-(SP)
  (3) 017204  010600                                                        MOV     SP,R0
  (4) 017206  104414                                                        TRAP    C$PNTB
  (4) 017210  062706  000012                                                ADD     #12,SP
5505  017214                    ENDMSG
  (3) 017214                                                    L10001:
  (3) 017214  104423                                                        TRAP    C$MSG
5506
5507  017216                    BGNMSG  ERR2
  (3) 017216                                                        ERR2::
5508  017216                    PRINTB  #EFM2,TEMP4                 ;TOTAL DATA COMPARE FAILS ERROR
  (8) 017216  013746  006540                                                MOV     TEMP4,-(SP)
  (7) 017222  012746  014307                                                MOV     #EFM2,-(SP)
  (6) 017226  012746  000002                                                MOV     #2,-(SP)
  (3) 017232  010600                                                        MOV     SP,R0
  (4) 017234  104414                                                        TRAP    C$PNTB
  (4) 017236  062706  000006                                                ADD     #6,SP
5509  017242                    ENDMSG
  (3) 017242                                                    L10002:
  (3) 017242  104423                                                        TRAP    C$MSG
5510
5511  017244                    BGNMSG  ERR10
  (3) 017244                                                        ERR10::
5512  017244                    PRINTB  #EFM11,R4,TEMP3             ;LENGH COMPARISON ERROR
  (9) 017244  013746  006536                                                MOV     TEMP3,-(SP)
  (8) 017250  010446                                                        MOV     R4,-(SP)
  (7) 017252  012746  014404                                                MOV     #EFM11,-(SP)
  (6) 017256  012746  000003                                                MOV     #3,-(SP)
  (3) 017262  010600                                                        MOV     SP,R0
  (4) 017264  104414                                                        TRAP    C$PNTB
  (4) 017266  062706  000010                                                ADD     #10,SP
5513  017272                    ENDMSG
  (3) 017272                                                    L10003:
  (3) 017272  104423                                                        TRAP    C$MSG
5514
```

```
 5515  017274                                BGNMSG  ERR4                                    ERR4::
   (3) 017274
 5516  017274                                PRINTB  #EFM13,MHRCNT,MGLCNT          ;MODEM STATUS CHANGE
   (9) 017274  013746  011466                                                              MOV     MGLCNT,-(SP)
   (8) 017300  013746  011470                                                              MOV     MHRCNT,-(SP)
   (7) 017304  012746  014533                                                              MOV     #EFM13,-(SP)
   (6) 017310  012746  000003                                                              MOV     #3,-(SP)
   (3) 017314  010600                                                                      MOV     SP,R0
   (4) 017316  104414                                                                      TRAP    C$PNTB
   (4) 017320  062706  000010                                                              ADD     #10,SP
 5517  017324                                ENDMSG                                  L10004:
   (3) 017324
   (3) 017324  104423                                                                      TRAP    C$MSG
 5518
 5519
 5529
 5530                                        ;
 5531                                        ;PRINT THE 2 OCTAL #'S IN TEMP3/4
 5532                                        ;
 5533
 5534  017326                                BGNMSG  ERR13                                   ERR13::
   (3) 017326
 5535  017326                                PRINTB  #EVTF3C,TEMP3,TEMP4
   (9) 017326  013746  006540                                                              MOV     TEMP4,-(SP)
   (8) 017332  013746  006536                                                              MOV     TEMP3,-(SP)
   (7) 017336  012746  015450                                                              MOV     #EVTF3C,-(SP)
   (6) 017342  012746  000003                                                              MOV     #3,-(SP)
   (3) 017346  010600                                                                      MOV     SP,R0
   (4) 017350  104414                                                                      TRAP    C$PNTB
   (4) 017352  062706  000010                                                              ADD     #10,SP
 5536  017356                                ENDMSG                                  L10005:
   (3) 017356
   (3) 017356  104423                                                                      TRAP    C$MSG
 5537
 5538                                        ;
 5539                                        ;PRINT THE 2 OCTAL #'S IN TEMP3/4
 5540                                        ; AND THE MESG. WHOSE ADDR. IS IN CONOTM
 5541                                        ;
 5542
 5543  017360                                BGNMSG  ERR14                                   ERR14::
   (3) 017360
 5544  017360                                PRINTB  #EVTF3D,TEMP3,TEMP4,CONOTM
  (10) 017360  013746  006544                                                              MOV     CONOTM,-(SP)
   (9) 017364  013746  006540                                                              MOV     TEMP4,-(SP)
   (8) 017370  013746  006536                                                              MOV     TEMP3,-(SP)
   (7) 017374  012746  015465                                                              MOV     #EVTF3D,-(SP)
   (6) 017400  012746  000004                                                              MOV     #4,-(SP)
   (3) 017404  010600                                                                      MOV     SP,R0
   (4) 017406  104414                                                                      TRAP    C$PNTB
   (4) 017410  062706  000012                                                              ADD     #12,SP
 5545  017414                                ENDMSG                                  L10006:
   (3) 017414
   (3) 017414  104423                                                                      TRAP    C$MSG
 5546
 5547  017416                                EXIT    MSG
   (4) 017416  000167                                                                      .WORD   J$JMP
```

```
  (3)  017420  177772                                          .WORD   L10006-2-.
5548
5549
```

```
5551                                    .SBTTL  GLOBAL SUBROUTINES SECTION
5552
5553                                    ;++
5554                                    ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
5555                                    ; THAT ARE USED IN MORE THAN ONE TEST.
5556                                    ;--
5557
5634
5635                                    .SBTTL          CLOCK SETUP SUBROUTINE
5636
5637                                    ;++
5638                                    ; FUNCTIONAL DESCRIPTION:
5639                                    ;       THIS SUBROUTINE SETS UP THE CLOCK INFORMATION TABLE FOLLOWING A ''CLOCK''
5640                                    ;       CALL EXECUTED IN THE INITIALIZATION CODE. BUT SINCE THE ''CLOCK'' CALL
5641                                    ;       SAYS NOTHING ABOUT AN LSI-11'S CLOCK, THIS ROUTINE IS ONLY USED IF A
5642                                    ;       LINE OR P-CLOCK IS FOUND.
5643                                    ;
5644                                    ; INPUTS:
5645                                    ;       R1= POINTS TO SUPERVISOR SPACE WHERE CLOCK INFO WAS RETURNED
5646                                    ;       R2= POINTS TO ''CLK'' TABLE WHERE CLOCK INFO WILL BE KEPT
5647                                    ;
5648                                    ; IMPLICIT INPUTS:
5649                                    ;       THE SUPERVISOR  SPACE WHERE CLOCK INFO WAS RETURNED BY THE ''CLOCK'' CALL
5650                                    ;
5651                                    ; OUTPUTS:
5652                                    ;       ''CLKCSR'' GETS LOADED WITH THE CLOCK'S CSR ADDRESS
5653                                    ;       ''CLKBR'' GETS LOADED WITH THE CLOCK'S INTERRUPT LEVEL
5654                                    ;       ''CLKVEC'' GETS LOADED WITH THE CLOCK'S INTERRUPT VECTOR
5655                                    ;       ''CLKHZ'' GETS LOADED WITH THE LINE FREQ. (HERTZ RATE) WHICH DETERMINES
5656                                    ;               THE NUMBER OF TICKS IN A SECOND
5657                                    ;
5658                                    ; CALLING SEQUENCE:
5659                                    ;       JSR     PC,CLKSET               ;CALL CLOCK SETUP WITH R1 & R2 SETUP
5660                                    ;--
5661
5662   017422                           CLKSET:
5663   017422  012122                           MOV     (R1)+,(R2)+             ;LOAD CLOCK'S CSR ADDR. INTO ''CLKCSR''
5664   017424  012112                           MOV     (R1)+,(R2)              ;LOAD CLOCK'S INT. LEVEL INTO ''CLKBR''
5665   017426  006312                           ASL     (R2)                    ;ADJUST THE INT. LEVEL FOR LOADING INTO
5666   017430  006312                           ASL     (R2)                    ;    THE PSW WITH A ''SETVEC'' CALL
5667   017432  006312                           ASL     (R2)
5668   017434  006312                           ASL     (R2)
5669   017436  006322                           ASL     (R2)+
5670   017440  012122                           MOV     (R1)+,(R2)+             ;LOAD CLOCK'S INT. VECTOR INTO ''CLKVEC''
5671   017442  012122                           MOV     (R1)+,(R2)+             ;LOAD CLOCK'S HERTZ RATE INTO ''CLKHZ''
5672   017444  000207                           RTS     PC
5673
```

```
5675
5676                                    .SBTTL          CLOCK INTERRUPT SERVICE ROUTINE
5677                                    ;++
5678                                    ; FUNCTIONAL DESCRIPTION:
5679                                    ;       THIS IS THE CLOCK INTERRUPT SERVICE ROUTINE WHICH TAKES CARE OF
5680                                    ;       KEEPING THE ''TIME-SINCE-START'' AND COUNTING DOWN ANY OF THE
5681                                    ;       ''EVENT'' TIMERS. THE TIMERS ARE USED TO TIME COMPLETION OF  DEVICE
5682                                    ;       REQUESTS. THE ''TIME-SINCE-START'' IS USED TO BE LOGGED WITH EACH ENTRY
5683                                    ;       INTO THE EVENT LOG.
5684                                    ;
5685                                    ; IMPLICIT INPUTS:
5686                                    ;       TIMTCK: THE CURRENT NO. OF TICKS LEFT TO BE COUNTED UNTIL A SECOND
5687                                    ;               HAS BEEN COUNTED OFF
5688                                    ;       CLKHZ:  THE NO. OF TICKS IN A SECOND, DETERMINED BY THE SYS. LINE FREQ.
5689                                    ;       TIMMIN & TIMSEC: CURRENT VALUE OF ''TIME-SINCE-START''
5690                                    ;                               IN MINUTES & SECONDS
5691                                    ;       TIMER 1,2, & S: CURRENT VALUES OF THE ''EVENT TIMERS''
5692                                    ;
5693                                    ; IMPLICIT OUTPUTS:
5694                                    ;       NEW VALUE OF EVENT TIMER ''1'' DECREMENTED BY 1 TICK IF IT WAS NON-ZERO
5695                                    ;       NEW VALUE OF EVENT TIMER ''2'' DECREMENTED BY 1 TICK IF IT WAS NON-ZERO
5696                                    ;       NEW VALUE OF EVENT TIMER ''S'' DECREMENTED BY 1 SECOND IF IT WAS NON-ZERO
5697                                    ;
5698                                    ; FUNCTIONAL SIDE EFFECTS:
5699                                    ;       THE CLOCK IS DISABLED UPON ENTRY AND REENABLED WHEN LEAVING
5700                                    ;
5701                                    ; CALLING SEQUENCE:
5702                                    ;       THIS ROUTINE IS CALLED WHEN THE CLOCK INTERRUPTS THRU ''CLKVEC''.
5703                                    ;       THE ADDRESS OF THIS ROUTINE WAS LOADED INTO THE CLOCK'S INTERRUPT
5704                                    ;       VECTOR WITH A SUPERVISOR ''SETVEC'' CALL.
5705                                    ;--
5706
5707  017446                                    BGNSRV  CLKINT
 (3)  017446                                                                              CLKINT::
5708
5709  017446  005077  167142                    CLR     @CLKCSR         ;DISABLE THE CLOCK FORM INTERRUPTING
5710  017452  005337  006632                    DEC     TIMTCK          ;DECREMENT THE # OF TICKS/SEC.
5711  017456  001015                            BNE     1$              ;GO CHECK TIMERS (1&2-TICKS, 3-SECONDS)
5712  017460  013737  006622  006632            MOV     CLKHZ,TIMTCK    ;RESET THE # OF TICKS/SEC.
5713  017466  005237  006630                    INC     TIMSEC          ;INC # OF SECS-SINCE-START
5714  017472  022737  000074  006630            CMP     #60.,TIMSEC     ;SEE IF WE'VE COUNTED 60 SECS. YET
5715  017500  001004                            BNE     1$              ;IF NOT, GO CHECK TIMERS
5716  017502  005237  006626                    INC     TIMMIN          ; ELSE INC MINUTES-SINCE-START
5717  017506  005037  006630                    CLR     TIMSEC          ; AND RESTART SECOND COUNTER
5718
5719  017512  005737  006634            1$:     TST     TIMER1          ;SEE IF TIMER #1, TIMING ANYTHING
5720  017516  001402                            BEQ     2$              ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
5721  017520  005337  006634                    DEC     TIMER1          ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
5722  017524  005737  006636            2$:     TST     TIMER2          ;SEE IF TIMER #2, TIMING ANYTHING
5723  017530  001402                            BEQ     3$              ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
5724  017532  005337  006636                    DEC     TIMER2          ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
5725  017536  005737  006640            3$:     TST     TIMERS          ;SEE IF TIMER #3, TIMING ANYTHING
5726  017542  001406                            BEQ     4$              ; IF=0, NOTHING BEING TIMED, LEAVE
5727  017544  023737  006622  006632            CMP     CLKHZ,TIMTCK    ;SEE IF A SECOND HAS BEEN COUNTED OFF
5728  017552  001002                            BNE     4$              ; BR IF NO
5729  017554  005337  006640                    DEC     TIMERS          ; ELSE DECREMENT THE TIMER VALUE (BY 1 SEC.)
```

```
 5730  017560  013777  006624  167026  4$:     MOV     CLKEN,@CLKCSR   ;REENABLE THE CLOCK TO INTERRUPT
 5731  017566                                  ENDSRV
  (3)  017566                                                          L10007:
  (2)  017566  000002                                                          RTI
```

```
                        .SBTTL          EVENT LOG SUBROUTINES

                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       THIS SUBROUTINE HAS A DIFFERENT ENTRY POINT
                        ;       FOR EACH EVENT TO BE LOGGED AND ALWAYS PRINTS
                        ;       THE SHORT "OPERATOR AWAKE" MESSAGE TO CONSOLE THEN LOGS THE
                        ;       EVENT TYPE, TIME, AND THE OTHER 3 WORDS OF INFO PASSED TO THE
                        ;       SUBROUTINE AT CALLING TIME
                        ;
                        ; INPUTS:
                        ;       TIMMIN & TIMSEC:        CURRENT VALUE OF "TIME-SINCE-START"
                        ;       TEMP2:  WORD #1 OF EVENT LOG INFORMATION (FOR MOST EVENT TYPES)
                        ;       TEMP3:  WORD #2 OF EVENT LOG INFORMATION
                        ;       TEMP4:  WORD #3 OF EVENT LOG INFORMATION
                        ;       MODS:   CURRENT VALUE OF THE MODEM SIGNALS AVAILABLE FROM THE DEVICE
                        ;
                        ; OUTPUTS:
                        ;       "OPERATOR AWAKE" MESSAGE SENT TO THE CONSOLE
                        ;       NEW EVENT LOGGED IN "EVTLOG" (EVENT LOG)
                        ;       UPDATED "EVTPTR" (EVENT LOG ENTRY POINTER)
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;       "DVMODS" THE DEVICE SUBROUTINE THAT RETURNS MODEM STATUS IN "MODS"
                        ;               (FOR SOME EVENT TYPES)
                        ;
                        ; FUNCTIONAL SIDE EFFECTS:
                        ;       TEMP:   USED TO STORE ADDRESS OF "OPERATOR AWAKE" MESSAGE
                        ;       TEMP1:  USED TO SETUP THE VALUE OF THE "EVENT TYPE" BYTE FOR LOGGING
                        ;
                        ; CALLING SEQUENCE:
                        ;       JSR     PC,LOGTXQ       ;CALL THE LOG EVENT SUBROUTINE WITH TEMP,TEMP1,
                        ;        ""      ""   ""        ; TEMP2, TEMP3, AND TEMP4 SETUP
                        ;        ""      ""   ""
                        ;
                        ;       JSR     PC,LOGCMP
                        ;--

017570                  LOGTXQ:
017570  012737  016231  006532        MOV     #STXQ,TEMP1     ;SET UP MSG. TO PRINT
017576  012737  000000  006530        MOV     #TXQ,TEMP       ;SET UP EVENT TYPE
017604  000517                        BR      LOGS1           ;GO LOG EVENT AND TIME

017606                  LOGTXC:
017606  012737  016242  006532        MOV     #STXC,TEMP1     ;SET UP MSG. TO PRINT
017614  012737  000002  006530        MOV     #TXC,TEMP       ;SET UP EVENT TYPE
017622  000510                        BR      LOGS1           ;GO LOG EVENT AND TIME

017624                  LOGRXQ:
017624  012737  016253  006532        MOV     #SRXQ,TEMP1     ;SET UP MSG. TO PRINT
017632  012737  000004  006530        MOV     #RXQ,TEMP       ;SET UP EVENT TYPE
017640  000501                        BR      LOGS1           ;GO LOG EVENT AND TIME

017642                  LOGRXC:
017642  012737  000006  006530        MOV     #RXC,TEMP       ;SET UP EVENT TYPE
017650  000475                        BR      LOGS1           ;GO LOG EVENT AND TIME
017652                  LGDVE:
```

```
                                                                J 6
 5789  017652  012737  016264  006532          MOV     #SDVE,TEMP1     ;SET UP MSG. TO PRINT
 5790  017660  012737  000010  006530          MOV     #DER,TEMP       ;SET UP EVENT TYPE
 5791  017666  000503                          BR      LOGS3           ;GO LOG EVENT AND TIME
 5792
 5793  017670                          LOGDVI:
 5794  017670  012737  016306  006532          MOV     #SDVI,TEMP1     ;SET UP MSG. TO PRINT
 5795  017676  012737  000012  006530          MOV     #DVI,TEMP       ;SET UP EVENT TYPE
 5796  017704  113737  006562  006534          MOVB    MODTYP,TEMP2
 5797  017712  113737  006564  006535          MOVB    MLTYP,TEMP2+1
 5798  017720  013737  006572  006536          MOV     RPASS,TEMP3
 5799  017726  013737  006570  006540          MOV     PARAM,TEMP4     ;SET UP EVNT ENTRIES
 5800  017734  000460                          BR      LOGS3           ;GO LOG EVENT AND TIME
 5801
 5802  017736                          LOGCMP:
 5803  017736  012737  016275  006532          MOV     #SCM,TEMP1      ;SET UP MSG. TO PRINT
 5804  017744  012737  000014  006530          MOV     #DCK,TEMP       ;SET UP EVENT TYPE
 5805  017752  000451                          BR      LOGS3
 5806  017754                          LOGCML:
 5807  017754  012737  016317  006532          MOV     #SCML,TEMP1
 5808  017762  012737  000020  006530          MOV     #DLE,TEMP       ;SET UP MSG. AND TYPE
 5809  017770  000442                          BR      LOGS3           ;GO LOG EVENT AND TIME
 5810  017772                          LOGCMD:
 5811  017772  012737  016330  006532          MOV     #SCMD,TEMP1
 5812  020000  012737  000022  006530          MOV     #DDE,TEMP
 5813  020006  000433                          BR      LOGS3           ;GO LOG MSG TYPE AND TIME
 5814  020010                          LOGEOP:
 5815  020010  012737  016341  006532          MOV     #SEOP,TEMP1
 5816  020016  012737  000024  006530          MOV     #EOP,TEMP
 5817  020024  000424                          BR      LOGS3           ;GO LOG MSG TYPE AND TIME
 5818
 5819
 5833  020026                          LOGMSC:
 5834  020026  012737  016352  006532          MOV     #SMSC,TEMP1
 5835  020034  012737  000016  006530          MOV     #MSC,TEMP
 5836  020042  000415                          BR      LOGS3
 5837
 5838
 5839  020044  013746  006502          LOGS1:  MOV     ERRCNT,-(SP)    ;SAVE CURRENT ERROR COUNT
 5840  020050  004737  033502                  JSR     PC,DVMODS       ; GO GET MODEM STATUS
 5841  020054  012604                          MOV     (SP)+,R4        ;GET SAVED ERRCNT VALUE
 5842  020056  020437  006502                  CMP     R4,ERRCNT       ;WHERE ANY ERRORS FOUND
 5843  020062  001402                          BEQ     1$              ; BR IF NONE
 5844  020064  000137  020300                  JMP     LOGEX           ; ELSE, LEAVE WITHOUT LOGGING ANYTHING
 5845                                                                  ;   BUT THE DEVICE ERROR FROM 'DVMODS'
 5846  020070  013737  007550  006540  1$:     MOV     MODS,TEMP4      ;AND PUT IT IN TEMP4
 5847
 5848  020076                          LOGS3:
 5849  020076  022737  000006  006530          CMP     #RXC,TEMP
 5850  020104  001434                          BEQ     LOGS5           ;IF RXC DONT PRINT
 5851  020106  032737  000001  006570          BIT     #STATB,PARAM
 5852  020114  001430                          BEQ     LOGS5           ;IF NO STATUS SELECTED
 5853                                                                  ;GO TO 5
 5854
 5855  020116  022737  000010  006474          CMP     #10,LNCNT       ;HAVE WE DONE 10?
 5856  020124  001012                          BNE     LOGS4           ;IF NOT GO TO 4
 5857  020126  005037  006474                  CLR     LNCNT           ;ESLE CLEAR IT
```

```
5858
5859  020132                                        PRINTF  #CR             ;ELSE PRINT CR
 (7)  020132  012746  016226                                                          MOV     #CR,-(SP)
 (6)  020136  012746  000001                                                          MOV     #1,-(SP)
 (3)  020142  010600                                                                  MOV     SP,R0
 (4)  020144  104417                                                                  TRAP    C$PNTF
 (4)  020146  062706  000004                                                          ADD     #4,SP
5860  020152                                  LOGS4:
5861  020152  005237  006474                          INC     LNCNT           ;INC COUNTER OF # OF AWAKE MSGS
5862  020156                                          PRINTF  TEMP1           ;PRINT OPERATOR AWAKE MSG.
 (7)  020156  013746  006532                                                          MOV     TEMP1,-(SP)
 (6)  020162  012746  000001                                                          MOV     #1,-(SP)
 (3)  020166  010600                                                                  MOV     SP,R0
 (4)  020170  104417                                                                  TRAP    C$PNTF
 (4)  020172  062706  000004                                                          ADD     #4,SP
5863  020176  010346                          LOGS5:  MOV     R3,-(SP)        ;SAVE R3 ON THE STACK
5864  020200  013703  006642                          MOV     EVTPTR,R3
5865  020204  113723  006530                          MOVB    TEMP,(R3)+      ;LOG EVENT
5866  020210  013737  006622  006530                  MOV     CLKHZ,TEMP
5867  020216  163737  006632  006530                  SUB     TIMTCK,TEMP
5868  020224  113723  006530                          MOVB    TEMP,(R3)+      ;LOG TIME SINCE START
5869  020230  113723  006630                          MOVB    TIMSEC,(R3)+
5870  020234  113723  006626                          MOVB    TIMMIN,(R3)+    ;TICKS,SECS AND MINS.
5871  020240  013723  006534                          MOV     TEMP2,(R3)+     ;LOG EVNT ENTRY 3
5872  020244  013723  006536                          MOV     TEMP3,(R3)+     ;LOG EVNT ENTRY 4
5873  020250  013723  006540                          MOV     TEMP4,(R3)+     ;LOG EVNT ENTRY 5
5874  020254  020327  007546                          CMP     R3,#EVTEND
5875  020260  103404                                  BLO     LOGS2           ;IF EVENT LOG FULL GO
5876                                                                          ;CONTINUE;ELSE GO TO 2
5877  020262  012713  177777                          MOV     #-1,(R3)        ;LOG A TABLE END
5878  020266  012703  006644                          MOV     #EVTLOG,R3      ;PUT R3 TO START OF TABLE
5879  020272  010337  006642          LOGS2:  MOV     R3,EVTPTR       ;RESTORE POINTER
5880  020276  012603                          MOV     (SP)+,R3        ;RESTORE R3
5881  020300  000207                  LOGEX:  RTS     PC
5882
```

```
                                        .SBTTL  REPORT EVENT LOG
                                                ;REV B BY EC
                                                ;THE FOLLOWING COMMANDS ADDED TO REVISION B CVCLHB
                                                ;;DPV DCLT PROGRAM
                                                ;;RPT> LOG
                                                ;;      HELP
                                                ;;      EXIT
                                                ;;AT SOME FUTURE DATE AFTER DDCMP PROTOCOL IS SUPPORTED BY THIS
                                                ;;PROGRAM, ERROR REPORTING CAN BE ADDED HERE.

020302  010246                  REPORT: MOV     R2,-(SP)        ;SAVE R2,R3,R4 ON THE STACK
020304  010346                          MOV     R3,-(SP)
020306  010446                          MOV     R4,-(SP)

                                        ;PRINT HELP MESSAGE
020310                                  PRINTF  #RHLP0          ;BASIC HELP MESSAGE
(7)  020310  012746  013034                                                     MOV     #RHLP0,-(SP)
(6)  020314  012746  000001                                                     MOV     #1,-(SP)
(3)  020320  010600                                                             MOV     SP,R0
(4)  020322  104417                                                             TRAP    C$PNTF
(4)  020324  062706  000004                                                     ADD     #4,SP

020330  105037  003143          GETRCL: CLRB    P$GDBD          ;INIT GOOD/BAD FLAG  -1=BAD INPUT
020334  105037  003142                  CLRB    P$NNUF          ;INIT MORE COMMAND LINE INPUT NEEDED

                                        ;PRINT PROMPT 'RPT>'
020340                                  GMANID  CLI$RP,CMDBUF,A,0,1,72.,NO
(3)  020340  104443                                                             TRAP    C$GMAN
(3)  020342  000406                                                             BR      10000$
(4)  020344  002666                                                             .WORD   CMDBUF
(5)  020346  000142                                                             .WORD   T$CODE
(5)  020350  011546                                                             .WORD   CLI$RP
(5)  020352  000000                                                             .WORD   0
(5)  020354  000001                                                             .WORD   T$LOLIM
(5)  020356  000110                                                             .WORD   T$HILIM
(3)  020360                                                             10000$:
020360  012737  002666  003126          MOV     #CMDBUF,P$BUFA  ;INPUT BUFFER
020366  012737  020522  003130          MOV     #CLIRT,P$TREE   ;REPORT CLI TREE
020374  012737  020610  003132          MOV     #CLIRAC,P$ACT   ;ACTION ROUTINES
020402  005037  003012                  CLR     QUALFG
020406  004737  023066                  JSR     PC,P$TRV        ;GO PARSE COMMAND LINE
020412  105737  003143                  TSTB    P$GDBD          ;COMMAND OK ?
020416  001412                          BEQ     1$              ;YES,BRANCH
020420                                  PRINTF  #CLIERM         ;PRINT INVALID INPUT MESSAGE
(7)  020420  012746  011553                                                     MOV     #CLIERM,-(SP)
(6)  020424  012746  000001                                                     MOV     #1,-(SP)
(3)  020430  010600                                                             MOV     SP,R0
(4)  020432  104417                                                             TRAP    C$PNTF
(4)  020434  062706  000004                                                     ADD     #4,SP
020440  000137  020330                  JMP     GETRCL          ;TRY AGAIN

020444  105737  003142          1$:     TSTB    P$NNUF          ;MORE COMMAND NEEDED ?
020450  001412                          BEQ     10$             ;NO,BRANCH
020452                                  PRINTF  #CLINUF         ;INCOMPLETE MESSAGE
(7)  020452  012746  011603                                                     MOV     #CLINUF,-(SP)
(6)  020456  012746  000001                                                     MOV     #1,-(SP)
```

```
 (3)  020462  010600                                                          MOV     SP,R0
 (4)  020464  104417                                                          TRAP    C$PNTF
 (4)  020466  062706  000004                                                  ADD     #4,SP
5919  020472  000137  020330                    JMP     GETRCL               ;TRY AGAIN
5920
5921  020476  023727  003010  000002  10$:     CMP     KEYWD1,#RPEXT        ;EXIT COMMAND ?
5922  020504  001402                            BEQ     20$                  ;YES,BRANCH
5923  020506  000137  020330                    JMP     GETRCL               ;GET ANOTHER COMMAND
5924  020512  012604                  20$:      MOV     (SP)+,R4             ;RESTORE R4
5925  020514  012603                            MOV     (SP)+,R3             ;RESTORE R3
5926  020516  012602                            MOV     (SP)+,R2             ;RESTORE R2
5927  020520  000207                            RTS     PC                   ;RETURN
```

```
 5929
 5930  020522                .SBTTL  COMMAND LINE PARSING TREE FOR REPORT
 5931  020526  CLIRT:  CLI CLISPA,0,R10$                ;SKIP SPACES IN COMMAND LINE
 5932  020532  R10$:   CLI <'?>,RPHLP,R11$             ;IF INPUT = ? THEN PRINT HELP MESSAGE
 5933  020534          CLI CLIEXI,0                     ;;AND EXIT PARSER
 5934  020550  R11$:   CLI CLISTR,RPHLP,R12$,<'HELP'>  ;IF INPUT = 'HELP' THEN PRINT HELP
 5935  020552          CLI CLIEXI,0                     ;;MESSAGE AND EXIT PARSER
 5936  020566  R12$:   CLI CLISTR,RPEXT,R13$,<'EXIT'>  ;IF INPUT = 'EXIT' THEN SET KEYWORD =
 5937  020570          CLI CLIEXI,0                     ;;RPEXT AND EXIT PARSER
 5938  020602  R13$:   CLI CLISTR,RPLOG,R30$,<'LOG'>   ;IF INPUT = 'LOG' THEN GO PRINT EVENT
 5939  020604          CLI CLIEXI,0                     ;;LOG AND EXIT PARSER
 5940  020606  R30$:   CLI CLIERR,0
          R125$:  CLI CLIEXI,0
```

```
                                        .SBTTL  CLI ACTION DISPATCHER AND ROUTINES
020610  006302                  CLIRAC: ASL     R2              ;SET UP INDEX
020612  016202  020626                  MOV     10$(R2),R2      ;
020616  062702  020626                  ADD     #10$,R2         ;
020622  004712                          JSR     PC,(R2)         ;GO DO ACTION
020624  000207                          RTS     PC              ;RETURN
020626  000010                  10$:    .WORD   ACTRNL-10$      ;NULL
020630  000012                          .WORD   ACTRHL-10$      ;HELP ROUTINE
020632  000056                          .WORD   ACTREX-10$      ;EXIT ROUTINE
020634  000066                          .WORD   ACTRLG-10$      ;REPORT EVENT LOG ROUTINE

                                        ;;;;;ACTION ROUTINES FOR REPORT;;;;;
020636  000207                  ACTRNL: RTS     PC              ;NULL

                                        ;PRINT HELP MESSAGE
020640  012702  003036          ACTRHL: MOV     #RHLPTB,R2      ;INDEX FOR HELP MESSAGES
020644                          1$:     PRINTF  #HLPF,(R2)+     ;PRINT IT
 (8)    020644  012246                                                  MOV     (R2)+,-(SP)
 (7)    020646  012746  012241                                          MOV     #HLPF,-(SP)
 (6)    020652  012746  000002                                          MOV     #2,-(SP)
 (3)    020656  010600                                                  MOV     SP,R0
 (4)    020660  104417                                                  TRAP    C$PNTF
 (4)    020662  062706  000006                                          ADD     #6,SP
020666  020227  003046                  CMP     R2,#RHLPEN      ;LAST MESSAGE ?
020672  001364                          BNE     1$              ;NO,BRANCH
020674  012737  000001  003010          MOV     #RPHLP,KEYWD1   ;SET KEYWORD
020702  000207                          RTS     PC              ;RETURN

                                        ;EXIT REPORT LEVEL
020704  012737  000002  003010  ACTREX: MOV     #RPEXT,KEYWD1   ;SET KEYWORD AND RETURN
020712  000207                          RTS     PC

                                        ;PRINT ERROR LOG
020714  004737  020730          ACTRLG: JSR     PC,REPLOG       ;GO PRINT EVENT LOG
020720  012737  000003  003010          MOV     #RPLOG,KEYWD1   ;SET KEYWORD
020726  000207                          RTS     PC              ;RETURN
```

```
 5976                                   .SBTTL          DUMP EVENT LOG
 5977
 5978
 5979  020730  010246                   REPLOG: MOV     R2,-(SP)        ;SAVE R2,R3,R4 ON THE STACK
 5980  020732  010346                           MOV     R3,-(SP)
 5981  020734  010446                           MOV     R4,-(SP)
 5982
 5990
 5991
 5992
 5993  020736  013702  006642                   MOV     EVTPTR,R2       ;MAKE R2 A POINTER TO EVENT TABLE
 5994  020742  023727  006644  177777           CMP     EVTLOG,#-1      ;SEE IF EVENT TABLE IS EMPTY
 5995  020750  001034                           BNE     RPT0            ;BR IF NO
 5996  020752                                   PRINTS  #NULEVT         ;IF EMPTY TELL OPRERATOR.
  (7)  020752  012746  015201                                                           MOV     #NULEVT,-(SP)
  (6)  020756  012746  000001                                                           MOV     #1,-(SP)
  (3)  020762  010600                                                                   MOV     SP,R0
  (4)  020764  104416                                                                   TRAP    C$PNTS
  (4)  020766  062706  000004                                                           ADD     #4,SP
 5997  020772  000137  021656                   JMP     ENDEVT          ;AND END
 5998
 5999  020776  162702  000012           RPT:    SUB     #12,R2          ;NOW POINT BACK TO TOP OF ENTRY U
 6000                                                                   ;JUST PRINTED
 6001
 6002  021002  020227  006644                   CMP     R2,#EVTLOG      ;POINTING TO TOP OF EVNT LOG QUEUE?
 6003  021006  001010                           BNE     RPT1            ; BR IF NO
 6004  021010  012702  007546                   MOV     #EVTEND,R2      ;SET R2 TO POINT TO BOTTOM OF LOG
 6005  021014  026227  177776  177777           CMP     -2(R2),#-1
 6006  021022  001007                           BNE     RPT0            ;IF END OF LOG IS NOT EMPTY
 6007  021024  000137  021656                   JMP     ENDEVT          ;CONTINUE...ELSE EXIT
 6008
 6009  021030  020237  006642           RPT1:   CMP     R2,EVTPTR       ;ARE WE BACK TO POINTER?
 6010  021034  001002                           BNE     RPT0            ;IF NOT CONTINUE
 6011  021036  000137  021656                   JMP     ENDEVT          ;IF SO EXIT....
 6012
 6013  021042  162702  000012           RPT0:   SUB     #12,R2          ;POINT R2 TO START OF ENTRY
 6014  021046                           RPTAA:  PRINTS  #EVTF0          ;PRINT EVENT ENTRY HEADER
  (7)  021046  012746  015241                                                           MOV     #EVTF0,-(SP)
  (6)  021052  012746  000001                                                           MOV     #1,-(SP)
  (3)  021056  010600                                                                   MOV     SP,R0
  (4)  021060  104416                                                                   TRAP    C$PNTS
  (4)  021062  062706  000004                                                           ADD     #4,SP
 6015  021066  112203                           MOVB    (R2)+,R3        ;PUT EVENT TYPE INTO R3
 6016  021070  112237  007640                   MOVB    (R2)+,EVTTCK
 6017  021074  112237  007634                   MOVB    (R2)+,EVTSEC    ;PUT EVENT TIME (TICKS,SECS,MINS IN TEMP LOC.S)
 6018  021100  112237  007636                   MOVB    (R2)+,EVTMIN
 6019  021104                                   PRINTS  #EVTF1,EVTMIN,EVTSEC,EVTTCK,EVTLST(R3)  ;PRINT EVENT TIME AND DESCRIPT.
 (11)  021104  016346  007606                                                           MOV     EVTLST(R3),-(SP)
 (10)  021110  013746  007640                                                           MOV     EVTTCK,-(SP)
  (9)  021114  013746  007634                                                           MOV     EVTSEC,-(SP)
  (8)  021120  013746  007636                                                           MOV     EVTMIN,-(SP)
  (7)  021124  012746  015335                                                           MOV     #EVTF1,-(SP)
  (6)  021130  012746  000005                                                           MOV     #5,-(SP)
  (3)  021134  010600                                                                   MOV     SP,R0
  (4)  021136  104416                                                                   TRAP    C$PNTS
  (4)  021140  062706  000014                                                           ADD     #14,SP
```

```
 6020  021144  000173  007650                JMP     @RPTDSP(R3)     ;DISPATCH TO DECODING SECTION FOR SPECIFIC TYPE
 6021
 6022  021150  012237  007642  RPTTXQ:       MOV     (R2)+,EVTADD    ;STORE MESSAGE ADDRESS FOR PRINTING
 6023  021154  012237  007644                MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT FOR PRINTING
 6024  021160  012203                        MOV     (R2)+,R3        ;STORE MODEM STATUS FOR PRINTING
 6025  021162                                PRINTS  #EVTF2,EVTADD,EVTBCT    ;PRINT ADDR,BYTE CNT
   (9) 021162  013746  007644                                                      MOV     EVTBCT,-(SP)
   (8) 021166  013746  007642                                                      MOV     EVTADD,-(SP)
   (7) 021172  012746  015364                                                      MOV     #EVTF2,-(SP)
   (6) 021176  012746  000003                                                      MOV     #3,-(SP)
   (3) 021202  010600                                                              MOV     SP,R0
   (4) 021204  104416                                                              TRAP    C$PNTS
   (4) 021206  062706  000010                                                      ADD     #10,SP
 6026  021212  004737  021666                JSR     PC,RPTMSB       ;GO PRINT MODEM STATUS
 6027  021216  000137  020776                JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
 6028
 6029  021222  012237  007646  RPTDER:       MOV     (R2)+,EVTTMP    ;GET ADDRESS OF DEVICE INFO MESSAGE
 6030  021226  012237  007676                MOV     (R2)+,DEV1      ;STORE DEVICE REG CONTENTS FOR PRINTING
 6031  021232  012237  007700                MOV     (R2)+,DEV2
 6032  021236                                PRINTS  #EVTF3,EVTTMP   ;PRINT DEVICE REG CONTENTS.
   (8) 021236  013746  007646                                                      MOV     EVTTMP,-(SP)
   (7) 021242  012746  015436                                                      MOV     #EVTF3,-(SP)
   (6) 021246  012746  000002                                                      MOV     #2,-(SP)
   (3) 021252  010600                                                              MOV     SP,R0
   (4) 021254  104416                                                              TRAP    C$PNTS
   (4) 021256  062706  000006                                                      ADD     #6,SP
 6033  021262                                PRINTS  #EVTF3C,DEV1,DEV2
   (9) 021262  013746  007700                                                      MOV     DEV2,-(SP)
   (8) 021266  013746  007676                                                      MOV     DEV1,-(SP)
   (7) 021272  012746  015450                                                      MOV     #EVTF3C,-(SP)
   (6) 021276  012746  000003                                                      MOV     #3,-(SP)
   (3) 021302  010600                                                              MOV     SP,R0
   (4) 021304  104416                                                              TRAP    C$PNTS
   (4) 021306  062706  000010                                                      ADD     #10,SP
 6034  021312  000137  020776                JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
 6035
 6036  021316  005037  007676  RPTDVI:       CLR     DEV1
 6037  021322  005037  007700                CLR     DEV2            ;CLEAR UPPER BYTES OF DEV1 & DEV2 BEFORE USE
 6038  021326  112237  007676                MOVB    (R2)+,DEV1      ;STORE SETUP OPERATION PARAMETERS FOR PRINTING
 6039  021332  112237  007700                MOVB    (R2)+,DEV2
 6040  021336  012237  007702                MOV     (R2)+,DEV3
 6041  021342  012237  007704                MOV     (R2)+,DEV4
 6042  021346  010246                        MOV     R2,-(SP)        ;SAVE R2 ON THE STACK
 6043  021350  004737  022564                JSR     PC,SHWOP        ;GO PRINT MODE, MAINT-LOOP TYPE, PARAMTERS.
 6044  021354  012602                        MOV     (SP)+,R2        ;RESTORE R2
 6045  021356  000137  020776                JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
 6046  021362  012237  007642  RPTEOP:       MOV     (R2)+,EVTADD
 6047  021366  012237  007644                MOV     (R2)+,EVTBCT
 6048  021372  012237  007646                MOV     (R2)+,EVTTMP
 6049  021376                                PRINTS  #EVTF4B,EVTADD,EVTBCT,EVTTMP    ;PRINT ADDR,RXBYTES,CMPBYTES.
  (10) 021376  013746  007646                                                      MOV     EVTTMP,-(SP)
   (9) 021402  013746  007644                                                      MOV     EVTBCT,-(SP)
   (8) 021406  013746  007642                                                      MOV     EVTADD,-(SP)
   (7) 021412  012746  015707                                                      MOV     #EVTF4B,-(SP)
   (6) 021416  012746  000004                                                      MOV     #4,-(SP)
   (3) 021422  010600                                                              MOV     SP,R0
```

```
  (4)  021424  104416                                                                    TRAP    C$PNTS
  (4)  021426  062706  000012                                                            ADD     #12,SP
6050
6051   021432  000137  020776          JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY
6052
6053
6054   021436  012237  007642  RPTDDE: MOV     (R2)+,EVTADD    ;STORE MESSAGE ADDRESS FOR PRINTING
6055   021442  012237  007644          MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT FOR PRINTING
6056   021446  012237  007646          MOV     (R2)+,EVTTMP    ;STORE TOTAL # OF CMP ERRORS
6057   021452                         PRINTS  #EVTF4,EVTADD,EVTBCT,EVTTMP     ;PRINT ADDR, BYTE CNT, # CMP ERRS
 (10)  021452  013746  007646                                                            MOV     EVTTMP,-(SP)
 (9)   021456  013746  007644                                                            MOV     EVTBCT,-(SP)
 (8)   021462  013746  007642                                                            MOV     EVTADD,-(SP)
 (7)   021466  012746  015507                                                            MOV     #EVTF4,-(SP)
 (6)   021472  012746  000004                                                            MOV     #4,-(SP)
 (3)   021476  010600                                                                    MOV     SP,R0
 (4)   021500  104416                                                                    TRAP    C$PNTS
 (4)   021502  062706  000012                                                            ADD     #12,SP
6058   021506  000137  020776          JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY
6059
6060   021512                 RPTDLE:
6061   021512  012237  007642  RPTDCK: MOV     (R2)+,EVTADD    ;STORE MSG ADDR FOR PRINT
6062   021516  012237  007644          MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT
6063   021522  012237  007646          MOV     (R2)+,EVTTMP    ;STORE BYTE COUNT COMP
6064   021526                         PRINTS  #EVTF4A,EVTADD,EVTBCT,EVTTMP    ;PRINT ADDR,RXBYTES,CMPBYTES.
 (10)  021526  013746  007646                                                            MOV     EVTTMP,-(SP)
 (9)   021532  013746  007644                                                            MOV     EVTBCT,-(SP)
 (8)   021536  013746  007642                                                            MOV     EVTADD,-(SP)
 (7)   021542  012746  015611                                                            MOV     #EVTF4A,-(SP)
 (6)   021546  012746  000004                                                            MOV     #4,-(SP)
 (3)   021552  010600                                                                    MOV     SP,R0
 (4)   021554  104416                                                                    TRAP    C$PNTS
 (4)   021556  062706  000012                                                            ADD     #12,SP
6065
6066   021562  000137  020776          JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY
6067
6068
6069
6080
6081
6082   021566  012237  007646  RPTMSC: MOV     (R2)+,EVTTMP
6083   021572                         PRINTS  #EVTF3,EVTTMP   ;PRINT CHANGE TYPE
 (8)   021572  013746  007646                                                            MOV     EVTTMP,-(SP)
 (7)   021576  012746  015436                                                            MOV     #EVTF3,-(SP)
 (6)   021602  012746  000002                                                            MOV     #2,-(SP)
 (3)   021606  010600                                                                    MOV     SP,R0
 (4)   021610  104416                                                                    TRAP    C$PNTS
 (4)   021612  062706  000006                                                            ADD     #6,SP
6084   021616  012203          MOV     (R2)+,R3        ;PUT OLD MODEM STATUS IN R3 FOR PRINTING
6085   021620  004737  021666  JSR     PC,RPTMSB       ;GO PRINT OLD MODEM STATUS
6086   021624                 PRINTS  #EVMOCG         ;GO PRINT ''CHANGED TO:''
 (7)   021624  012746  016051                                                            MOV     #EVMOCG,-(SP)
 (6)   021630  012746  000001                                                            MOV     #1,-(SP)
 (3)   021634  010600                                                                    MOV     SP,R0
 (4)   021636  104416                                                                    TRAP    C$PNTS
 (4)   021640  062706  000004                                                            ADD     #4,SP
```

```
                        DUMP EVENT LOG

 6087  021644  012203                MOV     (R2)+,R3        ;PUT NEW MODEM STATUS IN R3 FOR PRINTING
 6088  021646  004737  021666        JSR     PC,RPTMSB       ;GO PRINT NEW MODEM STATUS
 6089  021652  000137  020776  RPTMSE: JMP     RPT             ;THEN GO GET NEXT EVENT
 6090
 6091
 6092  021656  012604        ENDEVT: MOV     (SP)+,R4        ;RESTORE R4,R3,R2
 6093  021660  012603                MOV     (SP)+,R3
 6094  021662  012602                MOV     (SP)+,R2
 6095  021664  000207                RTS     PC              ;RETURN TO CALLING ROUTINE
 6096
 6097
 6098                          ;REPORT MODEM STATUS SUBROUTINE
 6099                          ;       PART OF STATISICAL REPORTING (DUMPING EVENT LOG)
 6100
 6101  021666        RPTMSB: PRINTS  #EVMOHD                 ;PRINT MODEM STATUS HEADER
   (7)  021666  012746  016074                                       MOV     #EVMOHD,-(SP)
   (6)  021672  012746  000001                                       MOV     #1,-(SP)
   (3)  021676  010600                                               MOV     SP,R0
   (4)  021700  104416                                               TRAP    C$PNTS
   (4)  021702  062706  000004                                       ADD     #4,SP
 6102  021706  012704  007552                MOV     #MOBITS,R4      ;MAKE R4 A POINTER TO MODEM SIG. BIT DEF. TABLE
 6103  021712  012705  007570                MOV     #MOMSGS,R5      ;MAKE R5 A POINTER TO MODEM MSG. POSITION TABLE
 6104  021716  005714        6$:     TST     (R4)            ;SEE IF BIT AVAIABLE FROM DEVICE
 6105  021720  001004                BNE     7$              ;BR IF THAT MODEM SIG. AVAIABLE
 6106  021722  112735  000130        MOVB    #'X,@(R5)+      ;ELSE PUT 'X' IN REPORT IF SIGNAL NOT AVAILABLE
 6107  021726  005724                TST     (R4)+           ;BUMP R4 TO POINT TO NEXT BIT DEFINITION
 6108  021730  000407                BR      9$              ;GO SEE IF CHECKED ALL MODEM SIGNALS
 6109  021732  032403        7$:     BIT     (R4)+,R3        ;IF THERE, SEE IF THAT BIT IN DEVICE'S ENTRY=1
 6110  021734  001403                BEQ     8$              ;BR IF BIT (SIGNAL) VALUE =0
 6111  021736  112735  000061        MOVB    #'1,@(R5)+      ;IF=1, PUT '1' IN REPORT MESSAGE
 6112  021742  000402                BR      9$              ;GO SEE IF ALL MODEM SIGNALS CHECKED
 6113  021744  112735  000060  8$:   MOVB    #'0,@(R5)+      ;IF BIT(SIGNAL)=0, PUT '0' IN REPORT MESSAGE
 6114  021750  020427  007570  9$:   CMP     R4,#MOBITE      ;SEE IF ALL BITS(SIGNALS) CHECKED
 6115  021754  002760                BLT     6$              ;LOOP UNTIL ALL SIGNALS(BITS) CHECKED
 6116  021756                        PRINTS  #EVMOST         ;THEN PRINT MODEM SIGNAL VALUE MESSAGE
   (7)  021756  012746  016154                                       MOV     #EVMOST,-(SP)
   (6)  021762  012746  000001                                       MOV     #1,-(SP)
   (3)  021766  010600                                               MOV     SP,R0
   (4)  021770  104416                                               TRAP    C$PNTS
   (4)  021772  062706  000004                                       ADD     #4,SP
 6117  021776  000207                RTS     PC              ;RETURN TO EVENT DECODING
 6118
 6119
```

```
                        .SBTTL          DUMP BYTES OR WORDS


                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       DUMPSR - DUMP BYTES OR WORDS SUBROUTINE
                        ;
                        ;       THIS SUBROUTINE PRINTS THE CONTENTS OF THE LOCATIONS BETWEEN
                        ;       A STARTING AND END ADDRESS IN LOCS. "STADD" AND "ENADD".
                        ;       THE WORD OR BYTE CONTENTS ARE PRINTED 8 TO A LINE WITH THE
                        ;       ADDRESS OF THE FIRST BYTE AS THE FIRST 6 OCTAL CHARS. FOLLOWED
                        ;       BY A SEMICOLON.
                        ;
                        ; INPUTS:
                        ;       STADD=  STARTING ADDRESS (FIRST LOC. TO PRINT)
                        ;       ENADD=  END ADDRESS (LAST LOCATION TO DUMP)
                        ;       BYTBIT= 1 IF SUPPOSED TO PRINT "BYTES"
                        ;               0 IF SUPPOSED TO PRINT "WORDS"
                        ;
                        ; OUTPUTS:
                        ;       CONTENTS OF A RANGE OF LOC.S PRINTED ON THE OPERATORS CONSOLE.
                        ;
                        ; CALLING SEQUENCE:
                        ;       JSR PC,DUMPSR           ;CALL DUMP BYTES SUBROUTINE
                        ;
                        ;--

022000  013702  006504  DUMPSR: MOV     STADD,R2        ;SET R2 UP TO STARTING ADDR.
022004  005003          DUM4:   CLR     R3              ;CLEAR R3
022006                  PRINTF  #BASM1,R2       ;PRINT ADDRESS
  (8)   022006  010246                                          MOV     R2,-(SP)
  (7)   022010  012746  015173                                  MOV     #BASM1,-(SP)
  (6)   022014  012746  000002                                  MOV     #2,-(SP)
  (3)   022020  010600                                          MOV     SP,R0
  (4)   022022  104417                                          TRAP    C$PNTF
  (4)   022024  062706  000006                                  ADD     #6,SP
022030  005737  006510  DUM3:   TST     BYTBIT          ;IS THIS BYTE OR WORD
022034  001416          BEQ     DUM1            ;BR IF WORD
022036  112237  006530  MOVB    (R2)+,TEMP      ;MOV BYTE TO TEMP
022042                  PRINTF  #BASM3,<B,TEMP> ;PRINT BYTE
  (8)   022042  005046                                          CLR     -(SP)
  (8)   022044  153716  006530                                  BISB    TEMP,(SP)
  (7)   022050  012746  015155                                  MOV     #BASM3,-(SP)
  (6)   022054  012746  000002                                  MOV     #2,-(SP)
  (3)   022060  010600                                          MOV     SP,R0
  (4)   022062  104417                                          TRAP    C$PNTF
  (4)   022064  062706  000006                                  ADD     #6,SP
022070  000411                  BR      DUM2
022072                  DUM1:   PRINTF  #BASM2,(R2)+    ;PRINT WORD
  (8)   022072  012246                                          MOV     (R2)+,-(SP)
  (7)   022074  012746  015164                                  MOV     #BASM2,-(SP)
  (6)   022100  012746  000002                                  MOV     #2,-(SP)
  (3)   022104  010600                                          MOV     SP,R0
  (4)   022106  104417                                          TRAP    C$PNTF
  (4)   022110  062706  000006                                  ADD     #6,SP
022114  020237  006506  DUM2:   CMP     R2,ENADD        ;COMPARE FOR LAST ADD
```

```
 6158  022120  003005                    BGT     DUMEX           ;IF DONE EXIT
 6159  022122  005203                    INC     R3              ;ELSE BUMP R3
 6160  022124  022703  000010            CMP     #8.,R3          ;HAVE WE PRINTED 8 ACCROSS
 6161  022130  001725                    BEQ     DUM4            ;IF SO GO BACK TO 4
 6162  022132  000736                    BR      DUM3            ;ELSE GO BACK AND PRINT ANOTHER
 6163                                                            ;BYTE OR WORD
 6164  022134  000207            DUMEX:  RTS     PC              ;RETURN TO CALLER
 6165
```

```
                        .SBTTL          UPDATE TOTAL CHAR. COUNT SUBROUTINE

                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;               UPDATES TOTAL CHAR. COUNT TOTCC BASED ON CURCC.
                        ;               LAST MESSAGE IS TRUNCATED TO FIT INTO THE
                        ;               BUFFER IF TOTAL CHAR. COUNT EXCEEDS 'BUFLIM'' A MESSAGE
                        ;               IS PRINTED TELLING THE OPERATOR THE TRUNCATION OCCURED.
                        ;
                        ; INPUTS:
                        ;               CURCC=  CHAR. COUNT OF MESSAGE BEING ADDED
                        ;               TOTCC=  TOTAL CHAR COUNT OF BUFFER ITS BEING ADDED TO
                        ;
                        ; OUTPUTS:
                        ;               MESSAGE TO OPERATOR IF MESSAGE TRUNCATED TO FIT
                        ;
                        ; FUNCTIONAL SIDE EFFECTS:
                        ;               LOCATION ''TEMP'' USED FOR CALCULATIONS
                        ; CALLING SEQUENCE:
                        ;               JSR     PC,ADCC         ;UPDATED TOTAL CHAR. COUNT
                        ;--

022136  063737  006514  006524  ADDCC:  ADD     CURCC,TOTCC     ;ADD CURRENT TO TOTAL
022144  022737  001000  006524          CMP     #BUFLIM,TOTCC   ; COMPARE TO 'BUFLIM''
022152  103027                          BHIS    ADDC1           ;IF NOT MORE THEN 'BUFLIM'' EXIT

                        ; PRINT MESSAGE AND TRUNCATE COUNT

022154                  PRINTF  #MSGTRU
(7) 022154  012746  014126                                              MOV     #MSGTRU,-(SP)
(6) 022160  012746  000001                                              MOV     #1,-(SP)
(3) 022164  010600                                                      MOV     SP,R0
(4) 022166  104417                                                      TRAP    C$PNTF
(4) 022170  062706  000004                                              ADD     #4,SP
022174  163737  006514  006524          SUB     CURCC,TOTCC     ;SUB CURRENT FROM TOTAL
022202  012737  001000  006530          MOV     #BUFLIM,TEMP    ;MOV 'BUFLIM'' TO TEMP
022210  163737  006524  006530          SUB     TOTCC,TEMP      ;SUB TOTAL FROM 'BUFLIM''
022216  013737  006530  006514          MOV     TEMP,CURCC      ;AND ESTABLISH NEW CURRENT
022224  063737  006514  006524          ADD     CURCC,TOTCC     ;ADD ''ADJUSTED CURRENT'' TO TOTAL CHAR. CNT.
022232  000207                  ADDC1:  RTS     PC              ;RETURN TO CALLER
```

```
                        .SBTTL          BUILD MESSAGE BUFFERS SUBROUTINE

                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       BLDBUF--  BUILD POINTER TABLE AND BUFFERS
                        ;
                        ;       THIS SUBROUTINE ADDS A MESSAGE TO THE TRANSMIT OR EXPECT LIST
                        ;       USING THE POINTER, BYTE COUNT, AND ADDRESS PASSED TO IT.
                        ;
                        ; INPUTS:
                        ;       CURCC=  CHAR. COUNT OF MESSAGE TO BE ADDED
                        ;       CURADD= ADDRESS OF MESSAGE TO BE ADDED
                        ;       CPTR=   ADDRESS OF POINTER TABLE WORD WHERE MESSAGE POINTERS ARE
                        ;                       TO BE BUILT
                        ;       MSGTYP= VALUE TO USE AS AN INDEX TO FIND SOURCE OF MESSAGE DATA
                        ;                       INDEX INTO DMSGCT() AND DMSGAD().
                        ; OUTPUTS:
                        ;       A MESSAGE ADDED TO EITHER TXBUF OR CMPBUF
                        ;       APPROPRIATE POINTERS IN PTRTAB POINTER TABLE
                        ;
                        ; CALLING SEQUENCE:
                        ;       JSR PC,BLDBUF           ;BUILD MESSAGE IN BUFFER AND ADD PTRS.
                        ;--

022234                  BLDBUF:
022234  010246                  MOV     R2,-(SP)        ;SAVE R2 AND R3 ON THE STACK
022236  010346                  MOV     R3,-(SP)
022240  013702  006520          MOV     CPTR,R2

022244  013722  006522  BLDB1:  MOV     CURADD,(R2)+    ;PUT CURRENT ADD ON POINTER TAB
022250  013722  006514          MOV     CURCC,(R2)+     ;PUT CURRENT CC ON POINTER TAB
022254  010237  006520          MOV     R2,CPTR         ;PUT UPDATED R2 BACK TO CURRENT POINT
022260  013702  006512          MOV     MSGTYP,R2       ;GET MESSAGE TYPE TO USE AS INDEX
022264  006302                  ASL     R2              ;DOUBLE FOR WORD INDEX
022266  013737  006522  006530  MOV     CURADD,TEMP     ;MOVE CURRENT ADD TO TEMP
022274  063737  006514  006530  ADD     CURCC,TEMP      ;ADD CHAR COUNT TO IT TO GET END
022302  013703  006522          MOV     CURADD,R3       ;SET R3 TO CURRENT START ADD
022306  016237  002150  006534  BLDB2:  MOV     DMSGCT(R2),TEMP2        ;GET BYTE COUNT
022314  016204  002172          MOV     DMSGAD(R2),R4   ;PUT STARTING FROM ADD IN R4
022320  060437  006534          ADD     R4,TEMP2        ;ADD IT TO TEMP2 TO GET END OF FROM
022324  112423          BLDB3:  MOVB    (R4)+,(R3)+     ;MOV BYTE FROM PATTERN TO BUFFER
022326  020337  006530          CMP     R3,TEMP         ;ALL DONE?
022332  001404                  BEQ     BLDBEX          ;IF SO EXIT
022334  020437  006534          CMP     R4,TEMP2        ;IS PATTERN COUNT EXPIRED
022340  001762                  BEQ     BLDB2           ;IF SO GO START AGAIN
022342  000770                  BR      BLDB3           ;IF NOT GET ANOTHER BYTE
022344  063737  006514  006522  BLDBEX: ADD     CURCC,CURADD    ;BUMP CURADD
022352  012603                  MOV     (SP)+,R3        ;RESTORE R3 AND R2
022354  012602                  MOV     (SP)+,R2
022356  000207                  RTS     PC              ;RETURN TO CALLER
```

```
6257                    .SBTTL  CREATE FACSIMILE OF TX BUFFER AND MESSAGE LIST
6258
6259                    ;++
6260                    ;       THIS ROUTINE ADDED FOR REV B BY EC
6261                    ;FUNCTIONAL DESCRIPTION:
6262                    ;       FACSIMILE: THIS ROUTINE IS USED TO CREATE A FACSIMILE OF THE
6263                    ;                  OF THE TRANSMIT LIST AND TRANSMIT BUFFER IN THE
6264                    ;                  EXPECTED LIST AND EXPECTED BUFFER. THE ROUTINE IS
6265                    ;                  NORMALLY CALLED WHEN USER COMMAND 'SET E [XPECT]=
6266                    ;                  T [RANSMIT] IS ENTERED.
6267                    ;
6268                    ;       CALLING SEQUENCE: JSR   PC,FACSIMILE
6269                    ;
6270                    ;--
6271                    ;       DEFINITIONS  CMPBUF = EXPECTED DATA BUFFER  HOLDS MAX 512 BYTES
6272                    ;                    TXBUF  = TRANSMIT DATA BUFFER  HOLDS MAX 512 BYTES
6273                    ;                    TTOTCC = NUMBER OF BYTES IN TXBUF
6274                    ;                    PTRTAB = TOP OF MESSAGE LIST POINTER TABLE
6275                    ;                    CTOTCC = NUMBER OF BYTES IN EXPECT MESSAGE
6276                    ;                    CMPTOT = NUMBER OF EXPECTED MESSAGES
6277                    ;                    CMPPTR = EXPECTED MESSAGE LIST POINTER
6278                    ;                    TXPTR  = TRANSMIT MESSAGE LIST POINTER
6279                    ;                    TXMTOT = NUMBER OF TRANSMIT MESSAGES
6280                    ;                    CCURAD = STORAGE ADDRESS OF MESSAGE IN CMPBUF
6281                    ;                    MSGLIN = MAXIMUM NUMBER OF MESSAGES THAT CAN BE STORED
6282                    ;
6283                    ;       BEGIN FACSIMILE ROUTINE
6284                    ;       (*COPY TXBUF ==> CMPBUF*)
6285                    ;       ..SAVE R1
6286                    ;       ..INIT R1
6287                    ;       ..REPEAT
6288                    ;       ....[CMPBUF]R1=[TXBUF]R1
6289                    ;       ....R1=R1+1
6290                    ;       ..UNTIL R1 = BUFLIM
6291                    ;
6292                    ;       (*NOW CALCULATE EXPECT LIST MESSAGE POINTER*)
6293                    ;       ..CMPPTR = PTRTAB + (2 * MSGLIM)
6294                    ;
6295                    ;       (*NOW PRIME THE WHILE - DO LOOP*)
6296                    ;       ..TXPTR = PTRTAB
6297                    ;       ..CCURAD = CMPBUF
6298                    ;       ..TXPTR = TXPTR + 2
6299                    ;       ..CTOTCC = [TXPTR]
6300                    ;       ..CMPTOT = 0
6301                    ;       ..WHILE TXMTOT <> CMPTOT DO
6302                    ;       ....[CMPPTR] = CCURAD
6303                    ;       ....CMPPTR = CMPPRT + 2
6304                    ;       ....[CMPPTR] = CTOTCC
6305                    ;       ....TXPTR = TXPTR + 4
6306                    ;       ....CCURAD = CCURAD + CTOTCC
6307                    ;       ....CTOTCC = [TXPTR]
6308                    ;       ....CMPPTR = CMPPTR + 2
6309                    ;       ....CMPTOT = CMPTOT + 1
6310                    ;       ..END WHILE DO
6311                    ;       ..CTOTCC = TTOTCC
6312                    ;       END FACSIMILE ROUTINE
```

```
6313
6314  022360                                  FACSIMILE:
6315
6316  022360  010146                                  MOV     R1,-(SP)                ;SAVE R1
6317  022362  005001                                  CLR     R1                      ;INIT R1
6318  022364  116161  003144  005144  10$:           MOVB    TXBUF(R1),CMPBUF(R1)    ;COPY TX BUFFER TO EXPECTED BUFFER
6319  022372  005201                                  INC     R1                      ;BUMP INDEX
6320  022374  020127  001000                        CMP     R1,#BUFLIM              ;ALL DATA COPIED ?
6321  022400  001371                                  BNE     10$                     ;NO,BRANCH
6322
6323  022402  012701  000017            20$:        MOV     #MSGLIM,R1              ;MESSAGE LIMIT
6324  022406  006301                                  ASL     R1                      ;MULTIPLY BY 2
6325  022410  006301                                  ASL     R1                      ;MULTIPLY BY 2
6326  022412  012737  006144  006440                MOV     #PTRTAB,CMPPTR          ;TOP OF POINTER TABLE
6327  022420  060137  006440                        ADD     R1,CMPPTR               ;START OF EXPECTED POINTER TABLE
6328  022424  005001                                  CLR     R1                      ;INIT R1
6329
6330                                          ;SET UP WHILE - DO LOOP
6331  022426  012737  006144  006436                MOV     #PTRTAB,TXPTR           ;TX POINTER NOW AT TOP OF TABLE
6332  022434  012737  005144  006446                MOV     #CMPBUF,CCURAD          ;TRANSFER ADDRESS OF 1ST MESSAGE
6333  022442  062737  000002  006436                ADD     #2,TXPTR                ;BUMP POINTER
6334  022450  017737  163762  006444                MOV     @TXPTR,CTOTCC           ;BYTE COUNTER 1ST MESSAGE
6335  022456  005037  006442                        CLR     CMPTOT                  ;INIT EXPECTED MESSAGE COUNT
6336
6337                                          ;WHILE TX MESSAGE TOTAL <> EXPECTED MESSAGE TOTAL DO
6338  022462  023737  006456  006442  30$:         CMP     TXMTOT,CMPTOT           ;ALL MESSAGES COPIED ?
6339  022470  001430                                  BEQ     40$                     ;YES,BRANCH
6340  022472  013777  006446  163740                MOV     CCURAD,@CMPPTR          ;TRANSFER ADDRESS OF MESSAGE
6341  022500  062737  000002  006440                ADD     #2,CMPPTR               ;BUMP POINTER
6342  022506  013777  006444  163724                MOV     CTOTCC,@CMPPTR          ;BYTE COUNT OF MESSAGE
6343  022514  062737  000004  006436                ADD     #4,TXPTR                ;BUMP TX MESSAGE POINTER
6344  022522  063737  006444  006446                ADD     CTOTCC,CCURAD           ;CALC. TRANSFER ADDRESS
6345  022530  017737  163702  006444                MOV     @TXPTR,CTOTCC           ;BYTE COUNT NEXT MESSAGE
6346  022536  062737  000002  006440                ADD     #2,CMPPTR               ;BUMP POINTER
6347  022544  005237  006442                        INC     CMPTOT                  ;INCREMENT MESSAGE COUNT
6348  022550  000744                                  BR      30$                     ;DO IT AGAIN
6349                                          ;END WHILE - DO
6350  022552  013737  006460  006444  40$:         MOV     TTOTCC,CTOTCC           ;COPY TOTAL CHARACTER COUNT
6351
6352                                          ;END ROUTINE
6353  022560  012601                          MOV     (SP)+,R1                ;RESTORE R1
6354  022562  000207                          RTS     PC                      ;RETURN
6355
6356
```

```
                                .SBTTL          SHOW MODE OF OPERATION, LOOP TYPE AND QUALIFIERS

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       SHWOP - SHOW MODE OF OPERATION, LOOP, QULAIFIERS
                                ;               PRINTED ON THE OPERATOR'S CONSOLE.
                                ;
                                ; INPUTS:
                                ;       DEV1=   MODE TYPE (MODTYP)
                                ;       DEV2=   MAINT LOOP TYPE (MLTYP)
                                ;       DEV3=   'RUN PASS'' COUNT (RPASS) - COUNT DOWN
                                ;       DEV4=   PARAMTERS WORD (PARAM)
                                ;
                                ; IMPLICIT INPUTS:
                                ;       MODES= TABLE OF ADDRESSES OF MODE NAME STRINGS
                                ;       LOOPS= TABLE OF ADDRESSES OF LOOP TYPE NAMES
                                ;
                                ; CALLING SEQUENCE:
                                ;       JSR PC,SHWOP
                                ;--

022564  013702  007676        SHWOP:  MOV     DEV1,R2         ;GET THE MODE TYPE IN R2
022570  006302                        ASL     R2              ;MAKE IT A WORD TABLE OFFSET
022572  016237  003076  006530        MOV     MODES(R2),TEMP  ;GET ADDRESS OF MODE-IN-ASCII
022600  013702  007700                MOV     DEV2,R2         ;GET MAINTENANCE LOOP TYPE
022604  006302                        ASL     R2
022606  012737  013440  006536        MOV     #LP00,TEMP3     ;LOAD TEMP3 TO POINT TO ''/LOOP=''
022614  005702                        TST     R2              ;SEE IF /LOOP=XXXXX OR NONE
022616  001003                        BNE     10$             ; BR IF /LOOP= OF SOME KIND
022620  012737  013437  006536        MOV     #LP0,TEMP3      ;IF NO LOOP THEN DON'T PRINT ''/LOOP=''
022626  016237  003114  006532  10$:   MOV     LOOPS(R2),TEMP1 ;GET ADDRESS OF LOOP-IN-ASCII
022634  013737  007702  006534        MOV     DEV3,TEMP2      ;GET NUMBER OF PASSES
022642                                PRINTS  #SHF0,TEMP,TEMP3,TEMP1,TEMP2
  (11)  022642  013746  006534                                                  MOV     TEMP2,-(SP)
  (10)  022646  013746  006532                                                  MOV     TEMP1,-(SP)
  (9)   022652  013746  006536                                                  MOV     TEMP3,-(SP)
  (8)   022656  013746  006530                                                  MOV     TEMP,-(SP)
  (7)   022662  012746  014211                                                  MOV     #SHF0,-(SP)
  (6)   022666  012746  000005                                                  MOV     #5,-(SP)
  (3)   022672  010600                                                          MOV     SP,R0
  (4)   022674  104416                                                          TRAP    C$PNTS
  (4)   022676  062706  000014                                                  ADD     #14,SP

022702  005002                        CLR     R2              ;NOW SET UP FOR QUALIFIERS IN ASCII
022704  012737  013517  006530        MOV     #PST,TEMP
022712  032737  000001  007704        BIT     #STATB,DEV4     ;SEE IF /STATUS OR /NOSTATUS
022720  001003                        BNE     1$              ;BR IF /STATUS
022722  012737  013515  006530        MOV     #PNST,TEMP
022730  012737  013530  006532  1$:   MOV     #PCK,TEMP1
022736  032737  000002  007704        BIT     #DATCKB,DEV4    ;SEE IF /CHECK OR /NOCHECK
022744  001003                        BNE     2$              ;BR IF /CHECK
022746  012737  013526  006532        MOV     #PNCK,TEMP1
022754  012737  013540  006534  2$:   MOV     #PEC,TEMP2
022762  032737  000004  007704        BIT     #ECHOB,DEV4     ;SEE IF /ECHO OR /NOECHO
022770  001003                        BNE     3$              ;BR IF /ECHO
022772  012737  013536  006534        MOV     #PNEC,TEMP2
```

```
 6405
 6422  023000  012737  013547  006542  3$:     MOV     #PMS,TEMP5
 6423  023006  032737  000010  007704          BIT     #MOCHK,DEV4     ;SEE IF /MODEM OR /NOMODEM
 6424  023014  001003                          BNE     5$              ;BR IF MODEM
 6425  023016  012737  013545  006542          MOV     #PNMS,TEMP5
 6426
 6427
 6428  023024                                  5$:     PRINTS  #SHF1,TEMP,TEMP1,TEMP2,TEMP5    ;,TEMP3,TEMP4 **RFU**
  (11) 023024  013746  006542                                                                          MOV     TEMP5,-(SP)
  (10) 023030  013746  006534                                                                          MOV     TEMP2,-(SP)
  (9)  023034  013746  006532                                                                          MOV     TEMP1,-(SP)
  (8)  023040  013746  006530                                                                          MOV     TEMP,-(SP)
  (7)  023044  012746  014247                                                                          MOV     #SHF1,-(SP)
  (6)  023050  012746  000005                                                                          MOV     #5,-(SP)
  (3)  023054  010600                                                                                  MOV     SP,R0
  (4)  023056  104416                                                                                  TRAP    C$PNTS
  (4)  023060  062706  000014                                                                          ADD     #14,SP
 6429  023064  000207                          RTS     PC              ;RETURN
 6430
 6431
```

```
                                .SBTTL          TRAVERSE COMMAND LINE SUBROUTINES

                                ;++
                                ;       P$TRV SUBROUITNE
                                ;
                                ;PARSE THE COMMAND LINE SUBROUTINE
                                ;TAKE ACTIONS (VIA ACTION TREE) AS PARSING LINE
                                ;PARSING DIRECTIONS FROM ''CLI PARSING NODES''
                                ;   REGS USED:
                                ;
                                ;       R1,R5=SCRATCH                           P$NUM=NUMERIC CODE FROM DATA
                                ;       R2=ACTION CODE PARAMETER FROM TREE
                                ;       R3=PARSE TREE POINTER
                                ;       R4=INPUT STRING POINTER
                                ; CALLING SEQUENCE:
                                ;       JSR     PC,P$TRV
                                ;--

023066                          P$TRV:
023066  013704  003126                  MOV     P$BUFA,R4
023072  013703  003130                  MOV     P$TREE,R3
023076  105714                 P$TR5:   TSTB    (R4)            ;SEE IF ANY CHARS LEFT IN INPUT STRING
023100  001441                          BEQ     P$EXIT          ;BR IF NO
023102  121327  000013                  CMPB    (R3),#11.       ;SEE IF SPECIAL CLI CHAR CODE OR ASCII
023106  003023                          BGT     20$             ;BR IF REGULAR ASCII CHAR.
023110  111305                          MOVB    (R3),R5         ;GET SPECIAL CHAR CODE INTO R5
023112  006305                          ASL     R5
023114  016505  023130                  MOV     10$(R5),R5      ;BUILD TRAVERSE ROUTINE ADDRESS
023120  062705  023130                  ADD     #10$,R5
023124  004715                          JSR     PC,(R5)         ;JSR TO SPECIAL CLI TRAVERSE ROUTINE
023126  000763                          BR      P$TR5           ;GO SEE IF MORE OF STRING LEFT

023130  000114                 10$:     .WORD   TRVERR-10$      ;TRAVERSE TABLE FOR ''CLI FUNTIONS''
023132  000134                          .WORD   TRVEXI-10$      ;1
023134  000152                          .WORD   TRVBR-10$       ;2
023136  000162                          .WORD   TRVBIF-10$      ;3
023140  000204                          .WORD   TRVSPA-10$      ;4
023142  000270                          .WORD   TRVNUM-10$      ;5
023144  000604                          .WORD   TRVALP-10$      ;6
023146  000650                          .WORD   TRVALN-10$      ;7
023150  000270                          .WORD   TRVOCT-10$      ;8
023152  000256                          .WORD   TRVDEC-10$      ;9
023154  000736                          .WORD   TRVSTR-10$      ;10

                                ;NOT A SPECIAL CODE

023156  121314                 20$:     CMPB    (R3),(R4)       ;SEE IF FIRST CHAR OF STRING IS A MATCH
023160  001403                          BEQ     22$             ;BR IF A MATCH
023162  004737  023226                  JSR     PC,TRVBRC       ;IF NOT A MATCH, GO TAKE MISS BRANCH
023166  000743                          BR      P$TR5           ; THEN GO BACK PT'G TO MISS NODE
023170  004737  023206          22$:    JSR     PC,TRVACT       ;IF A MATCH, GO DO ACTION DEFINED BY
023174  062703  000004                  ADD     #4,R3           ; ACTION CODE IN CLI NODE, THEN
                                                                ; ADJUST PTR TO NEXT CLI NODE
023200  005204                          INC     R4              ;ADJUST BUF PTR TO NEXT CHAR IF MATCH
023202  000735                          BR      P$TR5
```

```
 6489
 6490  023204  000207                  P$EXIT: RTS     PC                      ;RETURN FROM PARSER
 6491
 6492                                  ;-------------------------------------------------------------------------------
 6493
 6494                                  ;GOTO USER ACTION ROUTINE
 6495  023206  116302  000001          TRVACT: MOVB    1(R3),R2                ;GET ACTION CODE FROM CLI NODE
 6496  023212  042702  177400                  BIC     #177400,R2              ;CLEAR ANY SIGN EXTENSION
 6497  023216  013705  003132                  MOV     P$ACT,R5                ;GET ADDRESS OF CLI ACTION ROUTINE
 6498  023222  004715                          JSR     PC,(R5)                 ;GO DO ACTION DEFINED BY CODE
 6499  023224  000207                          RTS     PC                      ;RETURN TO CALLING CODE
 6500
 6501                                  ;TAKE BRANCH IN TREE
 6502  023226  016305  000002          TRVBRC: MOV     2(R3),R5                ;GET BRANCH DISPLACEMENT FROM TREE
 6503  023232  060503                          ADD     R5,R3                   ; AND POINT R3 TO THE 'MISS' NODE
 6504  023234  000207                          RTS     PC                      ; RETURN TO P$TRV
 6505
 6506                                  ;NO BRANCH TAKEN
 6507  023236  062703  000004          TRVNOB: ADD     #4,R3                   ;THINGS OK, UPDATE R3 TO POINT TO NEXT
 6508  023242  000207                          RTS     PC                      ; NODE AND RETURN TO P$TRV
 6509
 6510                                  ;-------------------------------------------------------------------------------
 6511  023244  C04737  023206          TRVERR: JSR     PC,TRVACT               ;TAKE ERROR ACTION
 6512  023250  112737  177777  003143          MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
 6513  023256  005726                          TST     (SP)+                   ;GET RID OF "JSR PUSH TO TRVERR"
 6514  023260  000137  023204                  JMP     P$EXIT                  ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
 6515
 6516  023264  004737  023206          TRVEXI: JSR     PC,TRVACT               ;TAKE EXIT ACTION
 6517  023270  105037  003143                  CLRB    P$GDBD                  ;SET GOOD/BAD FLAG TO "SUCCESS (0)"
 6518  023274  005726                          TST     (SP)+                   ;GET RID OF "JSR PUSH TO TRVEXI"
 6519  023276  000137  023204                  JMP     P$EXIT                  ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
 6520
 6521  023302  004737  023206          TRVBR:  JSR     PC,TRVACT               ;GO TAKE BRANCH ACTION
 6522  023306  000137  023226                  JMP     TRVBRC
 6523
 6524  023312  004737  023206          TRVBIF: JSR     PC,TRVACT
 6525  023316  105737  003143                  TSTB    P$GDBD                  ;SEE IF P$GDBD SET OR CLEARED BY ACTION
 6526  023322  001402                          BEQ     1$                      ;IF CLEAR FALL THRU TO NEXT NODE
 6527  023324  000137  023226                  JMP     TRVBRC                  ;ELSE TAKE THE 'MISS' BRANCH
 6528  023330  000137  023236          1$:     JMP     TRVNOB                  ;JUST UPDATE TO NEXT NODE IF THINGS OK
 6529
 6530  023334  005005                  TRVSPA: CLR     R5                      ;CLEAR "SPACE OR TAB FOUND" FLAG
 6531  023336  121427  000011          1$:     CMPB    (R4),#11                ;SEE IF CHAR. IN CMD LINE= TAB
 6532  023342  001003                          BNE     2$                      ;BR IF NO, NOT A TAB
 6533  023344  005204                          INC     R4                      ;INC INPUT STRING POINTER
 6534  023346  005205                          INC     R5                      ;INDICATE A TAB FOUND
 6535  023350  000772                          BR      1$                      ;GO CHECK NEXT CHAR
 6536
 6537  023352  121427  000040          2$:     CMPB    (R4),#40                ;SEE IF CHAR. IN CMD LINE= SPACE
 6538  023356  001003                          BNE     10$                     ;BR IF NO, NON-SPACE OR NON-TAB CHAR.
 6539  023360  005204                          INC     R4                      ;INC INPUT STRING POINTER
 6540  023362  005205                          INC     R5                      ;INDICATE A SPACE FOUND
 6541  023364  000764                          BR      1$                      ;GO CHECK NEXT CHAR
 6542  023366  005705          10$:    TST     R5                      ;SEE IF ANY SPACES OR TABS FOUND
 6543  023370  001404                          BEQ     15$                     ;BR IF NO, TAKE NO ACTION
 6544  023372  004737  023206                  JSR     PC,TRVACT               ;GO TAKE ACTION IF ANY FOUND
```

```
 6545  023376  000137  023236                    JMP     TRVNOB                  ;JUST GO UPDATE R3 TO NEXT NODE IF OK
 6546  023402  000137  023226            15$:    JMP     TRVBRC                  ;TAKE BRANCH (MISS) IF NONE FOUND
 6547
 6548
 6549  023406  012737  000012  003140    TRVDEC: MOV     #10.,P$RADX             ;USE DECIMAL AS RADIX AND ASSUME +
 6550  023414  000137  023426              TRVOCT: JMP     TRVNMA
 6551  023420                              ;(SAME AS TRVNUM SINCE DEFAULT RADIX IS OCTAL)
 6552  023420  012737  000010  003140    TRVNUM: MOV     #8.,P$RADX              ;USE OCTAL AS RADIX AND ASSUME +
 6553  023426  005005                      TRVNMA: CLR     R5                      ;CLEAR DIGIT COUNTER
 6554  023430  121427  000053                      CMPB    (R4),#'+                ;SEE IF THERE'S A + SIGN THERE
 6555  023434  001001                              BNE     10$                     ; BR IF NO
 6556  023436  000406                              BR      11$                     ; ELSE P$RADX ALREADY SAYS +, JUST BR
 6557  023440  121427  000055            10$:    CMPB    (R4),#'-                ;SEE IF THERE'S A - SIGN THERE
 6558  023444  001004                              BNE     1$                      ; BR IF NO
 6559  023446  112737  177777  003141            MOVB    #-1,P$RADX+1            ;SET 'MINUS FLAG' (HI BYTE OF P$RADX)
 6560  023454  005204                      11$:    INC     R4                      ;BUMP R4 TO POINT TO FIRST CHAR
 6561
 6562  023456  121427  000060            1$:     CMPB    (R4),#60                ;SEE IF CHAR. LESS THAN A '0'
 6563  023462  002434                              BLT     2$                      ;BR IF YES (NOT NUMERIC)
 6564  023464  121427  000067                      CMPB    (R4),#67                ;SEE IF CHAR. GREATER THAN A '7'
 6565  023470  003426                              BLE     13$                     ; BR IF YES
 6566  023472  123727  003140  000012              CMPB    P$RADX,#10.             ;SEE IF IN DECIMAL MODE
 6567  023500  001417                              BEQ     12$                     ; BR IF YES (CAN USE HIGHER LIMIT)
 6568  023502  121427  000071                      CMPB    (R4),#71                ;SEE IF DIGIT WAS A 8 OR 9
 6569  023506  003022                              BGT     2$                      ;BR IF NON-NUMERIC
 6570  023510                                      PRINTF  #CLIBRX                 ;ELSE WAS A 8 OR 9 WHEN IN OCTAL RADIX
   (7) 023510  012746  011650                                                                  MOV     #CLIBRX,-(SP)
   (6) 023514  012746  000001                                                                  MOV     #1,-(SP)
   (3) 023520  010600                                                                          MOV     SP,R0
   (4) 023522  104417                                                                          TRAP    C$PNTF
   (4) 023524  062706  000004                                                                  ADD     #4,SP
 6571  023530  112737  177777  003143            MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
 6572  023536  000474                              BR      5$                      ; PRINT ERROR AND TAKE MISS
 6573
 6574  023540  121427  000071            12$:    CMPB    (R4),#71                ;SEE IF CHAR. GREATER THAN A '9'
 6575  023544  003003                              BGT     2$                      ;BR IF YES (NOT NUMERIC)
 6576  023546  005204                      13$:    INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR.
 6577  023550  005205                              INC     R5                      ;INDICATE A NUMERIC FOUND
 6578  023552  000741                              BR      1$                      ;GO LOOK AT NEXT CHAR.
 6579
 6580  023554  005705                      2$:     TST     R5                      ;SEE IF FOUND ANY NUMERICS
 6581  023556  001464                              BEQ     5$                      ;BR IF NO, TAKE 'MISS' BRANCH
 6582  023560  010401                              MOV     R4,R1                   ;GET POINTER TO START OF NUMERIC STRING
 6583  023562  160501                              SUB     R5,R1
 6584  023564  005037  003136                      CLR     P$NUM                   ;CLEAR LOC. WHERE VALUE WILL BE STORED
 6585  023570  112102                      3$:     MOVB    (R1)+,R2                ;GET ASCII CHAR AND CONVERT IT TO A #
 6586  023572  162702  000060                      SUB     #60,R2
 6587  023576  006337  003136                      ASL     P$NUM                   ;SHIFT CURRENT VALUE TO MAKE ROOM
 6588  023602  103437                              BCS     7$                      ;ERROR IF NUMBER TOO BIG
 6589  023604  013737  003136  003134            MOV     P$NUM,P$CNT             ;SAVE FOR LATER IN CASE DECIMAL RADIX
 6590  023612  006337  003136                      ASL     P$NUM
 6591  023616  103431                              BCS     7$                      ;ERROR IF NUMBER TOO BIG
 6592  023620  006337  003136                      ASL     P$NUM
 6593  023624  103426                              BCS     7$                      ;ERROR IF NUMBER TOO BIG
 6594  023626  123727  003140  000012              CMPB    P$RADX,#10.             ;SEE IF DECIMAL RADIX
 6595  023634  001004                              BNE     4$                      ;BR IF NOT EQUAL
```

```
 6596  023636  063737  003134  003136          ADD     P$CNT,P$NUM
 6597  023644  103416                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
 6598  023646  060237  003136          4$:     ADD     R2,P$NUM
 6599  023652  103413                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
 6600  023654  005305                          DEC     R5
 6601  023656  001344                          BNE     3$
 6602  023660  105737  003141                  TSTB    P$RADX+1                ;SEE IF NUM WAS PRECEDED BY A - SIGN
 6603  023664  001402                          BEQ     15$                     ; BR IF NO
 6604  023666  005437  003136                  NEG     P$NUM                   ; ELSE NEGATE THE NUMBER BEFORE LEAVING
 6605  023672  004737  023206          15$:    JSR     PC,TRVACT               ;SINCE NUMERIC FOUND, GO TAKE ACTION
 6606  023676  000137  023236                  JMP     TRVNOB                  ;GO POINT R3 TO NEXT NODE
 6607
 6608  023702                          7$:     PRINTF  #CLINBG                 ;PRINT NUMBER TOO BIG ERROR
   (7) 023702  012746  011626                                                          MOV     #CLINBG,-(SP)
   (6) 023706  012746  000001                                                          MOV     #1,-(SP)
   (3) 023712  010600                                                                  MOV     SP,R0
   (4) 023714  104417                                                                  TRAP    C$PNTF
   (4) 023716  062706  000004                                                          ADD     #4,SP
 6609  023722  112737  177777  003143          MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
 6610  023730  000137  023226          5$:     JMP     TRVBRC                  ;TAKE 'MISS' BRANCH
 6611
 6612
 6613  023734  005005                  TRVALP: CLR     R5                      ;CLEAR ALPHA FOUND FLAG
 6614  023736  121427  000101          1$:     CMPB    (R4),#101               ;SEE IF CHAR. LESS THAN A "A"
 6615  023742  002406                          BLT     2$                      ;BR IF YES (NOT ALPHA)
 6616  023744  121427  000132                  CMPB    (R4),#132               ;SEE IF CHAR. GREATER THAN A "Z"
 6617  023750  003003                          BGT     2$                      ;BR IF YES (NOT ALPHA)
 6618  023752  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR
 6619  023754  005205                          INC     R5                      ;INDICATE AN ALPHA WAS FOUND
 6620  023756  000767                          BR      1$                      ;GO LOOK AT NEXT CHAR.
 6621  023760  005705          2$:     TST     R5                      ;SEE IF ANY ALPHA'S WERE FOUND
 6622  023762  001404                          BEQ     3$                      ;BR IF NO
 6623  023764  004737  023206                  JSR     PC,TRVACT               ;IF ANY FOUND TAKE ACTION
 6624  023770  000137  023236                  JMP     TRVNOB                  ;THEN UPDATE R3 TO NEXT NODE -NO BRANCH
 6625  023774  000137  023226          3$:     JMP     TRVBRC                  ;NONE FOUND, TAKE MISS BRANCH
 6626
 6627  024000  005005                  TRVALN: CLR     R5                      ;CLEAR ALPHANUM FOUND FLAG
 6628  024002  121427  000060          10$:    CMPB    (R4),#60                ;SEE IF CHAR. LESS THAN A "0"
 6629  024006  002417                          BLT     2$                      ;BR IF YES (NOT NUMERIC OR ALPHA)
 6630  024010  121427  000072                  CMPB    (R4),#72                ;SEE IF CHAR. GREATER THAN A "9"
 6631  024014  003003                          BGT     1$                      ;BR IF YES (NOT NUMERIC)
 6632  024016  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR.
 6633  024020  005205                          INC     R5                      ;INDICATE A NUMERIC FOUND
 6634  024022  000767                          BR      10$                     ;GO LOOK AT NEXT CHAR.
 6635  024024  121427  000101          1$:     CMPB    (R4),#101               ;SEE IF CHAR. LESS THAN A "A"
 6636  024030  002406                          BLT     2$                      ;BR IF YES (NOT ALPHA)
 6637  024032  121427  000132                  CMPB    (R4),#132               ;SEE IF CHAR. GREATER THAN A "Z"
 6638  024036  003003                          BGT     2$                      ;BR IF YES (NOT ALPHA)
 6639  024040  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR
 6640  024042  005205                          INC     R5                      ;INDICATE AN ALPHA FOUND
 6641  024044  000756                          BR      10$                     ;GO LOOK AT NEXT CHAR.
 6642  024046  005705          2$:     TST     R5                      ;SEE IF ANY ALPHANUM'S WERE FOUND
 6643  024050  001404                          BEQ     3$                      ;BR IF NO
 6644  024052  004737  023206                  JSR     PC,TRVACT               ;IF ANY FOUND TAKE ACTION
 6645  024056  000137  023236                  JMP     TRVNOB                  ;THEN UPDATE R3 TO NEXT NODE -NO BRANCH
 6646  024062  000137  023226          3$:     JMP     TRVBRC                  ;NONE FOUND, TAKE MISS BRANCH
```

```
                                ;
                                ;
                                ;
024066  010401          TRVSTR: MOV     R4,R1           ;POINT R1 TO CMD STRING
024070  010305                  MOV     R3,R5
024072  062705  000006          ADD     #6,R5           ;POUNT R5 TO MATCH STRING FROM CLI NODE
024076  005037  003134          CLR     P$CNT           ;CLEAR CHAR MATCH COUNT
024102  105715          2$:     TSTB    (R5)            ;SEE IF END OF MATCH STRING YET
024104  001411                  BEQ     10$             ;BR IF YES
024106  105711                  TSTB    (R1)            ;SEE IF END OF CMD LINE YET
024110  001407                  BEQ     10$             ;BR IF YES
024112  121115                  CMPB    (R1),(R5)       ;SEE IF CHARACTERS MATCH
024114  001005                  BNE     10$             ;BR IF NO
024116  005237  003134          INC     P$CNT           ;MATCH -INCREMENT MATCH COUNT
024122  005201                  INC     R1              ;UPDATE STRING POINTERS
024124  005205                  INC     R5
024126  000765                  BR      2$              ;BR TO CONTINUE CHECKING CHARS.
                                ;
024130  005737  003134  10$:     TST     P$CNT           ;WHEN DONE SEE IF ANY MATCHES FOUND
024134  001406                  BEQ     15$             ;BR IF NO, GO TAKE THE MISS BRANCH
024136  010104                  MOV     R1,R4           ;POINT CMD POINTER TO END OF STRING &
024140  004737  023206          JSR     PC,TRVACT       ;IF A MATCH FOUND, GO DO MATCH ACTION
024144  066303  000004          ADD     4(R3),R3        ;UPDATE R3 TO NEXT NODE (NO BRANCH)
024150  000207                  RTS     PC              ; (NO RETURN THRU TRVNOB SINCE DIFFERNT
                                ;                       ;  DISPLACEMENT DUE TO MATCH STRING)
024152  000137  023226  15$:     JMP     TRVBRC          ; GO TAKE BRANCH
                                ;
                                ;                       ; (PARSED OK), -1 IF ILL CMD.....
                                ;-------------------------------------------------------------------------
                                ;
```

```
 6678                                   .SBTTL  REPORT CODING SECTION
 6679
 6680
 6681                                   ;++
 6682                                   ; THE REPORT CODING SECTION CONTAINS THE
 6683                                   ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
 6684                                   ;--
 6685
 6686  024156                                   BGNRPT
  (3)  024156                                                                    L$RPT::
 6687
 6699
 6700  024156  004737  020302                   JSR     PC,REPORT              ;CALL SUBROUTINE TO DUMP EVENT LOG
 6701                                                                           ; AND BASE TABLE
 6702
 6709
 6710
 6717
 6718  024162                                   ENDRPT
  (3)  024162                                                                    L10010:
  (3)  024162  104425                                                                    TRAP    C$RPT
```

```
                                .SBTTL  PROTECTION TABLE

                                ;++
                                ; THIS TABLE IS USED BY THE RUNTIME SERVICES
                                ; TO PROTECT THE LOAD MEDIA.
                                ;--

                                        BGNPROT
                                                                        L$PROT::

                                        -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
                                        -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
                                        -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

                                        ENDPROT
```

```
                                .SBTTL  INITIALIZE SECTION

                                ;++
                                ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ; AT THE BEGINNING OF EACH PASS.
                                ;--

024172                                  BGNINIT
024172                                                                  L$INIT::

024172  005737  006556                  TST     DCLFLG                  ;CLEANUP AND EXIT? REV B EC
024176  001403                          BEQ     INIT1                   ;NO,BRANCH REV B EC
024200  005037  006556                  CLR     DCLFLG                  ;CLEAR FLAG REV B EC
024204                                  DOCLN                           ;GO CLEANUP AND EXIT REV B EC
024204  104444                                                                  TRAP    C$DCLN

024206  012737  177777  006560  INIT1:  MOV     #-1,RESFLG              ;SET RESTART FLAG
024214                                  READEF  #EF.START               ;IF HERE CAUSE OF START,DO SOME INIT
024214  012700  000040                                                          MOV     #EF.START,R0
024220  104447                                                                  TRAP    C$REFG
024222                                  BCOMPLETE       START
024222  103417                                                                  BCS     START
024224                                  READEF  #EF.RESTART             ;IF HERE CAUSE OF RESTART, DO SOME INIT
024224  012700  000037                                                          MOV     #EF.RESTART,R0
024230  104447                                                                  TRAP    C$REFG
024232                                  BCOMPLETE       RESTRT
024232  103513                                                                  BCS     RESTRT
024234                                  READEF  #EF.CONTINUE            ;SEE IF WE'RE HERE CAUSE OF A CONTINUE
024234  012700  000036                                                          MOV     #EF.CONTINUE,R0
024240  104447                                                                  TRAP    C$REFG
024242                                  BNCOMPLETE      S1              ;BR IF NOT HERE CAUSE OF CONITNUE
024242  103002                                                                  BCC     S1
024244  000137  024714                  JMP     ENDIT                   ;JMP IF HERE CAUSE OF A CONTINUE
024250                          S1:     READEF  #EF.NEW                 ;SEE IF THIS IS A 'NEW PASS''
024250  012700  000035                                                          MOV     #EF.NEW,R0
024254  104447                                                                  TRAP    C$REFG
024256                                  BCOMPLETE       NEW             ;IF YES, BR AROUND LOGUNIT # SETUP
024256  103521                                                                  BCS     NEW
024260  000523                          BR      GETPRM

024262  005037  006560          START:  CLR     RESFLG                  ;CLEAR RESTART FLAG SINCE HERE ON START
024266  005037  006620                  CLR     CLKVEC                  ;CLEAR CLK VECTOR PTR. AS A FLAG IN
                                                                        ; NO CLOCK IS FOUND.
024272  012702  006614                  MOV     #CLKCSR,R2              ;SETUP R2 AS A PTR. TO CLOCK INFO BLOCK
024276                                  CLOCK   L,R1                    ;LOOK FOR A LINE CLOCK
024276  012700  000114                                                          MOV     #'L,R0
024302  104462                                                                  TRAP    C$CLCK
024304  010001                                                                  MOV     R0,R1
024306                                  BNCOMPLETE      S2              ; IF NONE THERE GO LOOK FOR A P-CLOCK
024306  103006                                                                  BCC     S2
024310  004737  017422                  JSR     PC,CLKSET               ; GO SET UP CLOCK INFO TABLE & CLK VEC.
024314  012737  000100  006624          MOV     #LCLKEN,CLKEN           ;SETUP THE ENABLE LINE CLOCK DATA
024322  000457                          BR      RESTRT

024324                          S2:     CLOCK   P,R1                    ;LOOK FOR A P-CLOCK SINCE NO LINE CLOCK
```

```
  (3)  024324  012700  000120                                                      MOV     #'P,R0
  (3)  024330  104462                                                              TRAP    C$CLCK
  (3)  024332  010001                                                              MOV     R0,R1
 6810  024334                                   BNCOMPLETE      S3              ; IF NONE THERE GO SEE IF THIS IS LSI
  (2)  024334  103017                                                              BCC     S3
 6811  024336  004737  017422                   JSR     PC,CLKSET               ; ELSE GO SET UP CLOCK INFO & VECTOR
 6812  024342  062737  000002  006614           ADD     #2,CLKCSR               ;POINT CLKCSR TO P-CLK COUNT SET REG.
 6813  024350  012777  001600  162236           MOV     #PCLKCT,@CLKCSR         ;LOAD CLK SET REG. WITH COUNT VALUE
 6814  024356  162737  000002  006614           SUB     #2,CLKCSR               ;POINT CLKCSR BAC TO P-CLK CSR
 6815  024364  012737  000111  006624           MOV     #PCLKEN,CLKEN           ;SETUP THE ENABLE THE P-CLK DATA
 6816  024372  000433                           BR      RESTRT
 6817
 6818  024374                           S3:     READBUS                         ;READ BUS TYPE TO SEE IF ON AN LSI
  (3)  024374  104407                                                              TRAP    C$RDBU
 6819  024376                                   BNCOMPLETE      S4              ;BR IF NOT, NO CHANCE OF A CLOCK
  (2)  024376  103021                                                              BCC     S4
 6820  024400  012737  000100  006620           MOV     #100,CLKVEC             ;LOAD 100 AS CLK VECTOR
 6821  024406  005037  006616                   CLR     CLKBR                   ;LOAD 0 AS CLK INT. LEVEL
 6822  024412  012737  006624  006614           MOV     #CLKEN,CLKCSR           ;KLUDGE UP THE CSR & ENABLE DATA LOCS
 6823  024420                                   GMANID  L5060,CLKHZ,D,377,50.,60.,YES
  (3)  024420  104443                                                              TRAP    C$GMAN
  (3)  024422  000406                                                              BR      10000$
  (4)  024424  006622                                                              .WORD   CLKHZ
  (5)  024426  000052                                                              .WORD   T$CODE
  (5)  024430  013573                                                              .WORD   L5060
  (5)  024432  000377                                                              .WORD   377
  (5)  024434  000062                                                              .WORD   T$LOLIM
  (5)  024436  000074                                                              .WORD   T$HILIM
  (3)  024440                                                               10000$:
 6824  024440  000410                           BR      RESTRT
 6825
 6826  024442                           S4:     PRINTF  #BDCLK                  ;INFORM OPR. NO CLOCK, & EXIT INIT
  (7)  024442  012746  013712                                                      MOV     #BDCLK,-(SP)
  (6)  024446  012746  000001                                                      MOV     #1,-(SP)
  (3)  024452  010600                                                              MOV     SP,R0
  (4)  024454  104417                                                              TRAP    C$PNTF
  (4)  024456  062706  000004                                                      ADD     #4,SP
 6827
 6828  024462  005037  006626           RESTRT: CLR     TIMMIN                  ;CLEAR TIME SINCE START LOCATIONS
 6829  024466  005037  006630                   CLR     TIMSEC
 6830  024472  013737  006622  006632           MOV     CLKHZ,TIMTCK            ;LOAD TICKS/SEC
 6831  024500  012702  006644                   MOV     #EVTLOG,R2              ;INIT EVENT TABLE TO ALL 1'S AFTER EACH
 6832  024504  010237  006642                   MOV     R2,EVTPTR               ; START OR RES AND INIT TABLE POINTER
 6833  024510  012722  177777           1$:     MOV     #-1,(R2)+
 6834  024514  020227  007546                   CMP     R2,#EVTEND              ;SEE IF REACHED END OF TABLE
 6835  024520  001373                           BNE     1$                      ;LOOP UNTIL DONE
 6836
 6837  024522  012737  177777  006552   NEW:    MOV     #-1,LOGUNT              ;INITIALIZE LOGICAL UNIT #
 6838
 6839  024530  005237  006552           GETPRM: INC     LOGUNT                  ;POINT TO NEXT LOGICAL UNIT
 6840  024534  023737  006552  002012           CMP     LOGUNT,L$UNIT           ;SEE IF PAST MAX. LOG. UNIT #
 6841  024542  002367                           BGE     NEW                     ;BR IF YES, AND START OVER
 6842
 6843  024544                                   GPHARD  LOGUNT,R1               ;GET THE P-TABLE FOR THIS LOG. UNIT
  (3)  024544  013700  006552                                                      MOV     LOGUNT,R0
  (3)  024550  104442                                                              TRAP    C$GPHRD
```

```
  (3)  024552  010001                                                          MOV     R0,R1
6844   024554                       BNCOMPLETE      GETPRM          ;IF NO P-TABLE AVAIL., GO GET NEXT ONE
  (2)  024554  103365                                                          BCC     GETPRM
6845
6846   024556  011137  006566       MOV     (R1),FHDPLX             ;PUT FULL OR HALF DUPLEX ANSWER IN LOC.
6847
6858
6859                                ;DEVICE DEPENDENT PART OF GETTING INFO FROM P-TABLE
6860
6861   024562  016137  000002  011416  MOV     2(R1),RXCSR             ;STORE AWAY CSR ADDRESSES
6862
6863
6864   024570  016137  000002  011420  MOV     2(R1),PCSAR
6865   024576  062737  000002  011420  ADD     #2,PCSAR
6866   024604  016137  000002  011422  MOV     2(R1),RDSR
6867   024612  062737  000002  011422  ADD     #2,RDSR
6868   024620  016137  000002  011424  MOV     2(R1),TXCSR
6869   024626  062737  000004  011424  ADD     #4,TXCSR
6870   024634  016137  000002  011426  MOV     2(R1),TDSR
6871   024642  062737  000006  011426  ADD     #6,TDSR
6872
6873   024650  016137  000004  011430  MOV     4(R1),INVEC             ;STORE AWAY INPUT INTERRUPT VECTOR
6874   024656  016137  000004  011432  MOV     4(R1),OUTVEC
6875   024664  062737  000004  011432  ADD     #4,OUTVEC               ;BUILD OUTPUT INTERRUPT VECTOR
6876   024672  016137  000006  011434  MOV     6(R1),INTPRI            ;STORE AWAY INTERRUPT PRIORITY
6877   024700  016137  000012  011436  MOV     12(R1),OPTYP            ;STORE AWAY DEVICE OPTION TYPE
6878   024706  016137  000014  011472  MOV     14(R1),RNODE            ;STORE AWAY THE REMOTE NODE TYPE
6879   024714                   ENDIT:
6880   024714                       SETVEC  CLKVEC,#CLKINT,#340     ;SETUP CLOCK VECTOR
  (7)  024714  012746  000340                                                  MOV     #340,-(SP)
  (6)  024720  012746  017446                                                  MOV     #CLKINT,-(SP)
  (5)  024724  013746  006620                                                  MOV     CLKVEC,-(SP)
  (4)  024730  012746  000003                                                  MOV     #3,-(SP)
  (3)  024734  104437                                                          TRAP    C$SVEC
  (2)  024736  062706  000010                                                  ADD     #10,SP
6881
6882                                ;DEVICE DEPENDENT VECTOR SETUP
6883
6892   024742                       SETVEC  INVEC,#DVRXI,#PRI04     ;SETUP INPUT INTERRUPT VECTOR
  (7)  024742  012746  000200                                                  MOV     #PRI04,-(SP)
  (6)  024746  012746  034132                                                  MOV     #DVRXI,-(SP)
  (5)  024752  013746  011430                                                  MOV     INVEC,-(SP)
  (4)  024756  012746  000003                                                  MOV     #3,-(SP)
  (3)  024762  104437                                                          TRAP    C$SVEC
  (2)  024764  062706  000010                                                  ADD     #10,SP
6893   024770                       SETVEC  OUTVEC,#DVTXI,#PRI04    ;SETUP OUTPUT INTERRUPT VECTOR
  (7)  024770  012746  000200                                                  MOV     #PRI04,-(SP)
  (6)  024774  012746  034626                                                  MOV     #DVTXI,-(SP)
  (5)  025000  013746  011432                                                  MOV     OUTVEC,-(SP)
  (4)  025004  012746  000003                                                  MOV     #3,-(SP)
  (3)  025010  104437                                                          TRAP    C$SVEC
  (2)  025012  062706  000010                                                  ADD     #10,SP
6894
6895   025016                       SETPRI  #PRI00                  ;SET THE "RUN" PRIORITY TO 0
  (3)  025016  012700  000000                                                  MOV     #PRI00,R0
  (3)  025022  104441                                                          TRAP    C$SPRI
```

```
 6896  025024                          EXIT    INIT
  (3)  025024  104432                                          TRAP    C$EXIT
  (3)  025026  000002                                          .WORD   L10012-.
 6897
 6909
 6910                                  .EVEN
 6911
 6912  025030                          ENDINIT
  (3)  025030                                          L10012:
  (3)  025030  104411                                          TRAP    C$INIT
```

```
                                .SBTTL  AUTODROP SECTION

                                ;++
                                ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
                                ; THE ''ADR'' FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
                                ; SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
                                ; DROPPED FROM TESTING.
                                ;--

 6923  025032                           BGNAUTO
  (3)  025032                                                   L$AUTO::
 6924
 6931
 6932  025032                           ENDAUTO
  (3)  025032                                                   L10013:
  (3)  025032  104461                                                   TRAP    C$AUTO
```

```
                                .SBTTL  CLEANUP CODING SECTION

                                ;++
                                ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
                                ;--

                                        BGNCLN
                                                                                L$CLEAN::
                                        CLR     @CLKCSR         ;DISABLE CLOCK
                                        SETPRI  #PRI07          ;SET PROCESSOR PRIORITY BACK TO 7
                                                                                MOV     #PRI07,R0
                                                                                TRAP    C$SPRI
                                        BRESET                  ;CLEAR ALL BEFORE END
                                                                                TRAP    C$RESET
                                        EXIT    CLN
                                                                                TRAP    C$EXIT
                                                                                .WORD   L10014-.


                                        .EVEN

                                        ENDCLN
                                                                                L10014:
                                                                                TRAP    C$CLEAN
```

```
                                .SBTTL  DROP UNIT SECTION

                                ;++
                                ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ; TO NO LONGER BE TESTED.
                                ;--

                                        BGNDU
                                                                        L$DU::

                                        EXIT    DU
                                                                                .WORD   J$JMP
                                                                                .WORD   L10015-2-.

                                        .EVEN

                                        ENDDU
                                                                        L10015:
                                                                                TRAP    C$DU
```

```
 7009                                   .SBTTL  ADD UNIT SECTION
 7010
 7011                                   ;++
 7012                                   ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
 7013                                   ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
 7014                                   ;  TO THE TEST CYCLE.
 7015                                   ;--
 7016
 7017  025064                           BGNAU
  (3)  025064                                                          L$AU::
 7018
 7027
 7028  025064                           EXIT    AU
  (4)  025064  000167                                                          .WORD   J$JMP
  (3)  025066  000000                                                          .WORD   L10016-2-.
 7029
 7041
 7042                                   .EVEN
 7043
 7044  025070                           ENDAU
  (3)  025070                                                          L10016:
  (3)  025070  104452                                                          TRAP    C$AU
 7045
 7046
```

```
                                        .SBTTL  TEST 1:  SETUP AND MODES OF OPERATION


                                        ;++
                                        ; TEST TO DETECT FAULTS IN THE DATA COMMUNICATION LINK.  THIS TEST WILL
                                        ; THE PROVIDE COVERAGE NECESSARY TO ISOLATE FAILURES TO THE COMPUTER
                                        ; EQUIPMENT, THE COMMUNICATION LINK, OR THE MODEM.
                                        ;--


025072                                          BGNTST
(3)  025072                                                                                     T1::

                                        .SBTTL          PROGRAM SETUP SECTION

025072  013777  006624  161514                  MOV     CLKEN,@CLKCSR   ;ENABLE THE CLOCK

025100                                  GTXRXB:
025100  005001                          GTRA2:  CLR     R1
025102  012737  000001  006634                  MOV     #1,TIMER1       ;SET TIMER TO COUNT 1 TICK
025110  005737  006634                  1$:     TST     TIMER1          ;CHECK FOR IT TO BE COUNTED OFF
025114  001412                                  BEQ     GTRA3           ;BRANCH IF CLOCK EXISTS (COUNTED A TICK)
025116  005301                                  DEC     R1
025120  001373                                  BNE     1$              ;KEEP CHECKING UNTIL R1 DOES FULL COUNTDWN
025122                                          PRINTF  #NOCLK          ;PRINT BAD CLK MSG AND WARN OF HANG IF TIMEOUT
(7)  025122  012746  013736                                                             MOV     #NOCLK,-(SP)
(6)  025126  012746  000001                                                             MOV     #1,-(SP)
(3)  025132  010600                                                                     MOV     SP,R0
(4)  025134  104417                                                                     TRAP    C$PNTF
(4)  025136  062706  000004                                                             ADD     #4,SP

025142  005737  006560                  GTRA3:  TST     RESFLG          ;SEE IF HERE AFTER A RESTART.
025146  001112                                  BNE     GTRA5           ;BR IF HERE CAUSE OF A RESTART

                                                ; CLEAR COUNTS AND SET UP DEFAULTS

025150  005037  006524                  GTRA4:  CLR     TOTCC           ;CLEAR TOTAL CHAR. COUNT TEMP. LOC.
025154  005037  006460                          CLR     TTOTCC          ; CLEAR TOTAL CHAR. COUNT FOR TX BUFF
025160  005037  006444                          CLR     CTOTCC          ; CLEAR TOTAL CHAR. COUNT FOR CMP BUFF
025164  012701  006144                          MOV     #PTRTAB,R1      ;INIT TRANSMIT MESSAGE POINTER
025170  010137  006436                          MOV     R1,TXPTR
025174  005037  006434                          CLR     RXPTR           ; ZERO RX POINTER

025200  012737  006240  006440                  MOV     #PTR13,CMPPTR   ;INIT COMPARE MESSAGE POINTER

025206  012737  000005  006512                  MOV     #5,MSGTYP       ;SET UP DEFAULT MSG TYPE (QUICK FOX - ITEP MSG)
025214  013737  002162  006514                  MOV     MSG5C,CURCC     ;SET UP DEFAULT CHAR COUNT
025222  012737  003144  006462                  MOV     #TXBUF,TCURAD   ;SET UP CURRENT ADD TO START OF TX BUFFER
025230  012737  005144  006446                  MOV     #CMPBUF,CCURAD  ;SET UP CURRENT ADD TO START OF CMP BUFFER

025236  013737  006462  006522                  MOV     TCURAD,CURADD   ;SETUP CURRENT ADDR TO START OF TXBUF
025244  013737  006436  006520                  MOV     TXPTR,CPTR      ;SETUP CURRENT POINTER TABLE POINTER FOR TXBUF
025252  004737  022234                          JSR     PC,BLDBUF       ; GO BUILD POINTER TABLE AND BUFFER
025256  012737  000001  006456                  MOV     #1,TXMTOT       ;BUMP TOTAL MESSAGE COUNT
```

```
7114
7115  025264  013737  006440  006520          MOV     CMPPTR,CPTR     ;SET UP START OF COMPARE POINTER TABLE
7116  025272  013737  006446  006522          MOV     CCURAD,CURADD   ;SET UP CURRENT ADDR. TO START OF CMPBUF
7117  025300  012737  000005  006512          MOV     #5,MSGTYP
7118  025306  013737  002162  006514          MOV     MSG5C,CURCC
7119  025314  004737  022234                  JSR     PC,BLDBUF       ;PUT DEFAULT MESSAGE INTO CMPBUF
7120  025320  012737  000001  006442          MOV     #1,CMPTOT       ;BUMP THE COMP MESG COUNT
7121  025326  012737  000003  006562          MOV     #ACT,MODTYP     ;SET DEFAULT MODE= ACTIVE
7122  025334  005037  006564                  CLR     MLTYP           ;SET DEFAULT MAINTENANCE LOOP MODE =NONE
7123  025340  012737  000001  006572          MOV     #1,RPASS        ;SET UP DEFAULT ''RUN PASS'' COUNT TO 1
7124  025346  012737  000002  006570          MOV     #2,PARAM        ;SET UP PROG. PARAMETERS - DATACHECKING ENABLED
7125                                          ;                               OPERATOR STATUS MSGS. PRINT OFF
7126  025354                                  PRINTF  #HLP0
 (7)  025354  012746  012163                                                    MOV     #HLP0,-(SP)
 (6)  025360  012746  000001                                                    MOV     #1,-(SP)
 (3)  025364  010600                                                            MOV     SP,R0
 (4)  025366  104417                                                            TRAP    C$PNTF
 (4)  025370  062706  000004                                                    ADD     #4,SP
7127  025374  013737  006562  007676  GTRA5:  MOV     MODTYP,DEV1
7128  025402  013737  006564  007700          MOV     MLTYP,DEV2
7129  025410  013737  006572  007702          MOV     RPASS,DEV3
7130  025416  013737  006570  007704          MOV     PARAM,DEV4
7131  025424  004737  022564                  JSR     PC,SHWOP        ;PRINT TO OPERATOR THE CURRENT MODE.....
7132
7133  025430                                  MANUAL                  ;SEE IF MANUAL INTERVENTION ALLOWED
 (3)  025430  104450                                                            TRAP    C$MANI
7134  025432                                  BCOMPLETE       GETCL   ;  BR IF YES (UAM=0 AND NOT CHAINED)
 (2)  025432  103412                                                            BCS     GETCL
7135  025434  005737  006572                  TST     RPASS           ;SEE IF THIS IS FIRST ''DCLT PASS''
7136  025440  001002                          BNE     1$              ; BR IF NOT COMPLETED 1 PASS
7137  025442                                  EXIT    TST             ; IF DONE 1 PASS IN UNATTENDED MODE - EXIT
 (3)  025442  104432                                                            TRAP    C$EXIT
 (3)  025444  007676                                                            .WORD   L10017-.
7138  025446  012737  000001  006564  1$:     MOV     #ITL,MLTYP      ;SET UP DEFAULT FOR UNATTENDED MODE
7139  025454  000137  030432                  JMP     GTR9            ;  ''R M=ACT/LO=I/PAS=1/NOST/CH'' AND RUN
7140
7141                                  .SBTTL          COMMAND LINE FETCH & INTERPRETATION SECTION
7142
7143  025460  105037  003143          GETCL:  CLRB    P$GDBD          ;CLEAR CMD LINE PARSING ERROR FLAGS
7144  025464  105037  003142                  CLRB    P$NNUF
7145  025470                                  GMANID  CLI$PM,CMDBUF,A,0,1,72.,NO      ;GET A COMMAND LINE FROM OPR.
 (3)  025470  104443                                                            TRAP    C$GMAN
 (3)  025472  000406                                                            BR      10000$
 (4)  025474  002666                                                            .WORD   CMDBUF
 (5)  025476  000142                                                            .WORD   T$CODE
 (5)  025500  011540                                                            .WORD   CLI$PM
 (5)  025502  000000                                                            .WORD   0
 (5)  025504  000001                                                            .WORD   T$LOLIM
 (5)  025506  000110                                                            .WORD   T$HILIM
 (3)  025510                                                            10000$:
7146  025510  012737  002666  003126          MOV     #CMDBUF,P$BUFA
7147  025516  012737  007706  003130          MOV     #CLITRE,P$TREE
7148  025524  012737  026434  003132          MOV     #CLIACT,P$ACT
7149  025532  005037  003012                  CLR     QUALFG                  ;CLEAR QUALIFIER FLAG LOCATION
7150  025536  004737  023066                  JSR     PC,P$TRV                ;GO PARSE COMMAND LINE
7151  025542  105737  003143                  TSTB    P$GDBD                  ;SEE IF PARSED OK OR AN ERROR
```

```
 7152  025546  001412                          BEQ     1$
 7153  025550                                  PRINTF  #CLIERM
  (7)  025550  012746  011553                                                    MOV     #CLIERM,-(SP)
  (6)  025554  012746  000001                                                    MOV     #1,-(SP)
  (3)  025560  010600                                                            MOV     SP,R0
  (4)  025562  104417                                                            TRAP    C$PNTF
  (4)  025564  062706  000004                                                    ADD     #4,SP
 7154  025570  000137  025460                  JMP     GETCL
 7155  025574  105737  003142          1$:     TSTB    P$NNUF          ;SEE IF INCOMPLETE COMMAND TYPED
 7156  025600  001412                          BEQ     10$
 7157  025602                                  PRINTF  #CLINUF
  (7)  025602  012746  011603                                                    MOV     #CLINUF,-(SP)
  (6)  025606  012746  000001                                                    MOV     #1,-(SP)
  (3)  025612  010600                                                            MOV     SP,R0
  (4)  025614  104417                                                            TRAP    C$PNTF
  (4)  025616  062706  000004                                                    ADD     #4,SP
 7158  025622  000137  025460                  JMP     GETCL
 7159
 7160  025626  023727  003010  000060  10$:     CMP     KEYWD1,#SETET   ;WAS''SET EXPECT=TRANSMIT'' TYPED ? REVB EC
 7161  025634  001711                          BEQ     GETCL           ;YES,BRANCH REV B EC
 7162  025636  023727  003010  000003          CMP     KEYWD1,#HLP     ;SEE IF HELP WAS TYPED
 7163  025644  001705                          BEQ     GETCL           ;GO GET CMD AGAIN IF YES
 7164  025646  023727  003010  000055          CMP     KEYWD1,#PRNT    ;SEE IF PRINT WAS TYPED
 7165  025654  001701                          BEQ     GETCL           ; GO GET CMD AGAIN IF YES
 7166  025656  023727  003010  000004          CMP     KEYWD1,#RUN     ;SEE IF RUN WAS TYPED
 7167  025664  001002                          BNE     11$             ; BR IF NO
 7168  025666  000137  030432                  JMP     GTR9            ; START EXEC. IF YES
 7169  025672  023727  003010  000052  11$:     CMP     KEYWD1,#DMPS    ;SEE IF DUMP WAS TYPED
 7170  025700  001004                          BNE     12$             ; BR IF NO
 7171  025702  004737  022000                  JSR     PC,DUMPSR       ; ELSE, DUMP PART OF MEMORY
 7172  025706  000137  025460                  JMP     GETCL           ; THEN RETURN TO GET ANOTHER CMD.
 7173  025712  023727  003010  000057  12$:     CMP     KEYWD1,#EXIT    ;EXIT ? REV B EC
 7174  025720  001005                          BNE     13$             ;NO,BRANCH REV B EC
 7175  025722  012737  000001  006556          MOV     #1,DCLFLG       ;SET CLEANUP FLAG REV B EC
 7176  025730                                  EXIT    TST             ;GO BACK TO INIT REV B EC
  (3)  025730  104432                                                            TRAP    C$EXIT
  (3)  025732  007410                                                            .WORD   L10017-.
 7177
 7178  025734  023727  003010  000001  13$:     CMP     KEYWD1,#CLEAR   ;SEE IF CLEAR WAS TYPED
 7179  025742  001646                          BEQ     GETCL           ; IF YES, BACK TO GET ANOTHER CMD.
 7180  025744  023727  003010  000002          CMP     KEYWD1,#SHOW    ;SEE IF SHOW WAS TYPED
 7181  025752  001642                          BEQ     GETCL           ; IF YES, BACK TO GET ANOTHER CMD.
 7182  025754  023727  003010  000010  4$:     CMP     KEYWD1,#SETEXP  ;SEE IF SET EXPECTED
 7183  025762  001512                          BEQ     2$              ; BR IF YES (A SETEXP WAS TYPED)
 7184  025764  013737  006460  006524  5$:     MOV     TTOTCC,TOTCC
 7185  025772  023727  006524  001000          CMP     TOTCC,#BUFLIM   ;SEE IF BUFFER ALREADY FULL
 7186  026000  002414                          BLT     15$             ; BR IF NOT FULL (BUFLIM # OF CHARS.)
 7187  026002                                  PRINTF  #MSGTRN,#BUFEX  ; ELSE TELL OPR. AND DON'T BUILD MSG.
  (8)  026002  012746  014057                                                    MOV     #BUFEX,-(SP)
  (7)  026006  012746  014075                                                    MOV     #MSGTRN,-(SP)
  (6)  026012  012746  000002                                                    MOV     #2,-(SP)
  (3)  026016  010600                                                            MOV     SP,R0
  (4)  026020  104417                                                            TRAP    C$PNTF
  (4)  026022  062706  000006                                                    ADD     #6,SP
 7188  026026  000137  025460                  JMP     GETCL           ; THEN GO GET A NEW COMMAND
 7189  026032  005737  006460          15$:    TST     TTOTCC          ;IF FIRST "SET" THEN GET RID OF DEFAULT
```

```
 7190  026036  001002                          BNE     6$
 7191  026040  005037  006456                  CLR     TXMTOT
 7192  026044  012737  006144  006436  6$:     MOV     #PTRTAB,TXPTR           ;GET POSITION OF END OF TX LIST
 7193  026052  013701  006456                  MOV     TXMTOT,R1
 7194  026056  020127  000017                  CMP     R1,#MSGLIM              ;SEE IF MSG COUNT EXCEEDED.
 7195  026062  002414                          BLT     17$                     ; BR IF NO
 7196  026064                                  PRINTF  #MSGTRN,#TABEX          ; ELSE TELL OPR. AND DON'T BUILD MSG.
   (8) 026064  012746  014017                                                          MOV     #TABEX,-(SP)
   (7) 026070  012746  014075                                                          MOV     #MSGTRN,-(SP)
   (6) 026074  012746  000002                                                          MOV     #2,-(SP)
   (3) 026100  010600                                                                  MOV     SP,R0
   (4) 026102  104417                                                                  TRAP    C$PNTF
   (4) 026104  062706  000006                                                          ADD     #6,SP
 7197  026110  000137  025460                  JMP     GETCL                   ; THEN GO GET A NEW COMMAND.
 7198  026114  006301                  17$:    ASL     R1                      ;# OF MSGS *4 = NEXT FREE PTR BLOCK
 7199  026116  006301                          ASL     R1
 7200  026120  060137  006436                  ADD     R1,TXPTR
 7201  026124  013737  006436  006520          MOV     TXPTR,CPTR              ;SETUP CHAR. COUNT, CURRENT ADDR, & PTR
 7202  026132  013737  006462  006522          MOV     TCURAD,CURADD
 7203  026140  004737  022136                  JSR     PC,ADDCC                ;ADD IN CHAR. COUNT AND CHECK TOTAL
 7204  026144  004737  022234                  JSR     PC,BLDBUF               ;GO BUILD MESSAGE IN BUFFER AND PTRS.
 7205  026150  013737  006520  006436          MOV     CPTR,TXPTR
 7206  026156  013737  006524  006460          MOV     TOTCC,TTOTCC            ;UPDATE CHAR. COUNT, CURR ADDR, & PTR
 7207  026164  013737  006522  006462          MOV     CURADD,TCURAD
 7208  026172  005237  006456                  INC     TXMTOT
 7209  026176  005337  003014                  DEC     QUALVL                  ;DEC THE COPY COUNT
 7210  026202  001270                          BNE     5$
 7211  026204  000137  025460                  JMP     GETCL
 7212
 7213  026210  013737  006444  006524  2$:     MOV     CTOTCC,TOTCC            ;SETUP CHAR. COUNT, CURR. ADDR. & PTR
 7214  026216  023727  006524  001000          CMP     TOTCC,#BUFLIM           ;SEE IF BUFFER ALREADY FULL
 7215  026224  002414                          BLT     16$                     ; BR IF NOT FULL (BUFLIM # OF CHARS.)
 7216  026226                                  PRINTF  #MSGTRN,#BUFEX          ; ELSE TELL OPR. AND DON'T BUILD MSG.
   (8) 026226  012746  014057                                                          MOV     #BUFEX,-(SP)
   (7) 026232  012746  014075                                                          MOV     #MSGTRN,-(SP)
   (6) 026236  012746  000002                                                          MOV     #2,-(SP)
   (3) 026242  010600                                                                  MOV     SP,R0
   (4) 026244  104417                                                                  TRAP    C$PNTF
   (4) 026246  062706  000006                                                          ADD     #6,SP
 7217  026252  000137  025460                  JMP     GETCL                   ; THEN GO GET A NEW COMMAND
 7218  026256  005737  006444          16$:    TST     CTOTCC                  ;IF FIRST "SET" THEN GET RID OF DEFAULT
 7219  026262  001002                          BNE     7$
 7220  026264  005037  006442                  CLR     CMPTOT
 7221  026270                          7$:
 7222  026270  012737  006240  006440          MOV     #PTR13,CMPPTR           ;INIT COMPARE MESSAGE POINTER
 7223  026276  013701  006442                  MOV     CMPTOT,R1
 7224  026302  020127  000017                  CMP     R1,#MSGLIM              ;SEE IF MSG COUNT EXCEEDED.
 7225  026306  002414                          BLT     18$                     ; BR IF NO
 7226  026310                                  PRINTF  #MSGTRN,#TABEX          ; ELSE TELL OPR. AND DON'T BUILD MSG.
   (8) 026310  012746  014017                                                          MOV     #TABEX,-(SP)
   (7) 026314  012746  014075                                                          MOV     #MSGTRN,-(SP)
   (6) 026320  012746  000002                                                          MOV     #2,-(SP)
   (3) 026324  010600                                                                  MOV     SP,R0
   (4) 026326  104417                                                                  TRAP    C$PNTF
   (4) 026330  062706  000006                                                          ADD     #6,SP
 7227  026334  000137  025460                  JMP     GETCL                   ; THEN GO GET A NEW COMMAND.
```

```
7228  026340  006301                  18$:    ASL     R1              ;# OF MSGS *4 = NEXT FREE PTR BLOCK
7229  026342  006301                          ASL     R1
7230  026344  060137  006440                  ADD     R1,CMPPTR
7231  026350  013737  006440  006520          MOV     CMPPTR,CPTR
7232  026356  013737  006446  006522          MOV     CCURAD,CURADD
7233  026364  004737  022136                  JSR     PC,ADDCC        ;ADD IN XHAR. COUNT AND CHECK TOTAL
7234  026370  004737  022234                  JSR     PC,BLDBUF
7235  026374  013737  006520  006440          MOV     CPTR,CMPPTR
7236  026402  005237  006442                  INC     CMPTOT
7237  026406  013737  006522  006446          MOV     CURADD,CCURAD   ;UPDATE CHAR. COUNT, CURR ADDRR. & PTR
7238  026414  013737  006524  006444          MOV     TOTCC,CTOTCC
7239  026422  005337  003014                  DEC     QUALVL          ;IF COPY WAS GIVEN, PUT MSG IN BUFF
7240  026426  001270                          BNE     2$              ; AGAIN
7241  026430  000137  025460                  JMP     GETCL           ;GO BACK UNTIL GET A 'RUN'
7242
7243
7244
7245
7246
```

```
                                .SBTTL          ACTION TABLE AND ROUTINES
                                ;       USER MUST CLEAR/SET P$GDBD IF USE "CLIBIF" IN CONNECTION WITH ACTION
                                ;       R2 WILL HOLD ACTION CODE FROM PARSING (CLI) NODE
                                ;
026434                          CLIACT:
026434  006302                          ASL     R2              ;MULTIPLY ACTION CODE BY 2
026436  016202  026452                  MOV     10$(R2),R2      ;OFFSET VALUE
026442  062702  026452                  ADD     #10$,R2         ;ADD BASE VALUE
026446  004712                          JSR     PC,(R2)         ;GO DO ACTION
026450  000207                          RTS     PC              ;RETURN TO TRVACT:

                                                                ;BRIEF DESCRIPTION OF ACTONS TAKEN
026452  000150                  10$:    .WORD   ACTNUL-10$      ;NULL
026454  000152                          .WORD   ACTCLR-10$      ;CLEAR
026456  000162                          .WORD   ACTSHO-10$      ;SHOW
026460  001550                          .WORD   ACTCHK-10$      ;CHECK
026462  000262                          .WORD   ACTRUN-10$      ;RUN
026464  000172                          .WORD   ACTHLP-10$      ;HELP
026466  000306                          .WORD   ACTCSE-10$      ;CLEAR OR SHOW EXPECT
026470  000400                          .WORD   ACTCST-10$      ;CLEAR OR SHOW TRANSMIT
026472  000722                          .WORD   ACTSTE-10$      ;SET EXPECT
026474  000732                          .WORD   ACTSTT-10$      ;SET TRANSMIT
026476  000750                          .WORD   ACTSZE-10$      ;SIZE
026500  000760                          .WORD   ACTCOP-10$      ;COPY
026502  000770                          .WORD   ACTNUM-10$      ;NUMERIC VALUE FOR SIZE OR COPY
026504  001062                          .WORD   ACTOPM-10$      ;QUOTED MESSAGE FROM USER
026506  001556                          .WORD   ACTSTS-10$      ;STATUS
026510  001102                          .WORD   ACTEQQ-10$      ;END OF QUOTED MESSAGE FROM USER
026512  001162                          .WORD   ACTMS0-10$      ;ONES AS DATA
026514  001170                          .WORD   ACTMS1-10$      ;ZEROS AS DATA
026516  001200                          .WORD   ACTMS2-10$      ;1ALT AS DATA
026520  001210                          .WORD   ACTMS3-10$      ;0ACT AS DATA
026522  001220                          .WORD   ACTMS4-10$      ;ITEP AS DATA
026524  001230                          .WORD   ACTMS5-10$      ;CCITT AS DATA
026526  001246                          .WORD   ACTMS6-10$      ;ALPHA AS DATA
026530  001334                          .WORD   ACTATV-10$      ;ACTIVE MODE
026532  001344                          .WORD   ACTPAS-10$      ;PASSIVE MODE
026534  001364                          .WORD   ACTREC-10$      ;RECEIVE MODE
026536  001372                          .WORD   ACTLIS-10$      ;LISTEN MODE
026540  001402                          .WORD   ACTDLL-10$      ;DOWNLINE LOAD
026542  001412                          .WORD   ACTTRA-10$      ;TRANSMIT MODE
026544  001422                          .WORD   ACTTAL-10$      ;TALK MODE
026546  001450                          .WORD   ACTNO-10$       ;NO   IE /NOCHECK
026550  001460                          .WORD   ACTECH-10$      ;ECHO
026552  001564                          .WORD   ACTCRC-10$      ;CRC
026554  001600                          .WORD   ACTPRO-10$      ;PROTOCOL
026556  001632                          .WORD   ACTRPS-10$      ;STATUS
026560  001642                          .WORD   ACTMOP-10$      ;SATELLITE IN MAINTENANCE LOOP MODE
026562  001652                          .WORD   ACTTLP-10$      ;INTERNALTTL
026564  001662                          .WORD   ACTCLP-10$      ;CABLE LOOP
026566  001672                          .WORD   ACTLLP-10$      ;LOCAL MODEM LOOP
026570  001702                          .WORD   ACTRLP-10$      ;REMOTE MODEM LOOP
026572  000142                          .WORD   ACTNUF-10$      ;MORE COMMAND NEEDED
026574  001140                          .WORD   ACTBCR-10$      ;BAD CHARACTER IN OPERATOR MESSAGE
026576  000656                          .WORD   ACTDMS-10$      ;DUMP MEMORY START ADDRESS
026600  000706                          .WORD   ACTDME-10$      ;DUMP MEMORY END ADDRESS
```

```
7304  026602  000700                .WORD   ACTDMQ-10$      ;DUMP WORD
7305  026604  000246                .WORD   ACTPRT-10$      ;PRINT
7306  026606  001572                .WORD   ACTMOS-10$      ;MODEM STATUS
7307  026610  000236                .WORD   ACTEXT-10$      ;EXIT ROUTINE REV B EC
7308  026612  001272                .WORD   ACTSEX-10$      ;SET EX=TR REV B EC
7309
```

```
 7311
 7312  026614  112737  177777  003142  ACTNUF: MOVB    #-1,P$NNUF      ;SET FLAG TO SAY NEED MORE OF COMMAND
 7313  026622  000207                  ACTNUL: RTS     PC              ;RETURN TO PARSER
 7314
 7315  026624  012737  000001  003010  ACTCLR: MOV     #CLEAR,KEYWD1   ;SET LOC TO SAY A CLEAR WAS TYPED
 7316  026632  000207                          RTS     PC
 7317
 7318  026634  012737  000002  003010  ACTSHO: MOV     #SHOW,KEYWD1    ;SET LOC. TO SAY A SHOW WAS TYPED
 7319  026642  000207                          RTS     PC
 7320
 7321  026644  012702  003016          ACTHLP: MOV     #HLPTAB,R2      ;SETUP R2 AS A POINTER TO HELP MSG TABLE
 7322  026650                          1$:     PRINTF  #HLPF,(R2)+     ;PRINT HELP INFORMATION MESSAGES
   (8)  026650  012246                                                             MOV     (R2)+,-(SP)
   (7)  026652  012746  012241                                                     MOV     #HLPF,-(SP)
   (6)  026656  012746  000002                                                     MOV     #2,-(SP)
   (3)  026662  010600                                                             MOV     SP,R0
   (4)  026664  104417                                                             TRAP    C$PNTF
   (4)  026666  062706  000006                                                     ADD     #6,SP
 7323  026672  020227  003036                  CMP     R2,#HLPEND      ;SEE IF ALL INFO PRINTED YET
 7324  026676  001364                          BNE     1$              ;IF NO KEEP PRINTING
 7325  026700  012737  000005  003010          MOV     #HLP,KEYWD1     ;SET LOC. TO SAY A HELP WAS TYPED
 7326  026706  000207                          RTS     PC
 7327
 7328  026710  012737  000057  003010  ACTEXT: MOV     #EXIT,KEYWD1    ;EXIT COMMAND WAS INPUT REV B EC
 7329  026716  000207                          RTS     PC              ;RETURN
 7330
 7331  026720  012737  000055  003010  ACTPRT: MOV     #PRNT,KEYWD1    ;SET LOC. TO SAY A HELP WAS TYPED
 7332  026726  004737  020302                  JSR     PC,REPORT       ;CALL ROUTINE TO PRINT EVENT LOG AND BASE TABLE
 7333  026732  000207                          RTS     PC
 7334
 7335  026734  012737  000004  003010  ACTRUN: MOV     #RUN,KEYWD1     ;SET RUN FLAG
 7336  026742  112737  177777  003142          MOVB    #-1,P$NNUF      ;SET FLAG TO SAY NEED MORE OF COMMAND
 7337  026750  012737  000001  006572          MOV     #1,RPASS        ;SET DEFAULT RUN 'PASS' TO 1
 7338  026756  000207                          RTS     PC
 7339
 7340  026760  012737  006240  006440  ACTCSE: MOV     #PTR13,CMPPTR   ;INIT COMPARE MESSAGE POINTER
 7341  026766  013701  006440                  MOV     CMPPTR,R1
 7342
 7343  026772  013702  006442                  MOV     CMPTOT,R2
 7344  026776  105037  003142                  CLRB    P$NNUF          ;FLAG THAT HAVE VALID COMMAND AT THIS PT.
 7345  027002  023727  003010  000002          CMP     KEYWD1,#SHOW    ;SEE IF A CLEAR OR SHOW WAS TYPED
 7346  027010  001471                          BEQ     ACTSHW          ;BR IF A SHOW WAS TYPED
 7347  027012  012737  000001  006442          MOV     #1,CMPTOT       ;CLEAR COMPARE MESSAGE COUNT, CHAR. COUNT
 7348  027020  005037  006444                  CLR     CTOTCC          ; AND RESET POINTER
 7349
 7350  027024  012737  006240  006440          MOV     #PTR13,CMPPTR   ;INIT COMPARE MESSAGE POINTER
 7351  027032  013737  006440  006520          MOV     CMPPTR,CPTR     ;SET UP TO FILL IN DEFAULT MESSAGE
 7352  027040  012701  005144                  MOV     #CMPBUF,R1
 7353  027044  010137  006446                  MOV     R1,CCURAD
 7354  027050  000431                          BR      ACTCLB
 7355
 7356  027052  012701  006144          ACTCST: MOV     #PTRTAB,R1
 7357  027056  013702  006456                  MOV     TXMTOT,R2
 7358  027062  105037  003142                  CLRB    P$NNUF          ;FLAG THAT HAVE VALID COMMAND AT THIS PT.
 7359  027066  023727  003010  000002          CMP     KEYWD1,#SHOW    ;SEE IF A CLEAR OR SHOW WAS TYPED
 7360  027074  001437                          BEQ     ACTSHW          ;BR IF A SHOW WAS TYPED
```

```
 7361  027076  012737  000001  006456          MOV     #1,TXMTOT       ;CLEAR TRANSMIT MESSAGE COUNT, CHAR. COUNT
 7362  027104  005037  006460                  CLR     TTOTCC          ; AND RESET POINTER
 7363  027110  012737  006144  006436          MOV     #PTRTAB,TXPTR
 7364  027116  013737  006436  006520          MOV     TXPTR,CPTR
 7365  027124  012701  003144                  MOV     #TXBUF,R1
 7366  027130  010137  006462                  MOV     R1,TCURAD
 7367
 7368  027134  012702  001000          ACTCLB: MOV     #BUFLIM,R2
 7369  027140  010137  006522                  MOV     R1,CURADD       ;SET UP TO PUT DEFAULT MSG IN LIST AFTER 033'S
 7370  027144  012737  000005  006512          MOV     #5,MSGTYP
 7371  027152  013737  002162  006514          MOV     MSG5C,CURCC
 7372  027160  105021                  1$:     CLRB    (R1)+           ;FILL EXPT OR TRAN BUFFER WITH 0'S IF A CLEAR
 7373  027162  005302                          DEC     R2              ;DO 'BUFLIM' NUMBER OF BYTE LOCATIONS
 7374  027164  001375                          BNE     1$
 7375  027166  004737  022234                  JSR     PC,BLDBUF       ;''CLEAR'' REALLY MEANS TO PUT DEFAULT MSG IN
 7376  027172  000207                          RTS     PC              ;WHEN DONE, RETURN TO PARSER
 7377
 7378
 7379  027174  012705  003066          ACTSHW: MOV     #SHTAB,R5
 7380  027200  122571  000000          5$:     CMPB    (R5)+,@(R1)     ;LOOK AT FIRST BYTE OF MSG TO DECIPHER TYPE
 7381  027204  001404                          BEQ     6$
 7382  027206  020527  003075                  CMP     R5,#SHTEND      ;SEE IF LOOKED AT ALL OF DEFAULTS YET
 7383  027212  001372                          BNE     5$
 7384  027214  005205                          INC     R5              ;MUST BE OPR. SPEC'D THEN
 7385  027216  162705  003067          6$:     SUB     #SHTAB+1,R5
 7386  027222  006305                          ASL     R5
 7387  027224  016137  000002  006530          MOV     2(R1),TEMP
 7388  027232                                  PRINTF  #SHMSG,SHTYTB(R5),TEMP  ;PRINT MSG SIZE & TYPE
  (9)  027232  013746  006530                                                          MOV     TEMP,-(SP)
  (8)  027236  016546  003046                                                          MOV     SHTYTB(R5),-(SP)
  (7)  027242  012746  013232                                                          MOV     #SHMSG,-(SP)
  (6)  027246  012746  000003                                                          MOV     #3,-(SP)
  (3)  027252  010600                                                                  MOV     SP,R0
  (4)  027254  104417                                                                  TRAP    C$PNTF
  (4)  027256  062706  000010                                                          ADD     #10,SP
 7389  027262  062701  000004                  ADD     #4,R1           ;BUMP R1 TO NEXT SET OF POINTERS
 7390  027266  005302                          DEC     R2
 7391  027270  001341                          BNE     ACTSHW
 7392  027272  013737  006562  007676          MOV     MODTYP,DEV1
 7393  027300  013737  006564  007700          MOV     MLTYP,DEV2
 7394  027306  013737  006572  007702          MOV     RPASS,DEV3
 7395  027314  013737  006570  007704          MOV     PARAM,DEV4
 7396  027322  004737  022564                  JSR     PC,SHWOP        ;SHOW THE OPERATOR THE CURRENT MODE..... ALSO
 7397  027326  000207                          RTS     PC
 7398
 7399  027330  013737  003136  006504  ACTDMS: MOV     P$NUM,STADD     ;SETUP STARTING ADDRESS FOR DUMP
 7400  027336  005037  006510                  CLR     BYTBIT          ;SET DEFAULT OF WORD DUMP
 7401  027342  012737  000052  003010          MOV     #DMPS,KEYWD1    ;FLAG THAT A DUMP WAS TYPED
 7402  027350  000403                          BR      ACTDME
 7403
 7404  027352  012737  177777  006510  ACTDMQ: MOV     #-1,BYTBIT      ;SET DUMP FLAG TO 'DUMP-WORD'
 7405  027360  013737  003136  006506  ACTDME: MOV     P$NUM,ENADD     ;SETUP END ADDRESS FOR DUMP (=START IF NO 'EEE'
 7406  027366  105037  003142          ACTDMX: CLRB    P$NNUF          ;CLEAR NOT-ENOUGH FLAG, 'DUMP N-N/B' IS VALID
 7407  027372  000207                          RTS     PC
 7408
```

```
 7410
 7411
 7412  027374  012737  000010  003010  ACTSTE: MOV     #SETEXP,KEYWD1
 7413  027402  000403                          BR      ACTSTX
 7414
 7415  027404  012737  000011  003010  ACTSTT: MOV     #SETTRN,KEYWD1
 7416  027412  012737  000001  003014  ACTSTX: MOV     #1,QUALVL       ;SET UP DEFAULT COPY TO 1 (/COPY=0)
 7417  027420  000207                          RTS     PC
 7418
 7419  027422  012737  000012  003012  ACTSZE: MOV     #SIZE,QUALFG
 7420  027430  000207                          RTS     PC
 7421
 7422  027432  012737  000013  003012  ACTCOP: MOV     #QCOPY,QUALFG
 7423  027440  000207                          RTS     PC
 7424
 7425  027442  023727  003012  000012  ACTNUM: CMP     QUALFG,#SIZE    ;SEE IF A SIZE OR COPY TYPED
 7426  027450  001023                          BNE     1$              ;BR IF IT WAS A COPY
 7427  027452  005737  003136                  TST     P$NUM           ;CHECK TO BE SURE DIDN'T TRY SIZE=0
 7428  027456  001014                          BNE     3$              ; BR IF NO
 7429  027460                                  PRINTF  #CLISEO
   (7) 027460  012746  012042                                                  MOV     #CLISEO,-(SP)
   (6) 027464  012746  000001                                                  MOV     #1,-(SP)
   (3) 027470  010600                                                          MOV     SP,R0
   (4) 027472  104417                                                          TRAP    C$PNTF
   (4) 027474  062706  000004                                                  ADD     #4,SP
 7430  027500  112737  177777  003143          MOVB    #-1,P$GDBD      ;SEE ERROR-IN-CMD FLAG
 7431  027506  000411                          BR      2$
 7432  027510  013737  003136  006514  3$:     MOV     P$NUM,CURCC     ;IF A SIZE LOAD CURCC WITH BYTE COUNT
 7433  027516  000405                          BR      2$
 7434  027520  013737  003136  003014  1$:     MOV     P$NUM,QUALVL    ;IF A COPY, LOAD COPY COUNT
 7435  027526  005237  003014                  INC     QUALVL          ;INCREMENT SO FIRST DEC MAKES IT REAL #
 7436  027532  000522                  2$:     BR      ACTMEX
 7437
 7438  027534  012737  000007  006512  ACTOPM: MOV     #7,MSGTYP
 7439  027542  010437  006530                  MOV     R4,TEMP         ;KEEP TRACK OF START OF QUOTED TEXT
 7440  027546  005237  006530                  INC     TEMP            ; SO CAN CALC OPCNT AT END OF QUOTES
 7441  027552  000207                          RTS     PC
 7442
 7443  027554  010402                  ACTEQO: MOV     R4,R2
 7444  027556  163702  006530                  SUB     TEMP,R2
 7445  027562  010237  006514                  MOV     R2,CURCC        ;CALC BYTE COUNT FOR QUOTED TEXT
 7446  027566  010237  002166                  MOV     R2,OPCNT
 7447  027572  013701  006530                  MOV     TEMP,R1
 7448  027576  012705  002520                  MOV     #OPBUF,R5
 7449  027602  112125                  1$:     MOVB    (R1)+,(R5)+     ;COPY QUOTED TEXT TO OPBUF
 7450  027604  005302                          DEC     R2
 7451  027606  001375                          BNE     1$
 7452  027610  000473                          BR      ACTMEX
 7453
 7454  027612                          ACTBCR: PRINTF  #CLIBCR         ;BAD CHAR. IN OPR. QUOTED STRING
   (7) 027612  012746  011775                                                  MOV     #CLIBCR,-(SP)
   (6) 027616  012746  000001                                                  MOV     #1,-(SP)
   (3) 027622  010600                                                          MOV     SP,R0
   (4) 027624  104417                                                          TRAP    C$PNTF
   (4) 027626  062706  000004                                                  ADD     #4,SP
 7455  027632  000207                          RTS     PC
```

```
 7456
 7457  027634  005037  006512          ACTMS0: CLR     MSGTYP
 7458  027640  000435                          BR      ACTME1
 7459  027642  012737  000001  006512  ACTMS1: MOV     #1,MSGTYP          ;SET MESSAGE TYPE = ALL ONES
 7460  027650  000431                          BR      ACTME1
 7461  027652  012737  000002  006512  ACTMS2: MOV     #2,MSGTYP          ;SET MESSAGE TYPE = ONES & ZEROS
 7462  027660  000425                          BR      ACTME1
 7463  027662  012737  000003  006512  ACTMS3: MOV     #3,MSGTYP          ;SET MESSAGE TYPE = ZEROS & ONES
 7464  027670  000421                          BR      ACTME1
 7465  027672  012737  000004  006512  ACTMS4: MOV     #4,MSGTYP          ;SET MESSAGE TYPE = CCITT
 7466  027700  000415                          BR      ACTME1
 7467  027702  012737  000005  006512  ACTMS5: MOV     #5,MSGTYP          ;SET MESS TYPE = QUICK FOX
 7468  027710  013737  002162  006514          MOV     MSG5C,CURCC        ;SETUP DEFAULT SIZE FOR THIS TYPE
 7469  027716  000430                          BR      ACTMEX
 7470  027720  012737  000006  006512  ACTMS6: MOV     #6,MSGTYP          ;SET MESSAGE TYPE = ALPHA/NUM
 7471  027726  013737  002164  006514          MOV     MSG6C,CURCC        ;SETUP DEFAULT SIZE FOR THIS TYPE
 7472
 7473  027734  012737  000100  006514  ACTME1: MOV     #64.,CURCC         ;SETUP DEFAULT SIZE FOR MSG0-4
 7474  027742  000416                          BR      ACTMEX             ;BRANCH TO EXIT
 7475
 7476                                          ;REV B EC
 7477  027744  022737  000010  003010  ACTSEX: CMP     #SETEXP,KEYWD1     ;DID WE GET HERE FROM ''SET E=''COMMAND?
 7478  027752  001404                          BEQ     10$                ;YES,BRANCH
 7479  027754  112737  000001  003143          MOVB    #1,P$GDBD          ;SET ERROR FLAG
 7480  027762  000406                          BR      ACTMEX             ;GO EXIT SUBROUTINE
 7481  027764  004737  022360          10$:    JSR     PC,FACSIMILE       ;GO COPY TRANSMIT LIST TO EXPECT LIST
 7482  027770  012737  000060  003010          MOV     #SETET,KEYWD1      ;SET FLAG TO BE USED IN T1::
 7483  027776  000400                          BR      ACTMEX             ;EXIT SUBROUTINE
 7484
 7485  030000  105037  003142  ACTMEX: CLRB    P$NNUF             ;CLEAR NOT-ENOUGH FLAG
 7486  030004  000207                          RTS     PC
 7487
```

```
 7489  030006  012737  000003  006562  ACTATV: MOV     #ACT,MODTYP             ;MODE = ACTIVE
 7490  030014  000432                          BR      ACTM2X
 7491
 7492  030016  012737  000002  006562  ACTPAS: MOV     #PAS,MODTYP             ;MODE = PASSIVE
 7493  030024  105037  003142                  CLRB    P$NNUF                  ;CLEAR NOT-ENOUGH FLAG
 7494  030030  005037  006564                  CLR     MLTYP                   ;CLEAR MAINT LOOP TYPE
 7495  030034  000207                          RTS     PC
 7496
 7497  030036  005037  006562          ACTREC: CLR     MODTYP                  ;MODE = RECEIVE
 7498  030042  000417                          BR      ACTM2X
 7499
 7500  030044  012737  000006  006562  ACTLIS: MOV     #LIS,MODTYP             ;MODE = LISTEN
 7501  030052  000413                          BR      ACTM2X
 7502
 7503  030054  012737  000004  006562  ACTDLL: MOV     #DOW,MODTYP             ;MODE = DOWNLINE LOAD
 7504  030062  000407                          BR      ACTM2X
 7505
 7506  030064  012737  000001  006562  ACTTRA: MOV     #TRA,MODTYP             ;MODE = TRANSMIT
 7507  030072  000403                          BR      ACTM2X
 7508
 7509  030074  012737  000005  006562  ACTTAL: MOV     #TAL,MODTYP             ;MODE = TALK
 7510
 7511  030102  042737  000004  006570  ACTM2X: BIC     #ECHOB,PARAM            ;DISABLE /ECHO (ALL BUT PASSIVE MODE)
 7512  030110  105037  003142                  CLRB    P$NNUF                  ;CLEAR NOT-ENOUGH FLAG
 7513  030114  005037  006564                  CLR     MLTYP                   ;CLEAR MAINT LOOP TYPE
 7514  030120  000207                          RTS     PC
 7515
```

```
 7517  030122  012737  000036  003012  ACTNO:  MOV     #NO,QUALFG
 7518  030130  000207                  RTS     PC
 7519
 7520  030132  022737  000036  003012  ACTECH: CMP     #NO,QUALFG
 7521  030140  001422                  BEQ     1$
 7522  030142  052737  000004  006570          BIS     #ECHOB,PARAM
 7523  030150  022737  000002  006562          CMP     #PAS,MODTYP             ;BE SURE IN PASSIVE MODE IF
 7524  030156  001416                  BEQ     2$                              ;IF TRYING TO SET /ECHO
 7525  030160                          PRINTF  #CLINPS
   (7) 030160  012746  011732                                                          MOV     #CLINPS,-(SP)
   (6) 030164  012746  000001                                                          MOV     #1,-(SP)
   (3) 030170  010600                                                                  MOV     SP,R0
   (4) 030172  104417                                                                  TRAP    C$PNTF
   (4) 030174  062706  000004                                                          ADD     #4,SP
 7526  030200  112737  177777  003143          MOVB    #-1,P$GDBD
 7527  030206  042737  000004  006570  1$:     BIC     #ECHOB,PARAM
 7528  030214  005037  003012          2$:     CLR     QUALFG                  ;CLEAR 'NO' OUT OF QUALIFIER FLAG
 7529  030220  000501                  BR      ACTLXX
 7530
 7531  030222  012701  000002          ACTCHK: MOV     #DATCKB,R1              ;SET DATA CHECK BIT
 7532  030226  000413                  BR      ACTQFG
 7533
 7534  030230  012701  000001          ACTSTS: MOV     #STATB,R1       ;SET THE STATUS BIT
 7535  030234  000410                  BR      ACTQFG
 7536
 7537  030236  012701  000020          ACTCRC: MOV     #CRCB,R1        ;SET THE CRC BIT
 7538  030242  000405                  BR      ACTQFG
 7539
 7540  030244  012701  000010          ACTMOS: MOV     #MOCHK,R1       ;SET THE MODEM BIT
 7541  030250  000402                  BR      ACTQFG
 7542
 7543  030252  012701  000040          ACTPRO: MOV     #PROTOB,R1      ;SET THE PROTOCOL BIT
 7544
 7545  030256  050137  006570          ACTQFG: BIS     R1,PARAM
 7546  030262  022737  000036  003012          CMP     #NO,QUALFG
 7547  030270  001002                  BNE     1$
 7548  030272  040137  006570                  BIC     R1,PARAM
 7549  030276  005037  003012          1$:     CLR     QUALFG                  ;CLEAR 'NO' OUT OF QUALIFIER FLAG
 7550  030302  000450                  BR      ACTLXX
 7551
 7552  030304  013737  003136  006572  ACTRPS: MOV     P$NUM,RPASS             ;GET NUMBER OF 'RUN PASSES'
 7553  030312  000444                  BR      ACTLXX
 7554
 7555  030314  012737  000005  006564  ACTMOP: MOV     #5,MLTYP
 7556  030322  000417                  BR      ACTLPX
 7557  030324  012737  000001  006564  ACTTLP: MOV     #1,MLTYP
 7558  030332  000413                  BR      ACTLPX
 7559  030334  012737  000002  006564  ACTCLP: MOV     #2,MLTYP
 7560  030342  000407                  BR      ACTLPX
 7561  030344  012737  000003  006564  ACTLLP: MOV     #3,MLTYP
 7562  030352  000403                  BR      ACTLPX
 7563  030354  012737  000004  006564  ACTRLP: MOV     #4,MLTYP
 7564
 7565  030362  022737  000003  006562  ACTLPX: CMP     #ACT,MODTYP     ;BE SURE IN ACTIVE IF TRYING TO SET LOOP
 7566  030370  001415                  BEQ     ACTLXX          ; BR IF IN ACTIVE
 7567  030372  112737  177777  003143          MOVB    #-1,P$GDBD
```

```
 7568  030400  005037  006564                CLR     MLTYP           ;CLEAR ANY LOOP TYPE THAT MAY HAVE GOT SET
 7569  030404                                PRINTF  #CLIBDL
  (7)  030404  012746  011670                                                MOV     #CLIBDL,-(SP)
  (6)  030410  012746  000001                                                MOV     #1,-(SP)
  (3)  030414  010600                                                        MOV     SP,R0
  (4)  030416  104417                                                        TRAP    C$PNTF
  (4)  030420  062706  000004                                                ADD     #4,SP
 7570  030424  105037  003142  ACTLXX:       CLRB    P$NNUF          ;CLEAR NOT-ENOUGH FLAG
 7571  030430  000207                        RTS     PC
 7572
```

```
7574
7575                                        ;REV B BY EC
7576  030432  005737  006564          GTR9:  TST     MLTYP            ;LOOP MODE ?
7577  030436  001422                        BEQ     10$              ;NO,BRANCH
7578  030440  032737  000002  006570        BIT     #DATCKB,PARAM    ;DATA CHECK ?
7579  030446  001416                        BEQ     10$              ;NO,BRANCH
7580  030450  023737  006442  006456        CMP     CMPTOT,TXMTOT    ;TX = CX ?
7581  030456  001412                        BEQ     10$              ;YES,BRANCH
7582  030460                              PRINTF  #CLIPW           ;PRINT WARNING
 (7)  030460  012746  012073                                                    MOV     #CLIPW,-(SP)
 (6)  030464  012746  000001                                                    MOV     #1,-(SP)
 (3)  030470  010600                                                            MOV     SP,R0
 (4)  030472  104417                                                            TRAP    C$PNTF
 (4)  030474  062706  000004                                                    ADD     #4,SP
7583  030500  000137  025460                    JMP     GETCL            ;TRY AGAIN
7584
7585
7586                                        ; RX ALLOCATE CODE
7587  030504  012737  006144  006436  10$:   MOV     #PTRTAB,TXPTR    ;INIT TRANSMIT MESSAGE POINTER
7588  030512  012737  006240  006440        MOV     #PTR13,CMPPTR    ;INIT COMPARE MESSAGE POINTER
7589  030520  012737  006334  006434        MOV     #PTR23,RXPTR     ;INIT RECEIVE MESSAGE POINTER
7590
7591  030526  013737  006442  006472        MOV     CMPTOT,RXMTOT    ;MAKE COMPARE AND RX MESSAGE COUNTS EQUAL
7592
7593
7594  030534  005037  006574          GTREX: CLR     FLAG             ;CLEAR FLAG
7595  030540  005037  006476                 CLR     OPVAR            ;CLEAR OPTIONAL VARIABLE COUNTER
7596  030544  005037  006500                 CLR     PSCNT            ;CLEAR PASS COUNT
7597  030550  005037  006502                 CLR     ERRCNT           ;CLEAR ERROR COUNT
7598  030554  005037  011466                 CLR     MGLCNT           ;CLEAR GLITCH COUNT
7599  030560  005037  011470                 CLR     MHRCNT           ;CLEAR HARD ERR. COUNT
7600  030564  005037  006474                 CLR     LNCNT            ;CLEAR LINE COUNTER
7601  030570  012737  000626  011456        MOV     #626,SYNCW       ;SET UP SYNCW FOR 226 SYNC +TSOM
7602  030576  052737  000200  011440        BIS     #BIT7,DPVP1      ;SET UP PARAM WORD FOR 226 RX SYNC
7603  030604  005737  011472                 TST     RNODE
7604  030610  001406                        BEQ     1$               ;IF NON ITEP GO TO 1
7605  030612  042737  000200  011440        BIC     #BIT7,DPVP1      ;SET UP FOR 26 SYNC WORD ON RX.
7606  030620  012737  000426  011456        MOV     #426,SYNCW       ;ELSE SET UP SYNC FOR 26 AND TSOM
7607  030626  004737  017670          1$:    JSR     PC,LOGDVI        ;LOG ABOUT TO INIT DEVICE
7608  030632  004737  032672                 JSR     PC,DVINIT        ;INIT DEVICE
7609
7610  030636  012737  001000  006514  GTRX2: MOV     #BUFLIM,CURCC    ;SET CHAR COUNT TO 'BUFLIM'' NO. OF BYTES
7611  030644  012737  004144  006522        MOV     #RXBUF,CURADD    ;SET UP RX BUFFER AS CURRRENT ADD.
7612  030652  013737  006434  006520        MOV     RXPTR,CPTR
7613  030660  012737  000010  006512        MOV     #10,MSGTYP       ;SET UP FOR 33 TO FILL RX BUFFERS
7614  030666  004737  022234                 JSR     PC,BLDBUF        ;CLEAR RX BUFFER
7615  030672  013702  006562                 MOV     MODTYP,R2
7616  030676  006302                        ASL     R2
7617  030700  000172  006576                 JMP     @MODE(R2)        ;MODE DISPATCH
7618
```

```
                                .SBTTL          RECEIVE MODE SECTION
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       RECEIVE-ONLY (OR ONE-WAY-IN) ROUTINE
                                ;       IN THIS MODE OF TESTING THE DEVICE'S RECEIVER IS ENABLED IN EXPECTATION
                                ;       OF RECEIVING A MESSAGE.  AFTER RECEIVING AN "EXPECTED" NUMBER OF
                                ;       MESSAGES, THE DATA RECEIVED CAN BE COMPARED AGAINST A LIST OF "EXPECT
                                ;       TO RECEIVE" MESSAGES IF DATA-CHECKING IS ENABLED.
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;               "ALLTR"
                                ;
                                ; CALLING SEQUENCE:
                                ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
                                ;--

030704                          RXONLY:
030704  013737  006434  006516  RXON2:  MOV     RXPTR,CPTRR
030712  013737  006472  006470          MOV     RXMTOT,DVRCT    ;SET UP MESSAGE COUNT
030720  052737  000104  006574          BIS     #QRX+#ERX,FLAG  ;SET UP RX QUE
030726  005037  006520                  CLR     CPTR            ;CLEAR THE TX POINTER
030732  000137  031074                  JMP     ALLTR           ;GO RX.
```

```
                                .SBTTL          TRANSMIT MODE SECTION

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       TRANSMIT-ONLY (OR ONE-WAY-OUT) ROUTINE
                                ;       IN THIS MODE OF TESTING A LIST OF MESSAGES IS TRANSMITTED WITHOUT
                                ;       EXPECTING ANY DATA TO BE RECEIVED.  A REPETITION COUNT CAN BE
                                ;       SPECIFIED TO REPETITIVELY TRANSMIT THE LIST.
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;               "ALLTR"
                                ;
                                ; CALLING SEQUENCE:
                                ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
                                ;--

030736  042737  000002  006570  TXONLY: BIC     #DATCKB,PARAM   ;SET NOCHECK
030744  013737  006436  006520  TXON2:  MOV     TXPTR,CPTR
030752  013737  006456  006454          MOV     TXMTOT,DVTCT    ;COPY COUNTER FOR THIS PASS
030760  052737  000210  006574          BIS     #QTX+#ETX,FLAG  ;SET THE QUE TX FLAG
030766  005037  006516                  CLR     CPTRR           ;CLEAR RX POINTER
030772  000137  031074                  JMP     ALLTR           ;GO TX.
```

```
                                        .SBTTL          PASSIVE MODE SECTION

                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       PASSIVE MODE SECTION
                                        ;       IN THIS MODE OF TESTING, THE DEVICE'S RECEIVER IS ENABLED IN
                                        ;       EXPECTATION OF RECEIVING A MESSAGE.  THEN EVERY TIME A MESSAGE IS
                                        ;       RECEIVED, A MESSAGE IS TRANSMITTED.  DATA CHECKING CAN BE DONE ON THE
                                        ;       RECEIVED DATA.
                                        ;
                                        ; SUBORDINATE ROUTINES USED:
                                        ;
                                        ;               "ALLTR"
                                        ;
                                        ; CALLING SEQUENCE:
                                        ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
                                        ;--

030776                                  PLCK:
030776  013737  006456  006454         PLCK2:  MOV     TXMTOT,DVTCT    ;SET UP  THE TRANSMIT COUNT
031004  013737  006436  006520                 MOV     TXPTR,CPTR      ;SET UP CPTR TO TRANSMIT POINTER
031012  013737  006434  006516         PLCK3:  MOV     RXPTR,CPTRR     ;SET UP CPTRR TO REC POINTER
031020  052737  000104  006574                 BIS     #QRX+#ERX,FLAG  ;SET UP Q AND EXPECT RX
031026  000137  031074                         JMP     ALLTR           ;AND GO RX FIRST MSG.
```

```
                                .SBTTL          ACTIVE MODE SECTION

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       ACTIVE MODE SECTION
                                ;       IN THIS MODE OF TESTING A LIST OF MESSAGES IS TRANSMITTED AND
                                ;       MESSAGES ARE EXPECTED TO BE RECEIVED.  RECEIVED DATA CAN BE COMPARED
                                ;       AGAINST "EXPECTED" DATA IF DATA-CHECKING IS ENABLED.
                                ;               NOTE: IF BOTH ENDS OF THE LINK ARE IN ACTIVE MODE, THEN THE
                                ;                       LINK MUST BE A FULL DUPLEX LINK!
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;               "ALLTR"
                                ;
                                ; CALLING SEQUENCE:
                                ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
                                ;--

031032  013737  006456  006454  ALCK:   MOV     TXMTOT,DVTCT
031040  013737  006436  006520          MOV     TXPTR,CPTR      ;SET UP TX COUNTS
031046  013737  006472  006470          MOV     RXMTOT,DVRCT    ;SET UP COUNTS
031054  013737  006434  006516          MOV     RXPTR,CPTRR
031062  052737  000314  006574          BIS     #QRX+#QTX+#ETX+#ERX,FLAG
031070  000137  031074                  JMP     ALLTR
```

```
                                .SBTTL          TRANSMIT - RECEIVE FOR ALL STANDARD MODES

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       THIS CODE PERFORMS THE FOLLOWING FUNCTIONS
                                ;       1.) IF RX BUFFERS ARE TO BE QUED,TELL DEVICE
                                ;           CODE TO QUE THEM ,LOG RECEIVE QUED.
                                ;       2.) IF TX BUFFERS ARE TO BE QUED ,TELL DEVICE
                                ;           CODE TO QUE THEM, LOG TRANSMIT QUED.
                                ;       3.) WAIT FOR EITHER RECIVE BUFFER OR TRANSMIT BUFFER OR
                                ;           BOTH TO COMPLETE
                                ;       4.) IF RECEIVE COMPLETE LOG IT UPDATE RX TABLE IF DATA
                                ;           CHECKING.
                                ;       5.) IF TRANSMIT COMPLETE LOG IT.
                                ;       6.) WHEN BOTH TRANSMIT AND RECIEVE LISTS ARE DONE
                                ;           GO TO THE COMPARE BUFFER CODE
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;               'DVRXQ' -QUE RECEIVE BUFFER SPACE TO DEVICE
                                ;               'LOGRXQ'-LOG RECEIVE BUFFER SPACE TO EVENT LOG
                                ;               'LOGTXQ'-LOG TRANSMIT BUFFER QUED TO EVENT LOG
                                ;               'DVTXRX'-QUE TRANSMIT BUFFER AND WAIT FOR RX
                                ;                       OR TX TO COMPLETE
                                ;               'LOGRXC'-LOG RECEIVE BUFFER COMPLETED TO EVENT LOG
                                ;               'LOGTXC'-LOG TRANSMIT BUFFER COMPLETED TO EVENT LOG
                                ;
                                ; USE OF FLAG BITS:
                                ;               QRX - SET ON INPUT TO ALLTR IF REC IS TO BE QUED TO
                                ;                     DEVICE. CLEARED BY DVRXQ AND THEN SET BY DVTXRX
                                ;                     WHEN RX BUFFER IS COMPLETED.
                                ;               QTX - SET ON INPUT TO ALLTR IF TRANSMIT IS TO BE QUED TO
                                ;                     DEVICE. CLEARED ON ENTRY TO DVTXRX AND SET BY DVTXRX
                                ;                     WHEN TX BUFFER IS COMPLETED.
                                ;               ETX - USED BY DVTXRX TO DETERMINE IF TX BUFFER COMPLETED IS
                                ;                     EXPECTED.
                                ;               ERX - USED BY DVTXRX TO DETERMINE IF RX BUFFER COMPLETED IS
                                ;                     EXPECTED.
                                ;
                                ; CALLING SEQUENCE:
                                ;       JMP     ALLTR           ;GO TO TRANSMIT-RECEIVE FOR ALL STANDARD MODES
                                ;--

031074                          ALLTR:
031074  032737  000004  006574  ALCK5:  BIT     #QRX,FLAG
031102  001420                  BEQ     ALCK1           ;IF NOT RX GO TO TX'S
031104  013702  006516          MOV     CPTRR,R2
031110  011237  006534          MOV     (R2),TEMP2
031114  012237  006464          MOV     (R2)+,DVRXA
031120  011237  006536          MOV     (R2),TEMP3
031124  011237  006466          MOV     (R2),DVRCC
031130  010237  006516          MOV     R2,CPTRR
031134  004737  033544          JSR     PC,DVRXQ        ;GO QUE DEVICE
031140  004737  017624          JSR     PC,LOGRXQ       ;LOG REC QUED
031144  032737  000010  006574  ALCK1:  BIT     #QTX,FLAG
```

```
7778  031152  001416                          BEQ     ALCK2           ;IF NO TX'S GO TO 2
7779  031154  013702  006520                  MOV     CPTR,R2
7780  031160  011237  006534                  MOV     (R2),TEMP2
7781  031164  012237  006450                  MOV     (R2)+,DVTXA
7782  031170  011237  006536                  MOV     (R2),TEMP3
7783  031174  012237  006452                  MOV     (R2)+,DVTCC
7784  031200  010237  006520                  MOV     R2,CPTR
7785  031204  004737  017570                  JSR     PC,LOGTXQ
7786
7787  031210  004737  033646          ALCK2:  JSR     PC,DVTXRX       ;GO TO TX AND RX SUB ROUT.
7788
7789  031214  032737  000004  006574          BIT     #QRX,FLAG       ;CHECK FOR REC. MSG.
7790  031222  001514                          BEQ     ALCK3
7791  031224  013737  006464  006534          MOV     DVRXA,TEMP2
7792  031232  013737  006466  006536          MOV     DVRCC,TEMP3
7793  031240  004737  017642                  JSR     PC,LOGRXC       ;LOG REC COMPLETE
7794  031244  032737  000004  006570  UPTABL: BIT     #ECHOB,PARAM    ;IS THIS ECHO MODE(PASSIVE)
7795  031252  001406                          BEQ     UPTA4           ;IF NOT GO TO 4
7796  031254  013702  006520                  MOV     CPTR,R2         ;ELSE SET R2 TO PRESENT TX TABL
7797  031260  013722  006534                  MOV     TEMP2,(R2)+     ;STORE OFF RX ADD
7798  031264  013712  006536                  MOV     TEMP3,(R2)      ;AND CC
7799  031270  032737  000002  006570  UPTA4:  BIT     #DATCKB,PARAM   ;IS DATA CHECKING ASKED FOR
7800  031276  001015                          BNE     UPTA1           ;IF SO GO TO 1
7801  031300  012737  000001  006470          MOV     #01,DVRCT       ;ELSE SET DVRCT TO A 1
7802  031306  013737  006434  006516          MOV     RXPTR,CPTRR     ;RESET POINTER
7803  031314  022737  000003  006562          CMP     #ACT,MODTYP     ;IS THIS ACTIVE
7804  031322  001002                          BNE     UPTA3
7805  031324  005237  006470                  INC     DVRCT           ;IF YES BUMP COUNT
7806  031330  000424                  UPTA3:  BR      UPTEX
7807  031332  013702  006516          UPTA1:  MOV     CPTRR,R2
7808  031336  011237  006530                  MOV     (R2),TEMP       ;LOAD TEMP WITH PREV. COUNT
7809  031342  163737  006536  006530          SUB     TEMP3,TEMP      ;LOAD TEMP WITH PREV.COUNT-CURRENT
7810  031350  013722  006536                  MOV     TEMP3,(R2)+
7811  031354  063737  006536  006534          ADD     TEMP3,TEMP2
7812  031362  013722  006534                  MOV     TEMP2,(R2)+     ;STORE OF NEW ADD
7813  031366  013712  006530                  MOV     TEMP,(R2)       ;AND NEW CC
7814  031372  162702  000002                  SUB     #2,R2           ;PUT POINTER BACK TO ADDR.
7815
7816  031376  010237  006516                  MOV     R2,CPTRR        ;AND RESTORE IT.
7817
7818  031402                          UPTEX:
7819  031402  022737  000002  006562          CMP     #PAS,MODTYP
7820  031410  001007                          BNE     ALCK2A          ;IF NOT PASSIVE LOOP THEN GO TO 2A
7821  031412  042737  000104  006574          BIC     #QRX+#ERX,FLAG  ;CLEAR BOTH EXPECTED AND COMPLETED FLAGS
7822  031420  052737  000210  006574          BIS     #QTX+#ETX,FLAG  ;SET THE TX FLAGS
7823  031426  000646                          BR      ALCK1
7824
7825  031430  005337  006470          ALCK2A: DEC     DVRCT           ;DEC REC COUNT
7826  031434  005737  006470                  TST     DVRCT           ;IS IT ALL DONE
7827  031440  001005                          BNE     ALCK3           ;NO. GO CHECK TX
7828  031442  042737  000004  006574          BIC     #QRX,FLAG       ;CLEAR THE RX FLAG
7829  031450  005037  006516                  CLR     CPTRR           ;YES. CLEAR POINTER
7830  031454  032737  000010  006574  ALCK3:  BIT     #QTX,FLAG       ;IS IT TX
7831  031462  001447                          BEQ     ALCK4           ;IF NOT TX THEN GO BACK
7832  031464  013737  006450  006534          MOV     DVTXA,TEMP2
7833  031472  013737  006452  006536          MOV     DVTCC,TEMP3     ;LOG TX COMPLETED
```

```
7834  031500  004737  017606                      JSR     PC,LOGTXC
7835  031504  005337  006454                      DEC     DVTCT           ;DEC TX COUNT
7836  031510  022737  000002  006562              CMP     #PAS,MODTYP
7837  031516  001013                              BNE     ALCK3A          ;IF NOT PASSIVE MODE GO TO 3A
7838  031520  042737  000210  006574              BIC     #QTX+ETX,FLAG   ;CLEAR THE TX FLAGS
7839  031526  052737  000104  006574              BIS     #QRX+ERX,FLAG   ;AND SET THE RX FLAGS
7840  031534  005737  006454                      TST     DVTCT
7841  031540  001005                              BNE     ALCK3C          ;IF MORE RX'S DO IT
7842  031542  000137  031622                      JMP     CMPSR           ; ELSE COMPARE
7843  031546  005737  006454          ALCK3A:     TST     DVTCT           ;IS IT ALL DONE
7844  031552  001402                              BEQ     ALCK3B          ;IF NOT GO BACK TO 5
7845  031554  000137  031074          ALCK3C:     JMP     ALCK5
7846  031560  005037  006520          ALCK3B:     CLR     CPTR            ;IF SO CLEAR POINTER
7847  031564  042737  000010  006574              BIC     #QTX,FLAG       ;CLEAR TX FLAG
7848  031572  032737  000002  006570              BIT     #DATCKB,PARAM   ;IS IT DAT CK
7849  031600  001403                              BEQ     ALCK4A          ;IF NOT THEN END WO CKING RX.
7850  031602  005737  006516          ALCK4:      TST     CPTRR
7851
7852  031606  001362                              BNE     ALCK3C          ;IF SOME RX'S LEFT GO BACK
7853  031610  005737  006520          ALCK4A:     TST     CPTR
7854  031614  001402                              BEQ     ALCK4B          ;BRANCH IF ANY TX'S LEFT
7855  031616  000137  031210                      JMP     ALCK2
7856  031622                          ALCK4B:
7857
7858
7859
```

```
                                .SBTTL          DATA COMPARISON CODE


                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       CMPSR - COMPARE CODE
                                ;       THIS CODE COMPARES THE RECEIVED DATA AGAINST THE
                                ;       EXPECTED AND FILLS THE EVENT LOG WITH 1 OF 3 MSGS.
                                ;
                                ;       NOTE: IF NO DATA CHECKING SKIP THIS CODE
                                ;
                                ;       1.) A DATA COMPARISON ENTRY WHICH REPORTS THE NUMBER
                                ;           OF COMPARISON ERRORS FOUND.
                                ;       2.) A DATA COMPARISON ENTRY WHICH REPORTS DIFFERENCES
                                ;           IN REC LENGTH TO COMPARE LENGTH.
                                ;       3.) A DATA COMPARISON STARTED ENTRY WHICH REPORTS ADDRESS
                                ;           OF RECEIVE BUFFER AND BYTE COUNT.
                                ;       THIS CODE ALSO REPORTS SOFT ERRORS FOR DATA COMPARISON
                                ;       (THE FIRST 5 ONLY),LENGTH ERROR,AND TOTAL NUMBER OF ERRORS
                                ;
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;               'LOGCMP' - SEE ITEM 3 ABOVE
                                ;               'LOGCML' - SEE ITEM 2 ABOVE
                                ;               'LOGCMD' - SEE ITEM 1 ABOVE
                                ;
                                ; CALLING SEQUENCE:
                                ;       JMP     CMPSR           ;JUMP TO DATA COMPARISON CODE
                                ;--

031622  032737  000002  006570  CMPSR:  BIT     #DATCKB,PARAM   ;IS DATA CHECKING TO BE DONE
031630  001522                  BEQ     CMPSEX          ;IF NOT THEN EXIT
031632  013737  006434  006520          MOV     RXPTR,CPTR      ;PUT START OF RX POINTERS TO CPTR
031640  013737  006440  006516          MOV     CMPPTR,CPTRR    ; AND START OF COMPARE POINTS TO CPTRR
031646  013737  006472  006470          MOV     RXMTOT,DVRCT

031654                          CMPS3:
031654  013702  006520                  MOV     CPTR,R2         ;MOVE CURRET RX PT.TO R2
031660  011237  006534                  MOV     (R2),TEMP2      ;MOVE RX ADD TO EVENT LOG
031664  012201                          MOV     (R2)+,R1        ;SET R1 TO START ADD OF RX
031666  012237  006536                  MOV     (R2)+,TEMP3     ;SET CHAR  COUNT TO EVENT LOG
031672  010237  006520                  MOV     R2,CPTR         ;RESTORE RX POINT

031676  013702  006516                  MOV     CPTRR,R2        ;PUT R2 AT COMPARE TABLE
031702  012203                          MOV     (R2)+,R3        ;SET R3 TO COMPARE ADD
031704  012204                          MOV     (R2)+,R4        ;SET R4 TO COMP CC
031706  010237  006516                  MOV     R2,CPTRR        ;RESTORE POINTER
031712  010437  006540                  MOV     R4,TEMP4
031716  004737  017736                  JSR     PC,LOGCMP       ;LOG COMPARE START.

031722  020437  006536                  CMP     R4,TEMP3        ;IS COMPARE COUNT = TO RX COUNT
031726  001410                          BEQ     CMPS7           ;IF SO GO TO 7
031730  005237  006502                  INC     ERRCNT
031734                                  ERRSOFT 1,EDDLE,ERR10   ;PRINT ERROR
```

```
                                                                                    TRAP    C$ERSOFT
                                                                                    .WORD   1
                                                                                    .WORD   EDDLE
                                                                                    .WORD   ERR10
7917  031744  004737  017754          JSR     PC,LOGCML       ;LOG LENGTH ERROR
7918
7919  031750  005037  006540  CMPS7:  CLR     TEMP4           ;CLEAR BAD BYTE COUNTER
7920  031754  012737  000001  006526  MOV     #1,OFSET        ;SET OFSET BYTE COUNT TO 1
7921  031762  122123          CMPS1:  CMPB    (R1)+,(R3)+     ;COMPARE RX WITH EXPETED
7922  031764  001422                  BEQ     CMPS6           ;IF EQUAL THEN GO TO 6
7923
7924  031766  005237  006540  CMPS2:  INC     TEMP4           ;INC BAD COUNT
7925  031772  023727  006540  000005  CMP     TEMP4,#5        ;IS IT MORE THEN 5
7926  032000  101014                  BHI     CMPS6           ;IF SO GO  FOR MORE
7927  032002  114337  006550          MOVB    -(R3),GOOD      ;STORE GOOD BYTE FOR ERROR
7928  032006  114137  006551          MOVB    -(R1),BAD       ;STORE BAD BYTE FOR ERROR
7929  032012  005237  006502          INC     ERRCNT
7930  032016                          ERRSOFT 2,EDDDE,ERR1    ;REPORT COMPARISON FAILURE TO OPR.
                                                                                    TRAP    C$ERSOFT
                                                                                    .WORD   2
                                                                                    .WORD   EDDDE
                                                                                    .WORD   ERR1
7931  032026  005201                  INC     R1
7932  032030  005203                  INC     R3
7933  032032  005237  006526  CMPS6:  INC     OFSET           ;INC OFFSET
7934  032036  005304                  DEC     R4              ;ELSE DEC CHAR COUNT AND SEE IF 0
7935  032040  001350                  BNE     CMPS1           ;IF NOT GO BACK
7936  032042  005737  006540          TST     TEMP4           ;SEE IF ANY CMP ERRS FOR THIS MSG
7937  032046  001410                  BEQ     CMPS5A          ;BR IF NONE
7938  032050  005237  006502          INC     ERRCNT
7939  032054                          ERRSOFT 3,EDDDE,ERR2    ;REPORT # OF MISMATCHES FOR MESSAGE
                                                                                    TRAP    C$ERSOFT
                                                                                    .WORD   3
                                                                                    .WORD   EDDDE
                                                                                    .WORD   ERR2
7940  032064  004737  017772  CMPS5:  JSR     PC,LOGCMD       ;LOG DATA ERROR IN COMPARE
7941  032070                  CMPS5A:
7942  032070  005337  006470          DEC     DVRCT
7943  032074  001267                  BNE     CMPS3           ;IF NOT ALL DONE GO BACK
7944
```

```
                                .SBTTL          MODEM CHANGE REPORTS

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       THIS SECTION REPORTS THE NUMBER OF MOMDEM STATUS CHANGES
                                ;       THAT OCCUR ON EACH PASS. THE ERROR IS ONLY REPORTED IF
                                ;       THERE WERE ANY CHANGES IN OTHER WORDS A COUNT OF ZERO IS
                                ;       NOT REPORTED. THE CHANGES ARE REPORTED IN TWO CLASSES ..
                                ;       HARD ERRORS AND GLITCHES. HARD ERRORS ARE WHEN THE DEVICE
                                ;       IS ABLE TO LATCH UP THE BAD MODEM STATUS. GLITCHES OCCUR
                                ;       WHEN THE MODEM STATUS CHANGES TO CAUSE A DATA SET CHANGE
                                ;       INTERRUPT BUT THE CHANGE DOES NOT OCCUR LONG ENOUGH FOR
                                ;       THE DEVICE TO LATCH THE DATA
                                ;
                                ; INPUTS:
                                ;       'MGLCNT'        - CONTAINS NUMBER OF GLITCH ERRORS
                                ;       'MHRCNT'        - CONTAINS NUMBER OF HARD ERRORS
                                ;
                                ; OUTPUTS:
                                ;       'MGLCNT'        -ZEROED BY THIS SECTION
                                ;       'MHRCNT'        -ZEROED BY THIS SECTION
                                ;
                                ;;
                                ;;--

032076  005737  011466  CMPSEX: TST     MGLCNT          ;CHECK FOR ANY GLITCH ERRORS
032102  001003                  BNE     MCREP           ;IF NON ZERO REPORT THEM
032104  005737  011470          TST     MHRCNT          ;CHECK FOR ANY HARD ERRORS
032110  001412                  BEQ     ENDPS           ;IF NONE GO TO END OF PASS
                                ;
                                ;REPORT ANY MODEM ERRORS HERE
                                ;
032112  005237  006502  MCREP:  INC     ERRCNT          ;BUMP ERROR COUNT
032116                  ERRSOFT 4,MSCMS,ERR4
 (4)    032116  104457                                          TRAP    C$ERSOFT
 (5)    032120  000004                                          .WORD   4
 (5)    032122  014461                                          .WORD   MSCMS
 (5)    032124  017274                                          .WORD   ERR4
032126  005037  011466          CLR     MGLCNT          ;CLEAR GLITCH COUNT
032132  005037  011470          CLR     MHRCNT          ;CLEAR THE HARD COUNT
```

```
                                        .SBTTL          INTERNAL END OF PASS CODE

                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       THIS CODE INCREMENTS THE PASS COUNT FOR THE
                                        ;       EVENT LOG. LOGS THE END OF PASS EVENT
                                        ;       IF 'RPASS'' IS A MINUS ONE RETURN TO MODE
                                        ;       DISPATCHER. IF NOT -1 THEN DECREMENT RPASS
                                        ;       AND IF 'RPASS'' IS THEN = TO 0 GO TO DCLT PROMT
                                        ;       IN NOT = TO 0 THEN GO BACK TO MODE DISPATCHER
                                        ;
                                        ; SUBORDINATE ROUTINES USED:
                                        ;
                                        ;               'LOGEOP'' - LOG END OF PASS TO EVENT LOG
                                        ;------

032136  005237  006500          ENDPS:  INC     PSCNT           ;BUMP PASS COUNT

032142  013737  006476  006540          MOV     OPVAR,TEMP4
032150  013737  006500  006534          MOV     PSCNT,TEMP2
032156  013737  006502  006536          MOV     ERRCNT,TEMP3
032164  004737  020010                  JSR     PC,LOGEOP       ;LOG END OF PASS

032170  022737  177777  006572          CMP     #-1,RPASS       ;SEE IF RPASS=-1
032176  001403                          BEQ     1$              ;IF IT IS DON'T DECRMNT, LOOP FOREVER
032200  005337  006572                  DEC     RPASS           ;DEC PASS COUNT
032204  001402                          BEQ     2$              ;IF DONE EXIT TEST
032206  000137  030636          1$:     JMP     GTRX2           ;ELSE GO BACK AND DISPATCH
032212  000137  025374          2$:     JMP     GTRA5           ;WHEN RPASS=0 GO BACK TO 'DCLT>''
```

```
 8016                                   .SBTTL          DOWN-LINE-LOAD SECTION
 8017                                   ;++
 8018                                   ; FUNCTIONAL DESCRIPTION:
 8019                                   ;       DOWN LINE LOAD IS NOT SUPPORTED BY THIS DEVICE..
 8020                                   ;       IF THIS MODE IS CALLED BY THE COMMAND LINE INTERPRETER
 8021                                   ;       THEN A MESSAGE WILL BE PRINTED ....THAT SAYS DOWN LINE
 8022                                   ;       LOAD IS NOT!! SUPPORTED BY THIS DEVICE.
 8023                                   ;--
 8024
 8025
 8035
 8036  032216                           DLL:
 8037  032216                                   PRINTF  #DLLCM
  (7)  032216  012746  013632                                                     MOV     #DLLCM,-(SP)
  (6)  032222  012746  000001                                                     MOV     #1,-(SP)
  (3)  032226  010600                                                             MOV     SP,R0
  (4)  032230  104417                                                             TRAP    C$PNTF
  (4)  032232  062706  000004                                                     ADD     #4,SP
 8038  032236  000137  025374                   JMP     GTRA5
 8039
```

```
 8041                                   .SBTTL          TALK MODE SECTION
 8042
 8043                                   ;++
 8044                                   ; FUNCTIONAL DESCRIPTION:
 8045                                   ;       TALK MODE SECTION
 8046                                   ;       IN THIS MODE, THE ''TALK'' END OF THE LINK TRANSMITS OPERATOR
 8047                                   ;       SPECIFIED MESSAGES UNTIL A ''EXIT'' MESSAGE IS TYPE.  AT THAT POINT,
 8048                                   ;       THIS END OF THE LINK GOES INTO ''LISTEN'' MODE.
 8049                                   ;
 8050                                   ; SUBORDINATE ROUTINES USED:
 8051                                   ;
 8052                                   ;               ''LOGTXQ'' - LOG TX BUFFER QUED TO EVENT LOG
 8053                                   ;               ''DVTXRX'' - QUE TX BUFFER TO DEVICE AND WAIT FOR COMPLETE
 8054                                   ;               ''LOGTXC'' - LOG TX COMPLETE TO EVENT LOG
 8055                                   ;
 8056                                   ; CALLING SEQUENCE:
 8057                                   ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
 8058                                   ;--
 8059
 8060  032242                           TALCK:
 8061  032242  042737  000002  006570          BIC     #DATCKB,PARAM   ;SET NOCHECK
 8062  032250  012702  002520                  MOV     #OPBUF,R2
 8063  032254  012722  177777          1$:     MOV     #-1,(R2)+       ;CLEAR OUT OPBUFFER FIRST
 8064  032260  022702  002642                  CMP     #OPEND,R2
 8065  032264  001373                          BNE     1$
 8066  032266                                  GMANID  OPRMM,OPBUF,A,0,1,72.,NO        ;GET TALK MESSAGE
   (3) 032266  104443                                                                  TRAP    C$GMAN
   (3) 032270  000406                                                                  BR      10001$
   (4) 032272  002520                                                                  .WORD   OPBUF
   (5) 032274  000142                                                                  .WORD   T$CODE
   (5) 032276  013566                                                                  .WORD   OPRMM
   (5) 032300  000000                                                                  .WORD   0
   (5) 032302  000001                                                                  .WORD   T$LOLIM
   (5) 032304  000110                                                                  .WORD   T$HILIM
   (3) 032306                                                                  10001$:
 8067  032306  005002                          CLR     R2              ;NOW GET CHAR COUNT
 8068  032310  122762  000377  002520  2$:     CMPB    #377,OPBUF(R2)
 8069  032316  001402                          BEQ     3$
 8070  032320  005202                          INC     R2
 8071  032322  000772                          BR      2$
 8072  032324  010237  002166          3$:     MOV     R2,OPCNT
 8073
 8074  032330  012737  002520  006450          MOV     #OPBUF,DVTXA    ;SET UP TX ADDR.
 8075  032336  012737  002520  006534          MOV     #OPBUF,TEMP2
 8076  032344  013737  002166  006536          MOV     OPCNT,TEMP3
 8077  032352  013737  002166  006452          MOV     OPCNT,DVTCC     ;SET UP TX CC
 8078  032360  004737  017570                  JSR     PC,LOGTXQ
 8079  032364  052737  000210  006574          BIS     #QTX+#ETX,FLAG  ;SET UP FLAGS
 8080  032372  005037  006516                  CLR     CPTRR           ;CLEAR RX POINTER
 8081
 8082  032376  004737  033646                  JSR     PC,DVTXRX
 8083
 8084  032402  013737  006450  006534          MOV     DVTXA,TEMP2
 8085  032410  013737  006452  006536          MOV     DVTCC,TEMP3
 8086  032416  004737  017606                  JSR     PC,LOGTXC
 8087  032422  022737  054105  002520          CMP     #'EX,OPBUF      ;CHECK FOR EXIT
```

```
8088  032430  001304                    BNE     TALCK
8089  032432  022737  052111  002522    CMP     #''IT,OPBUF+2
8090  032440  001300                    BNE     TALCK
8091  032442  042737  000210  006574    BIC     #QTX+#ETX,FLAG  ;CLEAR THE TX BITS
8092  032450  012737  000006  006562    MOV     #LIS,MODTYP     ;CHANGE TO LISTEN MODE
8093  032456  000137  030636            JMP     GTRX2           ;AND GO BACK TO DISPATCH
```

```
 8095                                   .SBTTL          LISTEN MODE SECTION
 8096
 8097                                   ;++
 8098                                   ; FUNCTIONAL DESCRIPTION:
 8099                                   ;       LISTEN MODE SECTION
 8100                                   ;       IN THIS MODE, THE ''LISTEN'' END OF THE LINK PRINTS ALL OF THE MESSAGES
 8101                                   ;       RECEIVED BY THE DEVICE ON THE OPERATOR'S CONSOLE.  IF THE MESSAGE
 8102                                   ;       RECEIVED IS AN ''EXIT'' MESSAGE, THEN THE NODE ENTERS ''TALK'' MODE.
 8103                                   ;
 8104                                   ; SUBORDINATE ROUTINES USED:
 8105                                   ;
 8106                                   ;               ''DVRXQ'' - QUE RECEIVE BUFFER SPACE TO DEVICE
 8107                                   ;              ''LOGRXQ'' - LOG RECEIVE BUFFER QUED TO EVENT LOG
 8108                                   ;              ''DVTXRX'' - WAIT FOR RX TO COMPLETE
 8109                                   ;              ''LOGRXC'' - LOG RX COMPLETE TO EVENT LOG
 8110                                   ;
 8111                                   ; CALLING SEQUENCE:
 8112                                   ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
 8113                                   ;--
 8114
 8115  032462  042737  000002  006570  LISCK:  BIC     #DATCKB,PARAM   ;CLEAR CHECK BIT
 8116  032470                          PRINTF  #LISP           ;PRINT PROMPT FOR OPR.
   (7) 032470  012746  013555                                                      MOV     #LISP,-(SP)
   (6) 032474  012746  000001                                                      MOV     #1,-(SP)
   (3) 032500  010600                                                              MOV     SP,R0
   (4) 032502  104417                                                              TRAP    C$PNTF
   (4) 032504  062706  000004                                                      ADD     #4,SP
 8117  032510  012737  002520  006464  LISCKA: MOV     #OPBUF,DVRXA    ;SET DEVICE UP TO REC AT OPBUF
 8118  032516  012737  002520  006534          MOV     #OPBUF,TEMP2
 8119  032524  012737  000122  006466          MOV     #82.,DVRCC      ;SET UP CHAR COUNT TO 82.
 8120  032532  012737  000122  006536          MOV     #82.,TEMP3
 8121  032540  052737  000104  006574          BIS     #QRX+#ERX,FLAG  ;SET UP FLAG
 8122  032546  005037  006520                  CLR     CPTR            ;CLEAR THE TX.
 8123
 8124  032552  004737  033544                  JSR     PC,DVRXQ        ;QUE RX
 8125  032556  004737  017624                  JSR     PC,LOGRXQ
 8126
 8127  032562  004737  033646                  JSR     PC,DVTXRX       ;GO TO DEVICE RX. SUBROUTINE
 8128
 8129  032566  013737  006464  006534          MOV     DVRXA,TEMP2
 8130  032574  013737  006466  006536          MOV     DVRCC,TEMP3     ;SET UP ADDR.AND CC.
 8131  032602  004737  017642                  JSR     PC,LOGRXC       ;LOG COMPLETED
 8132  032606  063737  006464  006466          ADD     DVRXA,DVRCC
 8133  032614  105077  153646                  CLRB    @DVRCC
 8134  032620                          PRINTF  #OPBFPT
   (7) 032620  012746  002514                                                      MOV     #OPBFPT,-(SP)
   (6) 032624  012746  000001                                                      MOV     #1,-(SP)
   (3) 032630  010600                                                              MOV     SP,R0
   (4) 032632  104417                                                              TRAP    C$PNTF
   (4) 032634  062706  000004                                                      ADD     #4,SP
 8135  032640  022737  054105  002520          CMP     #''EX,OPBUF     ;COMPARE FOR EX OF ''EXIT''
 8136  032646  001320                          BNE     LISCKA          ;IF NOT EXIT THEN GO BACK
 8137  032650  022737  052111  002522          CMP     #''IT,OPBUF+2   ;IF FIRST HALF OK CHECK NEXT PART
 8138  032656  001314                          BNE     LISCKA          ;IF NOT EXIT THE GO BACK
 8139  032660  012737  000005  006562          MOV     #TAL,MODTYP     ;CHANGE MODE TO TALK
 8140  032666  000137  030636                  JMP     GTRX2           ;RETURN TO DISPATCHER
```

```
 8141
 8142
```

```
                        .SBTTL          DEVICE FUNCTION SUBROUTINES
                        .SBTTL                      DEVICE INIT SUBROUTINE

                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       DVINIT- DEVICE INIT ROUTINE
                        ;       THIS ROUTINE IS DEVICE DEPENDENT CODE THAT INITS
                        ;       THE DEVICE BEING TESTED.
                        ;       IT SETS THE DEVICE UP TO THE MODE IT IS TO RUN IN AND
                        ;       INITIATES THE START,STACK,ACK SEQUENCE IF THE 'RNODE'(REMOTE
                        ;       NODE)INPUT INDICATES THE REMOTE NODE IS NON-ITEP.
                        ;
                        ; INPUTS:       'FHDPLX' INDICATES IF MODE IS FULL OR HALF DUPLEX. (1=FULL)
                        ;               ADDRESS POINTERS (SEL0,...) ALREADY POINT TO DEVICE'S REG.S
                        ;
                        ;               'MLTYP' INDICATES THE LOOP TYPE (1=TTL,2=CAB,3=RM,4=LM)
                        ;               'RNODE' INDICATES THE TYPE OF REMOTE NODE (ITEP=1,NON-ITEP=0)
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;
                        ;               'CTSSR' - CLEAR TO SEND SUB ROUTINE
                        ;               'DVIN31' - SEND CONTROL AND REC OR TIME OUT
                        ;               'CLRRTS' - CLEAR REQUEST TO SEND ROUTINE
                        ;                'LGDVE' - LOG DEVICE ERROR TO EVENT LOG
                        ;
                        ; CALLING SEQUENCE:
                        ;               JSR     PC,DVINIT
                        ;--
                        DVINIT:
                                ;MASTER CLEAR DEVICE
032672  012777  000001  156524          MOV     #RESET,@TXCSR   ;DO A MASTER CLEAR
                        
032700  105777  156512                  TSTB    @RXCSR          ;SEE IF IT WORKED
032704  001423                          BEQ     DVIN1           ;BRANCH IF OK
032706                                  BREAK
(3)     032706  104422                                                  TRAP    C$BRK

                                ;REPORT ERROR IF RESET
                                ;DOES NOT WORK

032710  012737  016441  006534          MOV     #DVEM0,TEMP2
032716  017737  156474  006536          MOV     @RXCSR,TEMP3
032724  017737  156474  006540          MOV     @TXCSR,TEMP4    ;LOAD UP ERRM. AND REG OUTPUTS
032732  004737  017652                  JSR     PC,LGDVE        ;LOG TIME OUT WAITING FOR RUN
032736  005237  006502                  INC     ERRCNT
032742                                  ERRSOFT 5,DVEM0,ERR13
(4)     032742  104457                                                  TRAP    C$ERSOFT
(5)     032744  000005                                                  .WORD   5
(5)     032746  016441                                                  .WORD   DVEM0
```

```
  (5)  032750  017326                                                                    .WORD   ERR13
8243   032752  000747                    BR DVINIT                 ;GO BACK AND TRY MSTR CLR AGAIN IF ERROR
8244
8245                                     ;SET TTL LOOP IF REQU'D
8246
8247   032754  042737  000003  006574  DVIN1:  BIC     #3,FLAG           ;CLEAR INPUT AND OUTPUT INT FLAGS
8248   032762  042777  000010  156434          BIC     #TTLL,@TXCSR      ;CLEAR INTERNAL LOOP
8249   032770  022737  000001  006564          CMP     #TTL,MLTYP        ;IS TTL SELECTED
8250   032776  001004                          BNE     DVIN3             ; IF NOT GO TO 3
8251   033000  052777  000010  156416          BIS     #TTLL,@TXCSR      ;ELSE SET INTERNAL LOOP
8252   033006  000455                          BR      DVIN37
8253
8254   033010  022737  000002  006564  DVIN3:  CMP     #CABLE,MLTYP
8255   033016  001451                          BEQ     DVIN37            ;IF CABLE LOOP SKIP CHECK
8256                                                                     ;FOR MODEM READY
8257
8258   033020  022737  000004  006562          CMP     #DOW,MODTYP       ;CHECK IF DLL
8259   033026  001002                          BNE     DVIN3A            ;BRANCH IF NOT DLL
8260   033030  000137  033460                  JMP     DVINEX            ;ELSE EXIT
8261
8262   033034  012777  000002  156354  DVIN3A: MOV     #DTR,@RXCSR       ;SET UP DTR.
8263
8264   033042  012737  002000  006634          MOV     #2000,TIMER1
8265   033050  005737  006634          DVIN38: TST     TIMER1
8266   033054  001022                          BNE     DVIN39            ;IF TIMER NOT OUT GO TO 39
8267
8268                                     ;SET ERROR FOR NO MODEM READY
8269
8270   033056  012737  017067  006534          MOV     #DVEM6,TEMP2
8271   033064  017737  156326  006536          MOV     @RXCSR,TEMP3
8272   033072  017737  156326  006540          MOV     @TXCSR,TEMP4
8273   033100  004737  017652                  JSR     PC,LGDVE
8274   033104  005237  006502                  INC     ERRCNT
8275   033110                                  ERRSOFT 11,DVEM6,ERR13
  (4)  033110  104457                                                                    TRAP    C$ERSOFT
  (5)  033112  000013                                                                    .WORD   11
  (5)  033114  017067                                                                    .WORD   DVEM6
  (5)  033116  017326                                                                    .WORD   ERR13
8276   033120  000745                          BR      DVIN3A            ;THEN TRY TO SET DTR AGAIN
8277   033122                          DVIN39: BREAK
  (3)  033122  104422                                                                    TRAP    C$BRK
8278   033124  017737  156266  011444          MOV     @RXCSR,IRXCSR     ;GET COPY OR RXCSR
8279   033132  032737  001000  011444          BIT     #BIT9,IRXCSR      ;IS MODEM READY SET
8280   033140  001743                          BEQ     DVIN38
8281   033142  013777  011440  156250  DVIN37: MOV     DPVP1,@PCSAR      ;SET UP PCASAR
8282   033150  005737  011472                  TST     RNODE             ;CHECK REMOTE NODE
8283   033154  001141                          BNE     DVINEX            ;EXIT IF ITEP
8284   033156  005737  006566                  TST     FHDPLX            ;IS THIS FULL DUPLEX
8285   033162  001536                          BEQ     DVINEX            ;BANCH IF NOT
8286
8287                                     ;SET UP TO SEND STRT
8288   033164  112737  000005  002645          MOVB    #5,HDMSG+1        ;SET UP ENQ
8289   033172  052737  000060  006574          BIS     #RXM!TXM,FLAG     ;SET FLAG WORD
8290   033200  012737  000074  006640          MOV     #60.,TIMERS       ;SET TIMER FOR 1 MINUTE
8291   033206  004737  035136                  JSR     PC,CTSSR          ;SET CTS IF NESC.
8292   033212  012737  000006  002646  DVIN41: MOV     #6,HDMCC          ;SET UP STRT CODE
```

```
 8293  033220  004737  035006              JSR     PC,DVIN31       ;GO TX STRT AND CHK FOR RX.
 8294  033224  005737  006640              TST     TIMERS
 8295  033230  001466                      BEQ     DVIN81          ;IF TIMER EXPIERED EXIT
 8296
 8297  033232  022737  000006  002660  DVIN4:  CMP     #6,RHDMCC       ;IS THE RCVD=STRT
 8298  033240  001441                      BEQ     DVIN8           ;IF SO GO TO ASTRT
 8299  033242  022737  000007  002660          CMP     #7,RHDMCC       ;IS IT A STACK
 8300  033250  001360                      BNE     DVIN41          ;IF NOT STACK ETIHER GO BACK
 8301
 8302  033252  004737  035136  DVIN9:  JSR     PC,CTSSR        ;SET REQUEST TO SEND
 8303  033256  042737  001010  006574          BIC     #QTX!PAD,FLAG   ;CLEAR TX COMPT FLAG.
 8304  033264  012737  000001  002646          MOV     #1,HDMCC        ;SET UP ACK
 8305  033272  012737  002645  011450          MOV     #HDMSG+1,MSGPTR ;SET UP POINTER
 8306  033300  013737  002654  011452          MOV     HDMC,MSGCC
 8307  033306  012737  000010  011454          MOV     #8.,SYNCC       ;SET UP SYNC COUNT
 8308  033314  052777  000120  156102          BIS     #TXENA!TINTEN,@TXCSR
 8309  033322  032737  000010  006574  DIVN91: BIT     #QTX,FLAG
 8310  033330  001053                      BNE     DVINEX          ;EXIT IF ACK SENT
 8311  033332                              BREAK
  (3)  033332  104422                                                             TRAP    C$BRK
 8312  033334  005737  006640              TST     TIMERS
 8313  033340  001370                      BNE     DIVN91          ;IF NOT TIMER EXPIRED RECHK TX.
 8314  033342  000421                      BR      DVIN81          ;IF TIMER OUT REPORT IT
 8315
 8316  033344  012737  000007  002646  DVIN8:  MOV     #7,HDMCC        ;SET POTINTER TO STACK
 8317  033352  004737  035006              JSR     PC,DVIN31               ;AND GO SEND STACK
 8318  033356  005737  006640              TST     TIMERS
 8319  033362  001411                      BEQ     DVIN81          ;REPORT ERROR IF TIME OUT
 8320  033364  022737  000001  002660          CMP     #1,RHDMCC       ;IS IT ACK RCVD?
 8321  033372  001432                      BEQ     DVINEX          ;IF SO EXIT
 8322  033374  022737  000007  002660          CMP     #7,RHDMCC       ;IS IT STACK RCVD
 8323  033402  001723                      BEQ     DVIN9           ;IF SO SEND ACK
 8324  033404  000757                      BR      DVIN8           ;IF NEITHER SEND ANOTHER ACK
 8325
 8326                                      ;DO ERROR AND REPEAT
 8327
 8328  033406  012737  017003  006534  DVIN81: MOV     #DVEM5,TEMP2
 8329  033414  013737  002660  006536          MOV     RHDMCC,TEMP3
 8330  033422  013737  002646  006540          MOV     HDMCC,TEMP4     ;LOAD UP ERRM. AND REG OUTPUTS
 8331  033430  004737  017652              JSR     PC,LGDVE        ;LOG TIME OUT WAITING FOR RUN
 8332  033434  005237  006502              INC     ERRCNT
 8333  033440                              ERRSOFT 10.,DVEM5,ERR13
  (4)  033440  104457                                                             TRAP    C$ERSOFT
  (5)  033442  000012                                                             .WORD   10
  (5)  033444  017003                                                             .WORD   DVEM5
  (5)  033446  017326                                                             .WORD   ERR13
 8334  033450  005237  006476              INC     OPVAR           ;COUNT HOW MANY TIMES WE DO THIS.
 8335  033454  000137  032672              JMP     DVINIT          ;TRY ALL OVER AGAIN
 8336
 8337  033460  004737  035310  DVINEX: JSR     PC,CLRRTS       ;CLEAR RTS IF NESC
 8338  033464  042737  173777  006574          BIC     #173777,FLAG    ;CLEAR FLAG WORD
 8339  033472  052737  002000  006574          BIS     #INOVR,FLAG     ;SET THE INITT OVER FLAG
 8340  033500  000207                      RTS     PC              ;RETURN TO CALLER
 8341
 8342
```

```
                                .SBTTL                  DEVICE GET MODEM STATUS SUBROUTINE
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       'DVMODS''  GET MODEM STATUS
                                ;
                                ; IMPLICIT INPUTS:
                                ;       THE BIT POSITION AND AVAILABILITY OF THE MODEM SIGNALS CTS,DSR,...RI,..
                                ;       FOUND IN THE DEPENDENT PORTION OF THE GLOBAL EQUATES SECTION.
                                ;
                                ; OUTPUTS:
                                ;       CURRENT MODEM SIGNAL VALUES IN 'MODS''
                                ;
                                ; CALLING SEQUENCE:
                                ;       JSR     PC,DVMODS
                                ;--
033502  017737  155710  007550  DVMODS: MOV     @RXCSR,MODS
033510  042737  000040  007550         BIC     #BIT5,MODS      ;CLEAR BIT 5
033516  032777  000040  155700         BIT     #BIT5,@TXCSR    ;SEE IT TM OR SQ SET
033524  001403                       BEQ     DVMEX           ;IF NOT EXIT
033526  052737  000040  007550         BIS     #BIT5,MODS      ;IF SET SET BIT 5 IN MODS
033534  042737  106720  007550  DVMEX:  BIC     #106720,MODS    ;CLEAR ALL UNUSED BITS
033542  000207                       RTS     PC              ;RETURN TO CALLER
```

```
                                        .SBTTL                  DEVICE QUEUE RECEIVE SPACE SUBROUTINE
                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       DVRXQ - THIS SUBROUTINE QUEUES THE RECIEVER BUFFER SPACE TO THE
                                        ;               DEVICE, THEN CLEARS THE QRX BIT OF THE FLAG WORD.
                                        ;
                                        ; INPUTS:
                                        ;       DVRXA = ADDRESS OF RX BUFFER SPACE
                                        ;       DVRCC = BYTE CHAR COUNT OF RX BUFFER
                                        ;       QRX FLAG BIT = SET BY CALLING ROUTINE
                                        ;
                                        ; OUTPUTS:
                                        ;       QRX FLAG BIT = CLEARED BY ROUTINE
                                        ;
                                        ; CALLING SEQUENCE:
                                        ;       JSR     PC,DVRXQ
                                        ;--

033544                                  DVRXQ:
033544  032737  000004  006574                  BIT     #QRX,FLAG
033552  001434                                  BEQ     DVREX                   ;IF NOT RX THEN EXIT
                                                                                ;ELSE QUE RX
033554  042737  000444  006574                  BIC     #QRX+#BCC+#RXM,FLAG     ;CLEAR FLAG FOR RX
033562  005737  011472                          TST     RNODE                   ;IF NON ITEP GO TO 2
033566  001415                                  BEQ     DVRX2
033570  052737  000440  006574                  BIS     #RXM+#BCC,FLAG          ;GET JUST THE DATA NO CRC.
033576  013737  006464  011460                  MOV     DVRXA,RMSGPT
033604  012737  000072  011462                  MOV     #72,RMSGCC              ;SET UP RX TO GET ITEP MSG.
033612  012737  000070  006466                  MOV     #70,DVRCC
033620  000406                                  BR      DVRX3

                                        ;ENABLE RX, RX INTERRUPTS,AND DATA SET INTERRUPTS

033622  012737  002657  011460  DVRX2:  MOV     #RHDMSG+1,RMSGPT        ;SET UP POINTER
033630  013737  002654  011462          MOV     HDMC,RMSGCC             ;AND CC
033636  052777  000160  155552  DVRX3:  BIS     #RINTEN!RXENA!#DSITEN,@RXCSR
033644  000207                  DVREX:  RTS     PC                      ;RETURN TO CALLER
```

```
                                .SBTTL                  DEVICE TRANSMIT AND RECEIVE SUBROUTINE

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       DVTXRX-DEVICE TRANSMIT AND RECEIVE ROUTINE
                                ;       THIS CODE QUES THE TRANSMIT BUFFER TO THE DEVICE
                                ;       IF NEEDED. THE CODE THEN WAITS FOR A TX COMPLE,
                                ;       RX COMPLETE OR BOTH. THE CODE REPORTS A TIME OUT
                                ;       ERROR IF NO  OUTPUT INTERRUPT IS RECIEVED BEFORE
                                ;       60 SECONDS. AFTER REPORTING ERROR TIMER IS RE STARTED
                                ;       AND DEVICE WILL CONTINUE TO WAIT FOR INTERRUPT.
                                ;
                                ;INPUTS:
                                ;       'DVTXA' = ADDRESS OF TRANSMIT MSG.
                                ;       'DVTCC' = BYTE COUNT OF TRANSMIT MSG.
                                ;    'QTX' BIT  = SET IF TRANSMIT REQUESTED
                                ;    'ETX' BIT  = SET IF TRNASMIT EXPECTED
                                ;    'ERX' BIT  = SET IF RECIEVE EXPECTED
                                ;
                                ;OUTPUTS:
                                ;       'DVTXA' = ADDRESS OF TX MSG. COMPLETED
                                ;       'DVTCC' = BYTE COUNT OF TX MSG. COMPLETED
                                ;       'QTX'   = SET IF TX COMPLETED
                                ;       'DVRXA' = ADDRESS OF RX MSG. COMPLETED
                                ;       'DVRCC' = BYTE COUNT OF RX MSG. COMPLETED
                                ;       'QRX'   = SET IF RX COMPLETED
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;               'LGDVE'  - LOG DEVICE ERROR TO EVENT LOG
                                ;
                                ; CALLING SEQUENCE:
                                ;       JSR     PC,DVTXRX
                                ;--
033646  032737  000010  006574  DVTXRX: BIT     #QTX,FLAG               ;ANY TX TO QUE
033654  001444                        BEQ     DVTR3                   ;IF NOT GO WAIT FOR OUPUT
033656  042737  001030  006574        BIC     #QTX+#TXM+PAD,FLAG      ;CLEAR FLAG
033664  004737  035136                JSR     PC,CTSSR                ;GO SET CTS
033670  005737  011472                TST     RNODE
033674  001412                        BEQ     DVTR1                   ;IF NON-ITEP GO TO 1
033676  052737  000020  006574        BIS     #TXM,FLAG               ;SET THE BODY BIT
033704  013737  006450  011450        MOV     DVTXA,MSGPTR
033712  013737  006452  011452        MOV     DVTCC,MSGCC             ;AND SET UP FOR ACTUAL DATA
033720  000414                        BR      DVTR2
                                        ;ENABLE TX AND TX INTER.

033722  112737  000201  002645  DVTR1:  MOVB    #201,HDMSG+1            ;SET UP SOH
033730  012737  002645  011450        MOV     #HDMSG+1,MSGPTR         ;SET POINTER TO HEADER
033736  013737  006452  002646        MOV     DVTCC,HDMCC
033744  013737  002654  011452        MOV     HDMC,MSGCC              ;SET CC FOR HEADER
033752  012737  000177  011454  DVTR2:  MOV     #177,SYNCC              ;SET UP FOR 177 SYNCS.
033760  052777  000120  155436        BIS     #TXENA!#TINTEN,@TXCSR
```

```
 8519  033766  012737  000074  006640  DVTR3:  MOV     #60.,TIMERS             ;SET TIMER FOR 60 SECS
 8520
 8521  033774                          DVTR8:  BREAK
  (3)  033774  104422                                                                  TRAP    C$BRK
 8522  033776  005737  006640                  TST     TIMERS                  ;IS TIMER EXPIRED
 8523  034002  001022                          BNE     TOINOT
 8524
 8525                                          ;LOG ERROR TIME OUT RX OR TX NOT COMPLETED
 8526
 8527  034004  012737  016602  006534          MOV     #DVEM2,TEMP2
 8528  034012  017737  155400  006536          MOV     @RXCSR,TEMP3
 8529  034020  017737  155400  006540          MOV     @TXCSR,TEMP4
 8530  034026  004737  017652                  JSR     PC,LGDVE
 8531  034032  005237  006502                  INC     ERRCNT
 8532  034036                                  ERRSOFT 7,DVEM2,ERR13
  (4)  034036  104457                                                                  TRAP    C$ERSOFT
  (5)  034040  000007                                                                  .WORD   7
  (5)  034042  016602                                                                  .WORD   DVEM2
  (5)  034044  017326                                                                  .WORD   ERR13
 8533  034046  000747                          BR      DVTR3           ;RETURN TO CHECK TIMER
 8534
 8535  034050  032737  000010  006574  TOINOT: BIT     #QTX,FLAG       ;IS IT TX COMPL?
 8536  034056  001406                          BEQ     DVTR4           ;BRANCH IF TX NOT DONE.
 8537  034060  004737  035310                  JSR     PC,CLRRTS
 8538  034064  032737  000100  006574          BIT     #ERX,FLAG       ;ARE WE EXPECTING TO RX
 8539  034072  001416                          BEQ     DVTREX          ;BRANCH IF NOT.
 8540
 8541  034074  032737  000004  006574  DVTR4:  BIT     #QRX,FLAG       ;IS RX DONE
 8542  034102  001734                          BEQ     DVTR8           ;GO BACK AND TIME IF NOT
 8543
 8544  034104  032737  000200  006574          BIT     #ETX,FLAG       ;ARE WE EXPECTG TO TX.
 8545  034112  001406                          BEQ     DVTREX          ;BRANCH IF NOT.
 8546
 8547  034114  032737  000010  006574          BIT     #QTX,FLAG       ;IS IT TX COMPLETED
 8548  034122  001724                          BEQ     DVTR8           ;GO BACK AND TIME OUT
 8549  034124  004737  035310                  JSR     PC,CLRRTS       ;CLEAR RTS IF NESC.
 8550  034130  000207                  DVTREX: RTS     PC              ;AND EXIT
 8551
```

```
                        ; DEVICE DEPENDENT SUBROUTINES




                        .SBTTL                  DEVICE INTERRUPT SERVICE ROUTINES



                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       RECEIVER INTERRUPT ROUTINE. WHEN A RX INT. OCCURS
                        ;       THIS ROUTINE DECIDES IF IT IS A RX STATUS,DATA SET
                        ;       CHANGE OR DATA INTERRUPT. IF IT IS A DATA SET CHANGE
                        ;       INTERRUPT IT PUTS THE STATUS IN ''CMODS'' AND COMPARES
                        ;       THAT STATUS TO THE OLD STATUS IN 'MODS'. IF THEY ARE
                        ;       THE SAME THAT MEANS THE INTERRUPT WAS CAUSED BY A GLITCH
                        ;       ON ONE OF THE LINES. IF THEY ARE DIFFERENT THEN A HARD
                        ;       MODEM ERROR HAS OCCURED. IN ANY EVENT THE MODEM STATUS
                        ;       CHANGE IS LOGGED.
                        ;       IF A DATA INT. OCCURS THE ROUTINE PUTS THE DATA AWAY
                        ;       IN A BUFFER POINTED TO BY 'RMSGPT'' THE MSG. COUNT IS
                        ;       DECREMENTED BY ONE BYTE. IF COUNT IS EQUAL TO ZERO AND
                        ;       'BCC'' BIT AND 'RXM'' BIT IS SET THEN RX IS DISABLED AND
                        ;       'QRX'' BIT IS SET. IF COUNT IS ZERO AND 'BCC'' BIT IS SET
                        ;       BUT 'RXM'' BIT IS NOT SET THEN MSG COUNT IS SET TO LENGHT
                        ;       RECVD IN HEADER AND 'RMSGPT'' IS SET TO RX BUFFER LOCATION
                        ;       AND 'RXM'' BIT IS SET.
                        ;       IF COUNT IS EQUAL TO ZERO AND 'BCC'' IS NOT SET THEN
                        ;       COUNT IS SET TO 2 AND 'RMSGPT'' IS SET TO 'BCCW'' AND
                        ;       'BCC'' BIT IS SET.
                        ;
                        ;       IF A STATUS INTERRUPT OCCURS THEN OVERRUN ERROR BIT IS CHECKED.
                        ;       AN ERROR IS LOGGED AND ''QRX'' IS SET AND THE RX IS DISABLED.
                        ;
                        ; INPUTS:
                        ;       RMSGPT - ADDRESS OF RX BUFFER
                        ;       RMSCC  - COUNT OF DATA TO BE RXED.
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;               ''LOGMSC''      - LOG MODEM STATUS CHANGE
                        ;               ''LGDVE''       - LOG DEVICE ERROR
                        ;--

034132                          BGNSRV  DVRXI
(3)  034132                                                             DVRXI::
034132  017737  155260  011444          MOV     @RXCSR,IRXCSR   ;MOV RX CSR TO IMAGE
034140  032737  000010  006570          BIT     #MOCHK,PARAM    ;ANY MODEM CHANGES TO REPORT
034146  001456                          BEQ     RXIN21          ;IF NOT IGNORE DS CHANGE.
034150  032737  002000  006574          BIT     #INOVR,FLAG     ;IS INIT OVER
034156  001452                          BEQ     RXIN21          ;NO THEN IGNORE DS CHANGE.
034160  005737  011444                  TST     IRXCSR
034164  100047                          BPL     RXIN21          ;IF DATA SET CHANGE IS NOT SET BR
034166  013737  011444  011442          MOV     IRXCSR,CMODS    ;MOV THE NEW MODEM STATUS IN
034174  042737  106760  011442          BIC     #106760,CMODS
034202  032777  000040  155214          BIT     #TM,@TXCSR
```

```
 8622  034210  001403                          BEQ     RXIN2           ;IF TEST MODE SET
 8623  034212  052737  000040  011442          BIS     #TM,CMODS       ;SET IT IN NEW STATUS
 8624  034220  013737  011442  006536  RXIN2:  MOV     CMODS,TEMP3
 8625  034226  013737  007550  006540          MOV     MODS,TEMP4
 8626  034234  023737  006540  006536          CMP     TEMP4,TEMP3     ;COMPARE OLD TO CURRENT
 8627  034242  001406                          BEQ     GLINC           ;INC GLITCH COUNT
 8628  034244  005237  011470                  INC     MHRCNT          ;INC HARD COUNT
 8629  034250  012737  016411  006534          MOV     #HRDMSG,TEMP2   ;SET UP HARD MESG.
 8630  034256  000405                          BR      RXIN1
 8631  034260  005237  011466          GLINC:  INC     MGLCNT          ;INC GLITCH COUNT
 8632  034264  012737  016363  006534          MOV     #GLMSG,TEMP2    ;SET UP GLITCH
 8633  034272  004737  020026          RXIN1:  JSR     PC,LOGMSC       ;GO LOG MODEM STATUS CHANGE
 8634  034276  013737  011442  007550          MOV     CMODS,MODS      ;MOVE CURRENT TO OLD
 8635                                          ;
 8636                                          ;TEST FOR STATUS OR DATA
 8637
 8638  034304  032737  002200  011444  RXIN21: BIT     #RSTARY!RDATRY,IRXCSR
 8639  034312  001544                          BEQ     RXINEX          ;IF NEITHER EXIT
 8640  034314  017737  155102  011446          MOV     @RDSR,IRDSR
 8641  034322  032737  000200  011444          BIT     #RDATRY,IRXCSR  ;IS THIS DATA
 8642  034330  001455                          BEQ     RXIN3           ;IF NOT GO TO 3
 8643
 8644                                          ;GET HERE WITH GOOD DATA
 8645
 8646  034332  013702  011460          RXIN4:  MOV     RMSGPT,R2
 8647  034336  113722  011446                  MOVB    IRDSR,(R2)+     ;STORE DATA AWAY
 8648  034342  010237  011460                  MOV     R2,RMSGPT       ;PUT POINTER BACK
 8649
 8650
 8651  034346  005337  011462                  DEC     RMSGCC
 8652  034352  001124                          BNE     RXINEX          ;GET OUT IF NOT ALL DONE
 8653  034354  032737  000400  006574          BIT     #BCC,FLAG       ;IS THE BCC FLAG ALREADY SE
 8654  034362  001066                          BNE     RXIN5           ;BRANCH IF YES.
 8655  034364  032737  100000  011446          BIT     #RERR,IRDSR     ;IS THE ERR CHK BIT SET INDICATING
 8656                                                                  ;GOOD BCC.
 8657  034372  001022                          BNE     RXIN6           ;BRANCH IF GOO
 8658  034374  013737  011446  006536          MOV     IRDSR,TEMP3
 8659  034402  013737  011444  006540          MOV     IRXCSR,TEMP4
 8660  034410  012737  016677  006534          MOV     #DVEM3,TEMP2
 8661  034416  004737  017652                  JSR     PC,LGDVE
 8662  034422  005237  006502                  INC     ERRCNT
 8663  034426                                  ERRSOFT 8,DVEM3,ERR13
  (4)  034426  104457                                                  TRAP    C$ERSOFT
  (5)  034430  000010                                                  .WORD   8
  (5)  034432  016677                                                  .WORD   DVEM3
  (5)  034434  017326                                                  .WORD   ERR13
 8664
 8665  034436  000467                          BR      RXIN8           ;DISABLE INTERRUPTS AND EXIT
 8666
 8667  034440  052737  000400  006574  RXIN6:  BIS     #BCC,FLAG       ;SET FLAG
 8668  034446  012737  000002  011462          MOV     #2.,RMSGCC      ;SET THE COUNT TO 2
 8669  034454  012737  011464  011460          MOV     #BCCW,RMSGPT    ;SET POINTER TO BCC WORD
 8670  034462  000460                          BR      RXINEX
 8671
 8672                                          ;STATUS CHECK
 8673
```

```
 8674  034464  032737  002000  011444  RXIN3:  BIT     #RSTARY,IRXCSR  ;IS THIS A STATUS INT.
 8675  034472  001454                  BEQ     RXINEX          ;EXIT IF NOT
 8676
 8677                                  ;LOG OVERRUN ERROR
 8678
 8679  034474  012737  016737  006534          MOV     #DVEM4,TEMP2
 8680  034502  013737  011446  006536          MOV     IRDSR,TEMP3
 8681  034510  013737  011444  006540          MOV     IRXCSR,TEMP4
 8682  034516  004737  017652                  JSR     PC,LGDVE
 8683  034522  005237  006502                  INC     ERRCNT
 8684  034526                                  ERRSOFT 9,DVEM4,ERR13
   (4) 034526  104457                                                          TRAP    C$ERSOFT
   (5) 034530  000011                                                          .WORD   9
   (5) 034532  016737                                                          .WORD   DVEM4
   (5) 034534  017326                                                          .WORD   ERR13
 8685  034536  000424                  BR      RXIN7
 8686
 8687  034540  032737  000040  006574  RXIN5:  BIT     #RXM,FLAG       ;IS THE RX M BODY BIT SET
 8688  034546  001020                  BNE     RXIN7           ;IF YES THEN ALL DONE
 8689  034550  052737  000040  006574          BIS     #RXM,FLAG
 8690  034556  042737  000400  006574          BIC     #BCC,FLAG       ;CLEAR BCC AND SET RXM
 8691  034564  013737  006464  011460          MOV     DVRXA,RMSGPT    ;MOVE ADDRESS TO POINTER
 8692  034572  013737  002660  011462          MOV     RHDMCC,RMSGCC   ;MOVE THE CHAR COUNT IN
 8693  034600  013737  002660  006466          MOV     RHDMCC,DVRCC    ;SET THE CC TO AMOUNT IN HEADER
 8694  034606  000406                  BR      RXINEX          ;AND FINISH.
 8695
 8696  034610  052737  000004  006574  RXIN7:  BIS     #QRX,FLAG       ;SET FLAG BIT
 8697
 8698  034616  042777  000120  154572  RXIN8:  BIC     #RINTEN+RXENA,@RXCSR    ;CLEAR INTAND RX ENABLE
 8699
 8700  034624                          RXINEX:
 8701  034624                                  ENDSRV
   (3) 034624                                                                  L10020:
   (2) 034624  000002                                                                  RTI
```

```
                        .SBTTL                  DEVICE TRANSMIT INTERRUPT ROUTINE

                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       DEVICE TRANSMIT INT. ROUTINE
                        ;
                        ;       WHEN A TRANSMIT BUFFER EMPTY CAUSES AN INTERRUPT TO OCCUR
                        ;       THE PROGRAM COMES TO THIS ROUTINE.
                        ;       IF THE SYNC COUNT''SYNCC'' IS NON ZERO TSOM IS SET
                        ;       A SYNC CHAR IS LOADED TO TDSR AND THE SYNC COUNT IS
                        ;       DECREMENTED.
                        ;
                        ;       IF THE SYNC COUNT IS ZERO TSOM AND TEOM ARE RESET
                        ;       AND THE ''PAD'' BIT IN FLAG WORD IS CHECKED IF IT IS
                        ;       SET THEN A PAD(377) CHAR IS LOADED TO TDSR AND TX
                        ;       INTERRUPT ENABLE IS CLEARD.
                        ;
                        ;       IF THE SYNC COUNT IS ZERO AND THE ''PAD'' FLAG IS
                        ;       CLEAR THEN A BYTE IS PUT IN TDSR FROM THE ADDRESS
                        ;       IN MSGPTR AND THE MSG COUNT IS DECREMENTED
                        ;
                        ;       IF THE MSG COUNT GOES TO ZERO THE ''TXM'' BIT IS
                        ;       CHECKED IF IT  IS SET THE ''PAD'' FLAG IS SET
                        ;       IF IT IS CLEAR THEN IT GETS SET AND MSGPTR IS
                        ;       LOADED WITH THE ADDRESS OF TXBUFF AND THE MSG
                        ;       COUNT IS LOADED WITH THE COUNT OF THE MSG TO
                        ;       BE TRANSMITTED.
                        ;
                        ; INPUTS:
                        ;       MSGPTR - IS SET TO THE ADDRESS OF THE MSG OR HEADER TO BE TX'D
                        ;       MSGCC  - IS SET TO THE COUNT OF MSG TO BE TX'D
                        ; OUTPUTS:
                        ;       QTX - THIS BIT IS SET WHEN MSG IS TX'D OK.
                        ;--

034626                          BGNSRV  DVTXI
(3) 034626                                                                      DVTXI::
034626  005737  011454          TST     SYNCC           ;ANY SYNCS TO SEND
034632  001406                  BEQ     TXIN1           ;IF NOT GO TO 1
034634  013777  011456  154564  MOV     SYNCW,@TDSR     ;ELSE SET TSOM AND SYNC WORD
034642  005337  011454          DEC     SYNCC           ;DEC SYNC COUNT
034646  001056                  BNE     TXINEX          ;IF NOT ZERO EXIT
034650  042777  001400  154550  TXIN1:  BIC     #TEOM!TSOM,@TDSR
034656  032737  001000  006574          BIT     #PAD,FLAG       ;IS THE PAD BIT SET
034664  001412                  BEQ     TXIN2           ;GO TO 2 IF NOT SET
034666  012777  000377  154532  MOV     #377,@TDSR      ;LOAD FF TO TX DATA REG.
034674  042777  000100  154522  BIC     #TINTEN,@TXCSR  ;CLEAR TX INT ENABLE
034702  052737  000010  006574  BIS     #QTX,FLAG       ;SET THE TX COMPLETE IN FLAG
034710  000435                  BR      TXINEX          ;AND EXIT
034712  013702  011450  TXIN2:  MOV     MSGPTR,R2       ;LOAD R2 WITH TX ADDR.
034716  112277  154504          MOVB    (R2)+,@TDSR     ;LOAD DATA BYTE
034722  010237  011450          MOV     R2,MSGPTR       ;RESTORE POINTER
034726  005337  011452          DEC     MSGCC           ;DEC CC
034732  001024                  BNE     TXINEX
034734  052777  001000  154464  BIS     #TEOM,@TDSR
034742  032737  000020  006574  BIT     #TXM,FLAG       ;IS THIS THE END OF DATA MSG.
```

```
 8764   034750  001012                      BNE     TXIN3           ;IF SO SET THE PAD BIT
 8765   034752  052737  000020  006574      BIS     #TXM,FLAG       ;IF NOT MUST BE END OF HEADER
 8766   034760  013737  006450  011450      MOV     DVTXA,MSGPTR    ;SO SET UP MSGPTR FOR MSG
 8767   034766  013737  006452  011452      MOV     DVTCC,MSGCC     ;AND THE CC FOR MSG.
 8768   034774  000403                      BR      TXINEX
 8769   034776  052737  001000  006574  TXIN3:  BIS     #PAD,FLAG       ;SET THE PAD BIT
 8770
 8771   035004                          TXINEX:
 8772   035004                                  ENDSRV
   (3)  035004                                                          L10021:
   (2)  035004  000002                                                          RTI
```

```
                        .SBTTL                  DEVICE TRANSMIT CONTROL MSG
                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       THIS ROUTINE DOES THE FOLLOWING
                        ;       QUES A RX SPACE AT RHDMSG+1
                        ;       QUES A TX MSG FROM HDMSG+1
                        ;       CHECKS FOR A TIMER EXPIRED
                        ;       IF EXPIRED RETURN TO CALLER
                        ;       ELSE CHECK FOR A TX MSG COMPLETED
                        ;       IF TX COMPLETED CHECK FOR RX COMPLETED
                        ;       ELSE RECHECK TIMER AND TX COMPLETED UNTIL
                        ;       EITHER TX COMPLETE OR TIME OUT
                        ;       IF TX COMPLETE AND RX NOT COMPLETE THEN
                        ;       REQUE TX MSG.
                        ;       ELSE IF RX COMPLETE RETURN.
                        ;
                        ; INPUTS:
                        ;       TXM             - SET IN FLAG WORD
                        ;       HDMSG+2         - TYPE OF CONTROL MSG..
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;               "CLRRTS"        - CLEAR REQUEST TO SEND IF HALF DUP.
                        ; CALLING SEQUENCE:
                        ;               JSR     PC,DVIN31
                        ; RETURN:
                        ;       RETURN TO CALLER IF SOMETHING RX'D OR TIMER OUT.
                        ;
                        ;--
035006  042737  000004  006574  DVIN31: BIC     #QRX,FLAG               ;CLEAR RX COMPLE.
035014  012737  002657  011460          MOV     #RHDMSG+1,RMSGPT        ;SET UP POINTER
035022  013737  002654  011462          MOV     HDMC,RMSGCC             ;AND CC
                                                                        ;ENABLE RCVR.
035030  052777  000160  154360          BIS     #RINTEN!RXENA!DSITEN,@RXCSR
                                        ;SET UP TXMITTER TO SEND
035036  004737  035136          DVIN32: JSR     PC,CTSSR                ;SET RTS .
035042  042737  001010  006574          BIC     #QTX!PAD,FLAG           ;CLEAR TX COMPT FLAG.
035050  012737  002645  011450          MOV     #HDMSG+1,MSGPTR         ;MOVE THE CURRENT POINTER TO MSGPTR.
035056  013737  002654  011452          MOV     HDMC,MSGCC
035064  012737  000010  011454          MOV     #8.,SYNCC               ;SET UP SYNC COUNT
035072  052777  000120  154324          BIS     #TXENA!TINTEN,@TXCSR
                                        ;NOW WAIT FOR TIME OUT OR TX COMPLETE
035100                  DVIN35: BREAK
  (3)   035100  104422                                                  TRAP    C$BRK
035102  005737  006640                  TST     TIMERS          ;IS IT TIMED OUT
035106  001412                          BEQ     DVIN34          ;IF YES EXIT
035110  032737  000010  006574          BIT     #QTX,FLAG       ;IS TX DONE
035116  001770                          BEQ     DVIN35          ;IF NOT GO BACK AND CK TIME OUT
035120  004737  035310                  JSR     PC,CLRRTS       ;GO CLEAR RTS IF NESC.
035124  032737  000004  006574          BIT     #QRX,FLAG       ;DID WE RX ANYTHING
```

```
 8829  035132  001741                    BEQ     DVIN32          ;IF NOT RETRANSMIT LAST
 8830  035134  000207          DVIN34:   RTS     PC              ;RETURN TO CALLER
 8831
```

```
                                        .SBTTL                  DEVICE RTS TO CTS DELAY
                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       CTSSR--THIS ROUTINE SETS REQUEST TO SEND TO MODEM
                                        ;       AND CHECKS FOR CLEAR TO SEND TO COME BACK
                                        ;       IF CTS DOES NOT COME BACK BEFORE TIMER EXPIRES
                                        ;       AND ERROR IS REPORTED AND WE TRY AGAIN.
                                        ;       THE ROUTINE IS SKIPPED IF INTERNAL LOOP IS SET.
                                        ;
                                        ;
                                        ;; OUTPUTS:
                                        ;
                                        ; SUBORDINATE ROUTINES USED:
                                        ;               "LGDVE" -       LOG DEVICE ERROR
                                        ; CALLING SEQUENCE:
                                        ;               JSR     PC,CTSSR
                                        ;--

035136  022737  000001  006564  CTSSR:  CMP     #1,MLTYP        ;IS THIS TTL LOOP
035144  001460                         BEQ     DVTXR9          ;BR IF YES
                                        ;SET RTS AND WAIT FOR CTS
035146  032737  004000  006574  DVTXR3: BIT     #FIRST,FLAG
035154  001014                         BNE     CTSS3           ;IF NOT FIRST TIME SKIP DELY
035156  012737  000000  006530         MOV     #0,TEMP
035164  005237  006530          CTSS4: INC     TEMP
035170                                 BREAK
(3)     035170  104422                                                  TRAP    C$BRK
035172  005737  006530                 TST     TEMP
035176  001372                         BNE     CTSS4           ;IF NOT ZERO GO BACK
035200  052737  004000  006574         BIS     #FIRST,FLAG     ;SET FIRST FLAG.
035206  052777  000004  154202  CTSS3: BIS     #RTS,@RXCSR     ;SET REQUEST TO SEND
035214  012737  001750  006634         MOV     #1000.,TIMER1   ;SET UP TIMER
035222                          DVTXR2: BREAK
(3)     035222  104422                                                  TRAP    C$BRK
035224  032777  020000  154164         BIT     #CTS,@RXCSR     ;IS CLEAR TO SEND BACK
035232  001025                         BNE     DVTXR1          ;BR. IF CTS IS SET
035234  005737  006634                 TST     TIMER1          ;ELSE TEST IF TIME EXPIRED
035240  001370                         BNE     DVTXR2          ;BR IF TIME NOT EXPRIED.
                                        ;SET ERROR FOR NO CTS
035242  012737  016520  006534         MOV     #DVEM1,TEMP2
035250  017737  154142  006536         MOV     @RXCSR,TEMP3
035256  017737  154142  006540         MOV     @TXCSR,TEMP4
035264  004737  017652                 JSR     PC,LGDVE
035270  005237  006502                 INC     ERRCNT
035274                                 ERRSOFT 6,DVEM1,ERR13
(4)     035274  104457                                                  TRAP    C$ERSOFT
(5)     035276  000006                                                  .WORD   6
(5)     035300  016520                                                  .WORD   DVEM1
(5)     035302  017326                                                  .WORD   ERR13
035304  000720                         BR      DVTXR3          ;THEN TRY TO SET RTS AGAIN
```

```
 8883  035306
 8884  035306  000207          DVTXR1:
                               DVTXR9: RTS     PC              ;
```

```
                                        .SBTTL                  DEVICE CLEAR REQUEST TO SEND
                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       THIS ROUTINE CLEARS REQUEST TO SEND IF
                                        ;       IN HALF DUPLEX MODE
                                        ; CALLING SEQUENCE:
                                        ;       JSR     PC,CLRRTS
                                        ;--

035310                                  CLRRTS:

035310  005737  006566                          TST     FHDPLX          ;IS THIS FULL DUPLEX
035314  001011                                  BNE     DVTR5           ;BRANCH IF YES
035316  012737  002000  006534                  MOV     #2000,TEMP2
035324  005337  006534                  CLRR1:  DEC     TEMP2
035330  001375                                  BNE     CLRR1           ;DELAY A WHILE TO ALLOW CRC
                                                                        ;TO BE TXMITT'D BEFORE YOU
035332  042777  000004  154056                  BIC     #RTS,@RXCSR     ;CLEAR REQUEST TO SEND
035340  000207                          DVTR5:  RTS     PC              ;RETURN TO CALLER
```

```
 8918
 8919                                   .EVEN
 8920
 8921  035342                           ENDTST
  (3)  035342                                                  L10017:
  (3)  035342  104401                                                 TRAP    C$ETST
 8922
```

```
 8929
 8930
```

```
 8932                                   .SBTTL  HARDWARE PARAMETER CODING SECTION
 8933
 8934
 8935                                   ;++
 8936                                   ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
 8937                                   ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
 8938                                   ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
 8939                                   ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
 8940                                   ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
 8941                                   ; WITH THE OPERATOR.
 8942                                   ;--
 8943
 8944  035344                           BGNHRD
  (3)  035344  000016                                                    .WORD L10022-L$HARD/2
  (3)  035346                                                          L$HARD::
 8945
 8955
 8956                                   .SBTTL          DEVICE INDEPENDENT SECTION
 8957
 8958  035346                           GPRML   DPLX,0,1,YES
  (4)  035346  000130                                                    .WORD   T$CODE
  (4)  035350  035402                                                    .WORD   DPLX
  (4)  035352  000001                                                    .WORD   1
 8959
 8960
 8971
 8972
 8973                                   .SBTTL          DEVICE DEPENDENT SECTION
 8974
 8975  035354                                   GPRMA   CSRADR,2,0,160000,177776,YES
  (4)  035354  001031                                                    .WORD   T$CODE
  (4)  035356  035433                                                    .WORD   CSRADR
  (4)  035360  160000                                                    .WORD   T$LOLIM
  (4)  035362  177776                                                    .WORD   T$HILIM
 8976  035364                                   GPRMA   VECTOR,4,0,300,776,YES
  (4)  035364  002031                                                    .WORD   T$CODE
  (4)  035366  035461                                                    .WORD   VECTOR
  (4)  035370  000300                                                    .WORD   T$LOLIM
  (4)  035372  000776                                                    .WORD   T$HILIM
 8977  035374                                   GPRML   RNODM,14,1,YES
  (4)  035374  006130                                                    .WORD   T$CODE
  (4)  035376  035514                                                    .WORD   RNODM
  (4)  035400  000001                                                    .WORD   1
 8978
 8979  035402                                   ENDHRD
  (2)                                                                    .EVEN
  (3)  035402                                                          L10022:
 8980
 8981                                   .NLIST  BEX
 8982
 8983                                   ;DEVICE INDEPENDENT QUESTIONS
 8984
 8985  035402  052506  046114  042040  DPLX:   .ASCIZ  /FULL DUPLEX OPERATION : /
 8986
 8994
 8995                                   ;DEVICE DEPENDENT QUESTION
```

```
 8996
 8997  035433     104  053105  041511  CSRADR: .ASCIZ  /DEVICE CSR ADDRESS : /
 8998  035461     111  052116  051105  VECTOR: .ASCIZ  /INTERRUPT VECTOR ADDRESS: /
 8999  035514  042522  047515  042524  RNODM:  .ASCIZ  /REMOTE NODE ''ITEP'':/
 9000                                  .LIST   BEX
 9001                                          .EVEN
 9002
 9009
```

```
 9012                                   ;.SBTTL SOFTWARE PARAMETER CODING SECTION
 9013
 9014                                   ;++
 9015                                   ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
 9016                                   ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
 9017                                   ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
 9018                                   ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
 9019                                   ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
 9020                                   ; WITH THE OPERATOR.
 9021                                   ;--
 9022
 9023                                   ;       BGNSFT
 9024
 9033
 9034
 9035                                   ;       ENDSFT
 9036
 9037
 9044
 9045                                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 9046                                   ; TEMPORARY PATCH AREA - FOR DEBUG PURPOSES
 9047                                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 9048
 9049  035540                           $PATCH:
 9050  035540  000030                           .BLKW   30
 9051
 9058
 9059  035620                                   LASTAD
  (2)                                                                     .EVEN
  (4)  035620  000000                                                     .WORD   0
  (4)  035622  000000                                                     .WORD   0
  (3)  035624                       L$LAST::
 9060  035624                                   ENDMOD
 9061
 9062          000001               .END
```

```
ACT    = 000003            4295#   7121    7489    7565    7803
ACTATV  030006             7283    7489#
ACTBCR  027612             7301    7454#
ACTCHK  030222             7263    7531#
ACTCLB  027134             7354    7368#
ACTCLP  030334             7297    7559#
ACTCLR  026624             7261    7315#
ACTCOP  027432             7271    7422#
ACTCRC  030236             7292    7537#
ACTCSE  026760             7266    7340#
ACTCST  027052             7267    7356#
ACTDLL  030054             7287    7503#
ACTDME  027360             7303    7402    7405#
ACTDMQ  027352             7304    7404#
ACTDMS  027330             7302    7399#
ACTDMX  027366             7406#
ACTECH  030132             7291    7520#
ACTEQQ  027554             7275    7443#
ACTEXT  026710             7307    7328#
ACTHLP  026644             7265    7321#
ACTLIS  030044             7286    7500#
ACTLLP  030344             7298    7561#
ACTLPX  030362             7556    7558    7560    7562    7565#
ACTLXX  030424             7529    7550    7553    7566    7570#
ACTMEX  030000             7436    7452    7469    7474    7480    7483    7485#
ACTME1  027734             7458    7460    7462    7464    7466    7473#
ACTMOP  030314             7295    7555#
ACTMOS  030244             7306    7540#
ACTMSO  027634             7276    7457#
ACTMS1  027642             7277    7459#
ACTMS2  027652             7278    7461#
ACTMS3  027662             7279    7463#
ACTMS4  027672             7280    7465#
ACTMS5  027702             7281    7467#
ACTMS6  027720             7282    7470#
ACTM2X  030102             7490    7498    7501    7504    7507    7511#
ACTNO   030122             7290    7517#
ACTNUF  026614             7300    7312#
ACTNUL  026622             7260    7313#
ACTNUM  027442             7272    7425#
ACTOPM  027534             7273    7438#
ACTPAS  030016             7284    7492#
ACTPRO  030252             7293    7543#
ACTPRT  026720             7305    7331#
ACTQFG  030256             7532    7535    7538    7541    7545#
ACTREC  030036             7285    7497#
ACTREX  020704             5950    5965#
ACTRHL  020640             5949    5957#
ACTRLG  020714             5951    5969#
ACTRLP  030354             7299    7563#
ACTRNL  020636             5948    5954#
ACTRPS  030304             7294    7552#
ACTRUN  026734             7264    7335#
ACTSEX  027744             7308    7477#
ACTSHO  026634             7262    7318#
ACTSHW  027174             7346    7360    7379#   7391
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
ACTSTE  027374            7268    7412#
ACTSTS  030230            7274    7534#
ACTSTT  027404            7269    7415#
ACTSTX  027412            7413    7416#
ACTSZE  027422            7270    7419#
ACTTAL  030074            7289    7509#
ACTTLP  030324            7296    7557#
ACTTRA  030064            7288    7506#
ADDCC   022136            6190#   7203    7233
ADDC1   022232            6192    6202#
ADR   = 000020 G          4278#
ALCK    031032            4760    7712#
ALCK1   031144            7768    7777#   7823
ALCK2   031210            7778    7787#   7855
ALCK2A  031430            7820    7825#
ALCK3   031454            7790    7827    7830#
ALCK3A  031546            7837    7843#
ALCK3B  031560            7844    7846#
ALCK3C  031554            7841    7845#   7852
ALCK4   031602            7831    7850#
ALCK4A  031610            7849    7853#
ALCK4B  031622            7854    7856#
ALCK5   031074            7767#   7845
ALLTR   031074            7641    7665    7690    7717    7766#
ASSEMB= 000010            4030
ATVMOD= 000027            4432#   4905
BAD     006551            4729#   5504    7928*
BADCHR= 000051            4450#   5020
BASM1   015173            5336#   6150
BASM2   015164            5335#   6156
BASM3   015155            5334#   6154
BCC   = 000400            4386#   8418    8421    8653    8667    8690
BCCW    011464            5114#   8669
BDCLK   013712            5297#   6826
BIT0  = 000001 G          4278#   4315    4364    4504
BIT00 = 000001 G          4278#
BIT01 = 000002 G          4278#
BIT02 = 000004 G          4278#
BIT03 = 000010 G          4278#
BIT04 = 000020 G          4278#
BIT05 = 000040 G          4278#
BIT06 = 000100 G          4278#
BIT07 = 000200 G          4278#
BIT08 = 000400 G          4278#
BIT09 = 001000 G          4278#
BIT1  = 000002 G          4278#   4316    4365    4497    4505
BIT10 = 002000 G          4278#   4390    4502
BIT11 = 004000 G          4278#   4392    4503
BIT12 = 010000 G          4278#   4481
BIT13 = 020000 G          4278#   4479
BIT14 = 040000 G          4278#   4483
BIT15 = 100000 G          4278#   4512    4513
BIT2  = 000004 G          4278#   4317    4366    4482    4506
BIT3  = 000010 G          4278#   4318    4367    4507
BIT4  = 000020 G          4278#   4319    4382    4498    4508
BIT5  = 000040 G          4278#   4320    4384    4484    4485    4499    8374    8375    8377
```

```
BIT6   = 000100 G          4278#  4368   4500   4509
BIT7   = 000200 G          4278#  4369   4501   7602   7605
BIT8   = 000400 G          4278#  4386   4510
BIT9   = 001000 G          4278#  4388   4480   4511   8279
BLDBEX   022344            6247   6251#
BLDBUF   022234            6229#  7112   7119   7204   7234   7375   7614
BLDB1    022244            6234#
BLDB2    022306            6242#  6249
BLDB3    022324            6245#  6250
BOE    = 000400 G          4278#
BUFEX    014057            5300#  7187   7216
BUFLIM= 001000             4283#  4673   4674   4675   6191   6198   6320   7185   7214   7368   7610
BYTBIT   006510            4708#  6151   7400*  7404*
CABLE  = 000002            4302#  8254
CBLLOP= 000045             4446#  5049
CCURAD   006446            4688#  6332*  6340   6344*  7108*  7116   7232   7237*  7353*
CHECK  = 000003            4412#  4953
CLEAR  = 000001            4410#  4891   7178   7315
CLIACT   026434            7148   7252#
CLIALN= 000007             4403#  5018
CLIALP= 000006             4402#  4939   4949   5023
CLIBCR   011775            5202#  7454
CLIBDL   011670            5200#  7569
CLIBIF= 000003             4399#
CLIBR  = 000002            4398#  4888   4890   4892   4895   4897   4906   4908   4910   4912   4914   4917   4919
                           4934   4936   4946   4954   4965   4967   4984   4988   4991   4998   5000   5002   5004
                           5006   5008   5010   5012   5016   5019   5026   5028   5033   5048   5050   5052   5054
                           5059
CLIBRX   011650            5199#  6570
CLIDEC= 000011             4405#  5032   5058
CLIERM   011553            5196#  5913   7153
CLIERR= 000000             4396#  4898   5020   5939
CLIEXI= 000001             4397#  4880   4882   4884   4886   4925   4927   5064   5932   5934   5936   5938   5940
CLINBG   011626            5198#  6608
CLINPS   011732            5201#  7525
CLINUF   011603            5197#  5918   7157
CLINUM= 000005             4401#
CLIOCT= 000010             4404#  4941   4943
CLIPW    012073            5204#  7582
CLIRAC   020610            5908   5943#
CLIRT    020522            5907   5930#
CLISEO   012042            5203#  7429
CLISPA= 000004             4400#  4878   4902   4923   4932   4940   5017   5930
CLISTR= 000012             4406#  4881   4883   4885   4887   4889   4891   4894   4896   4903   4905   4907   4909
                           4911   4913   4916   4918   4924   4926   4933   4935   4951   4953   4964   4966   4983
                           4987   4990   4997   4999   5001   5003   5005   5007   5009   5011   5025   5027   5047
                           5049   5051   5053   5933   5935   5937
CLITRE   007706            4875#  7147
CLI$PM   011540            5194#  7145
CLI$RP   011546            5195#  5905
CLKBR    006616            4768#  6821*
CLKCSR   006614            4767#  5709*  5730*  6802   6812*  6813*  6814*  6822*  6951*  7080*
CLKEN    006624            4771#  5730   6806*  6815*  6822   7080
CLKHZ    006622            4770#  5712   5727   5866   6823   6830
CLKINT   017446 G          5707#  6880
CLKSET   017422            5662#  6805   6811
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
CLKVEC  006620          4769#   6800*   6820*   6880
CLRRTS  035310          8337    8537    8549    8827    8895#
CLRR1   035324          8900#   8901
CMDBUF  002666          4615#   5905    5906    7145    7146
CMODS   011442          5105#   8619*   8620*   8623*   8624    8634
CMPBUF  005144          4675#   6318*   6332    7108    7352
CMPPTR  006440          4685#   6326*   6327*   6340*   6341*   6342*   6346*   7103*   7115    7222*   7230*   7231    7235*
                        7340*   7341    7350*   7351    7588*   7896
CMPSEX  032076          7894    7971#
CMPSR   031622          7842    7893#
CMPS1   031762          7921#   7935
CMPS2   031766          7924#
CMPS3   031654          7899#   7943
CMPS5   032064          7940#
CMPS5A  032070          7937    7941#
CMPS6   032032          7922    7926    7933#
CMPS7   031750          7914    7919#
CMPTOT  006442          4686#   6335*   6338    6347*   7120*   7220*   7223    7236*   7343    7347*   7580    7591
CMSG0 = 000020          4425#   4999
CMSG1 = 000021          4426#   4997
CMSG2 = 000022          4427#   5001
CMSG3 = 000023          4428#   5003
CMSG4 = 000024          4429#   5007
CMSG5 = 000025          4430#   5005
CMSG6 = 000026          4431#   5009
CONOTM  006544          4726#   5544
CONTIN  006546          4727#
CPTR    006520          4716#   6232    6236*   7111*   7115*   7201*   7205    7231*   7235    7351*   7364*   7612*   7640*
                        7661*   7687*   7713*   7779    7784*   7796    7846*   7853    7895*   7900    7904*   8122*
CPTRR   006516          4715#   7637*   7664*   7688*   7715*   7769    7774*   7802*   7807    7816*   7829*   7850    7896*
                        7906    7909*   8080*
CR      016226          5378#   5859
CRC   = 000040          4441#
CRCB  = 000020          4319#   7537
CSHEXP= 000006          4415#   4924
CSHTRN= 000007          4416#   4926
CSRADR  035433          8975    8997#
CTOTCC  006444          4687#   6334*   6342    6344    6345*   6350*   7098*   7213    7218    7238*   7348*
CTS   = 020000          4479#   4796    8869
CTSSR   035136          8291    8302    8503    8813    8852#
CTSS3   035206          8859    8866#
CTSS4   035164          8861#   8864
CURADD  006522          4717#   6234    6239    6241    6251*   7110*   7116*   7202*   7207    7232*   7237    7369*   7611*
CURCC   006514          4714#   6190    6197    6200*   6201    6235    6240    6251    7106*   7118*   7371*   7432*   7445*
                        7468*   7471*   7473*   7610*
C$AU  = 000052          4030#   7044
C$AUTO= 000061          4030#   6932
C$BRK = 000022          4030#   8232    8277    8311    8521    8822    8862    8868
C$BSEG= 000004          4030#
C$BSUB= 000002          4030#
C$CEFG= 000045          4030#
C$CLCK= 000062          4030#   6803    6809
C$CLEA= 000012          4030#   6971
C$CLOS= 000035          4030#
C$CLP1= 000006          4030#
C$CVEC= 000036          4030#
```

```
C$DCLN= 000044          4030#  6785
C$DODU= 000051          4030#
C$DRPT= 000024          4030#
C$DU  = 000053          4030#  7007
C$EDIT= 000003          4030#  4116
C$ERDF= 000055          4030#
C$ERHR= 000056          4030#
C$ERRO= 000060          4030#
C$ERSF= 000054          4030#
C$ERSO= 000057          4030#  7916  7930  7939  7979  8242  8275  8333  8532  8663  8684  8881
C$ESCA= 000010          4030#
C$ESEG= 000005          4030#
C$ESUB= 000003          4030#
C$ETST= 000001          4030#  8921
C$EXIT= 000032          4030#  6896  6955  7137  7176
C$GETB= 000026          4030#
C$GETW= 000027          4030#
C$GMAN= 000043          4030#  5905  6823  7145  8066
C$GPHR= 000042          4030#  6843
C$GPLO= 000030          4030#
C$GPRI= 000040          4030#
C$INIT= 000011          4030#  6912
C$INLP= 000020          4030#
C$MANI= 000050          4030#  7133
C$MEM = 000031          4030#
C$MSG = 000023          4030#  5505  5509  5513  5517  5536  5545
C$OPEN= 000034          4030#
C$PNTB= 000014          4030#  5504  5508  5512  5516  5535  5544
C$PNTF= 000017          4030#  5859  5862  5899  5913  5918  5958  6150  6154  6156  6196  6570  6608
                        6826   7089  7126  7153  7157  7187  7196  7216  7226  7322  7388  7429  7454
                        7525   7569  7582  8037  8116  8134
C$PNTS= 000016          4030#  5996  6014  6019  6025  6032  6033  6049  6057  6064  6083  6086  6101
                        6116   6390  6428
C$PNTX= 000015          4030#
C$QIO = 000377          4030#
C$RDBU= 000007          4030#  6818
C$REFG= 000047          4030#  6788  6790  6792  6795
C$RESE= 000033          4030#  6953
C$REVI= 000003          4030#  4116
C$RFLA= 000021          4030#
C$RPT = 000025          4030#  6718
C$SEFG= 000046          4030#
C$SPRI= 000041          4030#  6895  6952
C$SVEC= 000037          4030#  6880  6892  6893
C$TPRI= 000013          4030#
DATCKB= 000002          4316#  6398  7531  7578  7660  7799  7848  7893  8061  8115
DCD   = 010000          4481#  4798
DCK   = 000014          4346#  5804
DCLFLG  006556          4736#  6782  6784*  7175*
DDE   = 000022          4359#  5812
DER   = 000010          4344#  5790
DEV1    007676          4854#  6030*  6033  6036*  6038*  6379  7127*  7392*
DEV2    007700          4855#  6031*  6033  6037*  6039*  6382  7128*  7393*
DEV3    007702          4856#  6040*  6389  7129*  7394*
DEV4    007704          4857#  6041*  6394  6398  6402  6423  7130*  7395*
DFPTBL  002130 G        4154#
```

```
DIAGML= 000000            4030
DIVN91  033322            8309#   8313
DLE   = 000020            4358#   5808
DLL     032216            4761    8036#
DLLCM   013632            5294#   8037
DLLMOD= 000033            4436#   4913
DMPE  = 000053            4452#   4943
DMPQ  = 000054            4453#   4945
DMPS  = 000052            4451#   4941    7169    7401
DMSGAD  002172            4540#   6243
DMSGCT  002150            4527#   6242
DOW   = 000004            4296#   7503    8258
DPLX    035402            8958    8985#
DPV   = 000000            4336#
DPVP1   011440            5095#   7602*   7605*   8281
DSITEN= 000040            4499#   8431    8809
DSR   = 001000            4480#   4797
DTR   = 000002            4497#   8262
DUMEX   022134            6158    6164#
DUMPSR  022000            6148#   7171
DUM1    022072            6152    6156#
DUM2    022114            6155    6157#
DUM3    022030            6151#   6162
DUM4    022004            6149#   6161
DVEM0   016441            5422#   8237    8242
DVEM1   016520            5424#   8876    8881
DVEM2   016602            5426#   8527    8532
DVEM3   016677            5428#   8660    8663
DVEM4   016737            5430#   8679    8684
DVEM5   017003            5432#   8328    8333
DVEM6   017067            5434#   8270    8275
DVI   = 000012            4345#   5795
DVINEX  033460            8260    8283    8285    8310    8321    8337#
DVINIT  032672            7608    8225#   8243    8335
DVIN1   032754            8231    8247#
DVIN3   033010            8250    8254#
DVIN3A  033034            8259    8262#   8276
DVIN31  035006            8293    8317    8803#
DVIN32  035036            8813#   8829
DVIN34  035134            8824    8830#
DVIN35  035100            8822#   8826
DVIN37  033142            8252    8255    8281#
DVIN38  033050            8265#   8280
DVIN39  033122            8266    8277#
DVIN4   033232            8297#
DVIN41  033212            8292#   8300
DVIN8   033344            8298    8316#   8324
DVIN81  033406            8295    8314    8319    8328#
DVIN9   033252            8302#   8323
DVMEX   033534            8376    8378#
DVMODS  033502            5840    8373#
DVRCC   006466            4698#   7773*   7792    8119*   8130    8132*   8133*   8424*   8693*
DVRCT   006470            4699#   7638*   7714*   7801*   7805*   7825*   7826    7897*   7942*
DVREX   033644            8416    8432#
DVRXA   006464            4697#   7771*   7791    8117*   8129    8132    8422    8691
DVRXI   034132 G          6892    8611#
```

```
DVRXQ   033544            7775    8124    8414#
DVRX2   033622            8420    8429#
DVRX3   033636            8425    8431#
DVTCC   006452            4691#   7783*   7833    8077*   8085    8508    8514    8767
DVTCT   006454            4692#   7662*   7686*   7712*   7835*   7840    7843
DVTREX  034130            8539    8545    8550#
DVTR1   033722            8505    8512#
DVTR2   033752            8509    8516#
DVTR3   033766            8501    8519#   8533
DVTR4   034074            8536    8541#
DVTR5   035340            8898    8904#
DVTR8   033774            8521#   8542    8548
DVTXA   006450            4690#   7781*   7832    8074*   8084    8507    8766
DVTXI   034626 G          6893    8744#
DVTXRX  033646            7787    8082    8127    8500#
DVTXR1  035306            8870    8883#
DVTXR2  035222            8868#   8872
DVTXR3  035146            8858#   8882
DVTXR9  035306            8854    8884#
ECHO  = 000037            4440#   4966
ECHOB = 000004            4317#   6402    7511    7522    7527    7794
EDDCK   014747            4824    5326#
EDDDE   015062            4827    5329#   7930    7939
EDDER   014732            4822    5325#
EDDLE   015025            4826    5328#   7916
EDDVI   014777            4823    5327#
EDEOP   015115            4828    5330#
EDMOS   015131            4825    5331#
EDRXC   014704            4821    5324#
EDRXQ   014657            4820    5323#
EDTXC   014630            4819    5322#
EDTXQ   014604            4818    5321#
EFM11   014404            5314#   5512
EFM13   014533            5316#   5516
EFM2    014307            5312#   5508
EF.CON= 000036 G          4278#   6792
EF.NEW= 000035 G          4278#   6795
EF.PWR= 000034 G          4278#
EF.RES= 000037 G          4278#   6790
EF.STA= 000040 G          4278#   6788
EMSG0   002215            4528    4552#
EMSG1   002216            4529    4554#
EMSG2   002217            4530    4556#
EMSG3   002220            4531    4558#
EMSG4   002320            4532    4564#
EMSG5   002412            4533    4569#
EMSG6   002513            4534    4573#
EMSG8   002643            4536    4587#
ENADD   006506            4707#   6157    7405*
ENDEVT  021656            5997    6007    6011    6092#
ENDIT   024714            6794    6879#
ENDPS   032136            7974    8001#
ENDQO = 000017            4424#   5015
EOP   = 000024            4360#   5816
ERRCNT  006502            4705#   5839    5842    7597*   7915*   7929*   7938*   7978*   8005    8241*   8274*   8332*   8531*
                          8662*   8683*   8880*
```

```
ERR1     017154 G          5503#   7930
ERR10    017244 G          5511#   7916
ERR13    017326 G          5534#   8242    8275    8333    8532    8663    8684    8881
ERR14    017360 G          5543#
ERR2     017216 G          5507#   7939
ERR4     017274 G          5515#   7979
ERX    = 000100            4368#   7639    7689    7716    7821    7839    8121    8538
ETX    = 000200            4369#   7663    7716    7822    7838    8079    8091    8544
EVL    = 000004 G          4278#
EVMCTS   016171            4807    5367#
EVMDCD   016201            4809    5369#
EVMDSR   016175            4808    5368#
EVMOCG   016051            5360#   6086
EVMOHD   016074            5365#   6101
EVMOST   016154            5366#   6116
EVMRI    016211            4811    5371#
EVMRTS   016205            4810    5370#
EVMSQD   016215            4812    5372#
EVMTM    016221            4813    5373#
EVTADD   007642            4835#   6022*   6025    6046*   6049    6054*   6057    6061*   6064
EVTBCT   007644            4836#   6023*   6025    6047*   6049    6055*   6057    6062*   6064
EVTEND   007546            4785#   5874    6004    6834
EVTF0    015241            5338#   6014
EVTF1    015335            5339#   6019
EVTF2    015364            5340#   6025
EVTF3    015436            5341#   6032    6083
EVTF3C   015450            5342#   5535    6033
EVTF3D   015465            5343#   5544
EVTF4    015507            5344#   6057
EVTF4A   015611            5345#   6064
EVTF4B   015707            5356#   6049
EVTF5A   015766            5358#   5504
EVTLOG   006644            4783    4784#   5878    5994    6002    6831
EVTLST   007606            4818#   6019
EVTMIN   007636            4833#   6018*   6019
EVTPTR   006642            4783#   5864    5879*   5993    6009    6832*
EVTSEC   007634            4832#   6017*   6019
EVTTCK   007640            4834#   6016*   6019
EVTTMP   007646            4837#   6029*   6032    6048*   6049    6056*   6057    6063*   6064    6082*   6083
EXIT   = 000057            4456#   4885    7173    7328
E$END  = 002100            4030#
E$LOAD= 000035             4030#   4116
FACSIM   022360            6314#   7481
FHDPLX   006566            4744#   6846*   8284    8897
FIRST  = 004000            4392#   8858    8865
FLAG     006574            4754#   7594*   7639*   7663*   7689*   7716*   7767    7777    7789    7821*   7822*   7828*   7830
                           7838*   7839*   7847*   8079*   8091*   8121*   8247*   8289*   8303*   8309    8338*   8339*   8415
                           8418*   8421*   8500    8502*   8506*   8535    8538    8541    8544    8547    8615    8653    8667*
                           8687    8689*   8690*   8696*   8751    8755*   8763    8765*   8769*   8803*   8814*   8825    8828
                           8858    8865*
F$AU   = 000015            4030#   7017    7044
F$AUTO= 000020             4030#   6923    6932
F$BGN  = 000040            4030#   4051    5503    5507    5511    5515    5534    5543    5547    5707    6686    6727    6756
                           6896    6923    6941    6955    6980    6991    7017    7028    7070    7137    7176    8611    8744
                           8921    8944    9060
F$CLEA= 000007             4030#   6941    6971
```

```
F$DU   = 000016          4030#  6980   7007
F$END  = 000041          4030#  4051   5505   5509   5513   5517   5536   5545   5547   5731   6718   6896   6912
                         6932   6955   6971   6991   7007   7028   7044   7070   7137   7176   8701   8772   8921
                         8979   9060
F$HARD= 000004           4030#  8944   8979
F$HW   = 000013          4030#  4154   4200
F$INIT= 000006           4030#  6756   6912
F$JMP  = 000050          4030#  5547   6896   6955   6991   7028   7137   7176
F$MOD  = 000000          4030#  4051   9060
F$MSG  = 000011          4030#  5503   5505   5507   5509   5511   5513   5515   5517   5534   5536   5543   5545
F$PROT= 000021           4030#  6727   6733
F$PWR  = 000017          4030#
F$RPT  = 000012          4030#  6686   6718
F$SEG  = 000003          4030#
F$SOFT= 000005           4030#
F$SRV  = 000010          4030#  5707   5731   8611   8701   8744   8772
F$SUB  = 000002          4030#
F$SW   = 000014          4030#
F$TEST= 000001           4030#  7070   8921
GETCL   025460           7134   7143#  7154   7158   7161   7163   7165   7172   7179   7181   7188   7197   7211
                         7217   7227   7241   7583
GETTRM  024530           6797   6839#  6844
GETRCL  020330           5901#  5914   5919   5923
GLINC   034260           8627   8631#
GLMSG   016363           5398#  8632
GOOD    006550           4728#  5504   7927*
GTRA2   025100           7083#
GTRA3   025142           7086   7091#
GTRA4   025150           7096#
GTRA5   025374           7092   7127#  8013   8038
GTREX   030534           7594#
GTRX2   030636           7610#  8012   8093   8140
GTR9    030432           7139   7168   7576#
GTXRXB  025100           7082#
G$CNTO= 000200           4030#
G$DELM= 000372           4030#
G$DISP= 000003           4030#
G$EXCP= 000400           4030#
G$HILI= 000002           4030#
G$LOLI= 000001           4030#
G$NO   = 000000          4030#  5905   7145   8066
G$OFFS= 000400           4030#  5905   6823   7145   8066   8958   8975   8976   8977
G$OFSI= 000376           4030#  5905   6823   7145   8066   8958   8975   8976   8977
G$PRMA= 000001           4030#  8975   8976
G$PRMD= 000002           4030#  5905   6823   7145   8066
G$PRML= 000000           4030#  8958   8977
G$RADA= 000140           4030#  5905   7145   8066
G$RADB= 000000           4030#
G$RADD= 000040           4030#  6823
G$RADL= 000120           4030#  8958   8977
G$RADO= 000020           4030#  8975   8976
G$XFER= 000004           4030#
G$YES  = 000010          4030#  6823   8958   8975   8976   8977
HDMC    002654           4602#  8306   8430   8515   8806   8816
HDMCC   002646           4596#  8292*  8304*  8316*  8330   8514*
HDMSG   002644           4594#  8288*  8305   8512*  8513   8815
```

```
HELP   = 000000          2195#   4021    4047    4090    4118    4138    4156    4233    4247    4263    5118    5167    5179
                         5452    5464    5487    5558    6688    6703    6711    6735    6758    6898    6925    6943    6957
                         6982    6993    7019    7030    7057    7064    7072    8907    8923    8946    9003    9025    9038
                         9052
HELPDC= 000000           2200#   2210    4108    4174    4204    4324    4348    4371    4462    4487    5067    5131    5156
                         5280    5347    5390    5403    5439    5520    5820    5983    6070    6848    6884    8026    8146
                         8168    8346    8384    8437    8555    8568    8704    8961    8987
HLP    = 000005          4414#   4879    4881    7162    7325
HLPEND   003036          4628#   7323
HLPF     012241          5206#   5958    7322
HLPTAB   003016          4620#   7321
HLP0     012163          5205#   7126
HLP1     012246          4620    5207#
HLP2     012261          4621    5208#
HLP3     012376          4622    5212#
HLP3A    012463          4623    5213#
HLP4     012510          4624    5214#
HLP4A    012567          4625    5215#
HLP5     012645          4626    5216#
HLP6     012735          4627    5217#
HOE    = 100000 G        4278#
HRDMSG   016411          5399#   8629
HSMSE    002653          4600#
IBE    = 010000 G        4278#
IDU    = 000040 G        4278#
IER    = 020000 G        4278#
ININT  = 000001          4364#
INIT1    024206          6783    6787#
INOVR  = 002000          4390#   8339    8615
INTPRI   011434          5092#   6876*
INVEC    011430          5090#   6873*   6892
IRDSR    011446          5107#   8640*   8647    8655    8658    8680
IRXCSR   011444          5106#   8278*   8279    8612*   8617    8619    8638    8641    8659    8674    8681
ISR    = 000100 G        4278#
IXE    = 004000 G        4278#
I$AU   = 000041          4030#   7017#   7044#
I$AUTO= 000041           4030#   6923#   6932#
I$CLN  = 000041          4030#   6941#   6955    6971#
I$DU   = 000041          4030#   6980#   7007#
I$HRD  = 000041          8944#   8979#
I$INIT= 000041           4030#   6756#   6896    6912#
I$MOD  = 000041          4030#   4051#   9060#
I$MSG  = 000041          4030#   5503#   5505#   5507#   5509#   5511#   5513#   5515#   5517#   5534#   5536#   5543#   5545#
I$PROT= 000040           4030#   6727#
I$PTAB= 000041           4030#
I$PWR  = 000041          4030#
I$RPT  = 000041          4030#   6686#   6718#
I$SEG  = 000041          4030#   7070
I$SETU= 000041           4030#
I$SRV  = 000041          4030#   5707#   5731#   8611#   8701#   8744#   8772#
I$SUB  = 000041          4030#   7070
I$TST  = 000041          4030#   7070#   7137    7176    8921#
J$JMP  = 000167          4030#   5547    6991    7028
KEYWD1   003010          4616#   5921    5961*   5965*   5970*   7160    7162    7164    7166    7169    7173    7178    7180
                         7182    7315*   7318*   7325*   7328*   7331*   7335*   7345    7359    7401*   7412*   7415*   7477
                         7482*
```

```
LCLKEN= 000100           4309#   6806
LGDVE   017652           5788#   8240    8273    8331    8530    8661    8682    8879
LIS   = 000006           4298#   7500    8092
LISCK   032462           4763    8115#
LISCKA  032510           8117#   8136    8138
LISMOD= 000032           4435#   4911
LISP    013555           5269#   8116
LMDLOP= 000046           4447#   5051
LNCNT   006474           4702#   5855    5857*   5861*   7600*
LOE   = 040000 G         4278#
LOGCMD  017772           5810#   7940
LOGCML  017754           5806#   7917
LOGCMP  017736           5802#   7911
LOGDVI  017670           5793#   7607
LOGEOP  020010           5814#   8006
LOGEX   020300           5844    5881#
LOGMSC  020026           5833#   8633
LOGRXC  017642           5785#   7793    8131
LOGRXQ  017624           5780#   7776    8125
LOGS1   020044           5773    5778    5783    5787    5839#
LOGS2   020272           5875    5879#
LOGS3   020076           5791    5800    5805    5809    5813    5817    5836    5848#
LOGS4   020152           5856    5860#
LOGS5   020176           5850    5852    5863#
LOGTXC  017606           5775#   7834    8086
LOGTXQ  017570           5770#   7785    8078
LOGUNT  006552           4734#   6837*   6839*   6840    6843
LOOPS   003114           4653#   6388
LOT   = 000010 G         4278#
LP0     013437           4653    5242#   6387
LP00    013440           5243#   6384
LP1     013447           4654    5244#
LP2     013460           4655    5245#
LP3     013466           4656    5246#
LP4     013501           4657    5247#
L$ACP   002110 G         4116#
L$APT   002036 G         4116#
L$AU    025064 G         4116    7017#
L$AUT   002070 G         4116#
L$AUTO  025032 G         4116    6923#
L$CCP   002106 G         4116#
L$CLEA  025034 G         4116    6941#
L$CO    002032 G         4116#
L$DEPO  002011 G         4116#
L$DESC  011504 G         4116    5176#
L$DESP  002076 G         4116#
L$DEVP  002060 G         4116#
L$DISP  002124 G         4116    4136#
L$DLY   002116 G         4116#
L$DTP   002040 G         4116#
L$DTYP  002034 G         4116#
L$DU    025056 G         4116    6980#
L$DUT   002072 G         4116#
L$DVTY  011474 G         4116    5165#
L$EF    002052 G         4116#
L$ENVI  002044 G         4116#
```

```
L$ETP   002102 G        4116#
L$EXP1  002046 G        4116#
L$EXP4  002064 G        4116#
L$EXP5  002066 G        4116#
L$HARD  035346 G        4116    8944#
L$HIME  002120 G        4116#
L$HPCP  002016 G        4116#
L$HPTP  002022 G        4116#
L$HW    002130 G        4116    4154#
L$ICP   002104 G        4116#
L$INIT  024172 G        4116    6756#
L$LADP  002026 G        4116#
L$LAST  035624 G        4116    9059#
L$LOAD  002100 G        4116#
L$LUN   002074 G        4116#
L$MREV  002050 G        4116#
L$NAME  002000 G        4116#
L$PRIO  002042 G        4116#
L$PROT  024164 G        4116    6727#
L$PRT   002112 G        4116#
L$REPP  002062 G        4116#
L$REV   002010 G        4116#
L$RPT   024156 G        4116    6686#
L$SPC   002056 G        4116#
L$SPCP  002020 G        4116#
L$SPTP  002024 G        4116#
L$STA   002030 G        4116#
L$TEST  002114 G        4116#
L$TIML  002014 G        4116#
L$UNIT  002012 G        4116#   6840
L10000  002150          4154    4200#
L10001  017214          5505#
L10002  017242          5509#
L10003  017272          5513#
L10004  017324          5517#
L10005  017356          5536#
L10006  017414          5545#   5547
L10007  017566          5731#
L10010  024162          6718#
L10012  025030          6896    6912#
L10013  025032          6932#
L10014  025054          6955    6971#
L10015  025062          6991    7007#
L10016  025070          7028    7044#
L10017  035342          7137    7176    8921#
L10020  034624          8701#
L10021  035004          8772#
L10022  035402          8944    8979#
L5060   013573          5271#   6823
MCREP   032112          7972    7978#
MGLCNT  011466          5115#   5516    7598*   7971    7980*   8631*
MHRCNT  011470          5116#   5516    7599*   7973    7981*   8628*
MLTYP   006564          4741#   5797    7122*   7128    7138*   7393    7494*   7513*   7555*   7557*   7559*   7561*   7563*
                        7568*   7576    8249    8254    8852
MOBITE  007570          4803#   6114
MOBITS  007552          4796#   6102
```

```
MOCHK = 000010           4318#   6423    7540    8613
MODE    006576           4757#   7617
MODES   003076           4645#   6381
MODLOC= 000003           4303#
MODREM= 000004           4304#
MODS    007550           4789#   5846    8373*   8374*   8377*   8378*   8625    8634*
MODTYP  006562           4738#   5796    7121*   7127    7392    7489*   7492*   7497*   7500*   7503*   7506*   7509*   7523
                         7565    7615    7803    7819    7836    8092*   8139*   8258
MOMSGS  007570           4807#   6103
MOP   = 000043           4305#   4444#
MOSC  = 000056           4455#   4987
MO0     013346           4645    5235#
MO1     013356           4646    5236#
MO2     013367           4647    5237#
MO3     013377           4648    5238#
MO4     013406           4649    5239#
MO5     013423           4650    5240#
MO6     013430           4651    5241#
MSC   = 000016           4356#   5835
MSCMS   014461           5315#   7979
MSGCC   011452           5109#   8306*   8508*   8515*   8760*   8767*   8816*
MSGLIM= 000017           4285#   4676    4677    4678    6323    7194    7224
MSGPTR  011450           5108#   8305*   8507*   8513*   8757    8759*   8766*   8815*
MSGTRN  014075           5301#   7187    7196    7216    7226
MSGTRU  014126           5302#   6196
MSGTYP  006512           4712#   6237    7105*   7117*   7370*   7438*   7457*   7459*   7461*   7463*   7465*   7467*   7470*
                         7613*
MSG0    002214           4528    4541    4551#
MSG0C   002150           4528#
MSG1    002215           4529    4542    4553#
MSG1C   002152           4529#
MSG2    002216           4530    4543    4555#
MSG2C   002154           4530#
MSG3    002217           4531    4544    4557#
MSG3C   002156           4531#
MSG4    002220           4532    4545    4559#
MSG4C   002160           4532#
MSG5    002320           4533    4546    4565#
MSG5C   002162           4533#   7106    7118    7371    7468
MSG6    002412           4534    4547    4570#
MSG6C   002164           4534#   7471
MSG8    002642           4536    4549    4586#
MSG8C   002170           4536#
NEW     024522           6796    6837#   6841
NO    = 000036           4439#   4951    7517    7520    7546
NOCLK   013736           5298#   7089
NOD0    007706           4878#
NOD1    007712           4879#
NOD10   007770           4886#
NOD100  010616           4965#
NOD101  010622           4966#
NOD102  010636           4967#
NOD103  010642           4983#
NOD104  010656           4984#
NOD105  010662           4987#
NOD106  010676           4988#
```

```
NOD107  010702            4990#
NOD11   007772            4887#
NOD110  010716            4991#
NOD111  010722            4994#
NOD112  010726            4997#
NOD113  010742            4998#
NOD114  010746            4999#
NOD115  010764            5000#
NOD116  010770            5001#
NOD117  011004            5002#
NOD12   010004            4888#
NOD120  011010            5003#
NOD121  011024            5004#
NOD122  011030            5005#
NOD123  011044            5006#
NOD124  011050            5007#
NOD125  011064            5008#
NOD126  011070            5009#
NOD127  011104            5010#
NOD13   010010            4889#
NOD130  011110            5011#
NOD131  011130            5012#
NOD132  011134            5014#
NOD133  011140            5015#
NOD134  011144            5016#
NOD135  011150            5017#
NOD136  011154            5018#
NOD137  011160            5019#
NOD14   010024            4890#
NOD140  011164            5020#
NOD141  011166            5023#
NOD142  011172            5024#
NOD143  011176            5025#
NOD144  011212            5026#
NOD145  011216            5027#
NOD146  011232            5028#
NOD147  011236            5031#
NOD15   010030            4891#
NOD150  011242            5032#
NOD151  011246            5033#
NOD152  011252            5036#
NOD153  011256            5047#
NOD154  011300            5048#
NOD155  011304            5049#
NOD156  011320            5050#
NOD157  011324            5051#
NOD16   010044            4892#
NOD160  011346            5052#
NOD161  011352            5053#
NOD162  011374            5054#
NOD163  011400            5057#
NOD164  011404            5058#
NOD165  011410            5059#
NOD166  011414            5064#
NOD167  020522            5930#
NOD17   010050            4893#
```

```
NOD170  020526          5931#
NOD171  020532          5932#
NOD172  020534          5933#
NOD173  020550          5934#
NOD174  020552          5935#
NOD175  020566          5936#
NOD176  020570          5937#
NOD177  020602          5938#
NOD2    007716          4880#
NOD20   010054          4894#
NOD200  020604          5939#
NOD201  020606          5940#
NOD21   010066          4895#
NOD22   010072          4896#
NOD23   010104          4897#
NOD24   010110          4898#
NOD25   010112          4902#
NOD26   010116          4903#
NOD27   010132          4904#
NOD3    007720          4881#
NOD30   010136          4905#
NOD31   010154          4906#
NOD32   010160          4907#
NOD33   010176          4908#
NOD34   010202          4909#
NOD35   010220          4910#
NOD36   010224          4911#
NOD37   010242          4912#
NOD4    007734          4882#
NOD40   010246          4913#
NOD41   010272          4914#
NOD42   010276          4915#
NOD43   010302          4916#
NOD44   010320          4917#
NOD45   010324          4918#
NOD46   010336          4919#
NOD47   010342          4923#
NOD5    007736          4883#
NOD50   010346          4924#
NOD51   010370          4925#
NOD52   010372          4926#
NOD53   010416          4927#
NOD54   010420          4932#
NOD55   010424          4933#
NOD56   010442          4934#
NOD57   010446          4935#
NOD6    007752          4884#
NOD60   010466          4936#
NOD61   010472          4939#
NOD62   010476          4940#
NOD63   010502          4941#
NOD64   010506          4942#
NOD65   010512          4943#
NOD66   010516          4944#
NOD67   010522          4945#
NOD7    007754          4885#
```

```
NOD70   010526            4946#
NOD71   010532            4949#
NOD72   010536            4950#
NOD73   010542            4951#
NOD74   010554            4952#
NOD75   010560            4953#
NOD76   010574            4954#
NOD77   010600            4964#
NONE  = 000000            4300#
NOTNUF= 000050            4449#   4889    4892    4893    4903    4942    4944    4950    5024
NULEVT  015201            5337#   5996
NULL  = 000000            4409#
NUM   = 000014            4421#   5032
N10$    007712            4878    4879#
N100$   010342            4892    4895    4923#
N102$   010346            4924#
N104$   010372            4924    4926#
N110$   010420            4897    4932#
N111$   010424            4933#
N112$   010446            4933    4935#
N114$   010536            4949    4950#
N115$   010532            4906    4908    4910    4912    4914    4917    4919    4949#   4954    4965    4967    4988    5048
                          5050    5052    5054    5059
N116$   010554            4951    4952#
N117$   010600            4952    4953    4964#
N118$   010622            4964    4966#
N120$   010722            4934    4936    4994#
N121$   011166            4998    5000    5002    5004    5006    5008    5010    5016    5023#   5033
N122$   011216            5025    5027#
N123$   011172            5023    5024#
N125$   011414            4942    4944    4946    4950    5012    5024    5064#
N126$   011236            5026    5028    5031#
N130$   010642            4966    4983#
N131$   010702            4987    4990#
N132$   010662            4983    4987#
N140$   011252            4991    5036#
N141$   011256            5047#
N142$   011304            5047    5049#
N143$   011324            5049    5051#
N144$   011352            5051    5053#
N150$   011400            4984    5057#
N20$    010050            4891    4893#
N25$    010072            4894    4896#
N30$    010110            4893    4896    4898#   4902    4903    4904    4915    4918    4923    4926    4932    4935    4941
                          4943    4945    4990    4994    5014    5027    5031    5032    5036    5053    5057    5058
N40$    010030            4889    4891#
N42$    007720            4879    4881#
N43$    007736            4881    4883#
N44$    007754            4883    4885#
N45$    007772            4885    4887#
N46$    010010            4887    4889#
N50$    010472            4890    4939#
N51$    010476            4939    4940#
N52$    010502            4940    4941#
N60$    010726            4997#
N61$    010746            4997    4999#
```

```
N62$    010770          4999    5001#
N63$    011010          5001    5003#
N64$    011030          5003    5005#
N65$    011050          5005    5007#
N66$    011070          5007    5009#
N67$    011110          5009    5011#
N68$    011134          5011    5014#
N70$    011140          5015#   5019
N71$    011150          5015    5017#
N72$    011154          5017    5018#
N73$    011164          5018    5020#
N80$    010112          4888    4902#
N81$    010116          4903#
N82$    010160          4905    4907#
N83$    010202          4907    4909#
N84$    010224          4909    4911#
N85$    010246          4911    4913#
N86$    010276          4913    4915#
N87$    010324          4916    4918#
OFSET   006526          4719#   5504    7920*   7933*
OPBFPT  002514          4579#   8134
OPBUF   002520          4548    4580#   7448    8062    8066    8068    8074    8075    8087    8089    8117    8118    8135
                        8137
OPCNT   002166          4535#   7446*   8072*   8076    8077
OPEND   002642          4581#   8064
OPRMM   013566          5270#   8066
OPRMSG= 000015          4422#   5014
OPTYP   011436          5093#   6877*
OPVAR   006476          4703#   7595*   8003    8334*
OTINT = 000002          4365#
OUTVEC  011432          5091#   6874*   6875*   6893
O$APTS= 000000          4030#   4116
O$AU  = 000001          4030#   4088#   4116
O$BGNR= 000001          4030#   4088#   4116
O$BGNS= 000000          4030#   4116
O$DU  = 000001          4030#   4088#   4116
O$ERRT= 000000          4030#   4116
O$GNSW= 000000          4030#   4116
O$POIN= 000001          4030#   4088#   4116
O$SETU= 000000          4030#   4116    9059
PAD   = 001000          4388#   8303    8502    8751    8769    8814
PARAM   006570          4745#   5799    5851    7124*   7130    7395    7511*   7522*   7527*   7545*   7548*   7578    7660*
                        7794    7799    7848    7893    8061*   8115*   8613
PAS   = 000002          4294#   7492    7523    7819    7836
PASC  = 000042          4443#   5058
PASMOD= 000030          4433#   4907
PCADD   006554          4735#
PCK     013530          5251#   6397
PCLKCT= 001600          4311#   6813
PCLKEN= 000111          4310#   6815
PCPM    014352          5313#
PCSAR   011420          5084#   6864*   6865*   8281*
PEC     013540          5253#   6401
PLCK    030776          4759    7685#
PLCK2   030776          7686#
PLCK3   031012          7688#
```

```
PMS     013547            5255#  6422
PNCK    013526            5250#  6400
PNEC    013536            5252#  6404
PNMS    013545            5254#  6425
PNST    013515            5248#  6396
PNT   = 001000 G          4278#
PRI   = 002000 G          4278#
PRI00 = 000000 G          4278#  6895
PRI01 = 000040 G          4278#
PRI02 = 000100 G          4278#
PRI03 = 000140 G          4278#
PRI04 = 000200 G          4278#  6892   6893
PRI05 = 000240 G          4278#
PRI06 = 000300 G          4278#
PRI07 = 000340 G          4116   4278#  6952
PRNT  = 000055            4454#  4883   7164   7331
PROTO = 000041            4442#
PROTOB= 000040            4320#  7543
PSCNT   006500            4704#  7596*  8001*  8004
PST     013517            5249#  6393
PTREND  006430            4679#
PTRTAB  006144            4676#  6326   6331   7099   7192   7356   7363   7587
PTR13   006240            4677#  7103   7222   7340   7350   7588
PTR23   006334            4678#  7589
P$ACT   003132            4663#  5908*  6497   7148*
P$BUFA  003126            4661#  5906*  6452   7146*
P$CNT   003134            4664#  6589*  6596   6653*  6660*  6665
P$EXIT  023204            6455   6490#  6514   6519
P$GDBD  003143            4668#  5901*  5911   6512*  6517*  6525   6571*  6609*  7143*  7151   7430*  7479*  7526*
                          7567*
P$NNUF  003142            4667#  5902*  5916   7144*  7155   7312*  7336*  7344*  7358*  7406*  7485*  7493*  7512*
                          7570*
P$NUM   003136            4665#  6584*  6587*  6589   6590*  6592*  6596*  6598*  6604*  7399   7405   7427   7432
                          7434   7552
P$RADX  003140            4666#  6549*  6552*  6559*  6566   6594   6602
P$TREE  003130            4662#  5907*  6453   7147*
P$TRV   023066            5910   6451#  7150
P$TR5   023076            6454#  6463   6483   6488
QCOPY = 000013            4420#  5027   7422
QRX   = 000004            4366#  7639   7689   7716   7767   7789   7821   7828   7839   8121   8415   8418   8541
                          8696   8803   8828
QTX   = 000010            4367#  7663   7716   7777   7822   7830   7838   7847   8079   8091   8303   8309   8500
                          8502   8535   8547   8755   8814   8825
QUALFG  003012            4618#  5909*  7149*  7419*  7422*  7425   7517*  7520   7528*  7546   7549*
QUALVL  003014            4619#  7209*  7239*  7416*  7434*  7435*
RDATRY= 000200            4501#  8638   8641
RDSR    011422            5085#  6866*  6867*  8640
REC   = 000000            4292#
RECMOD= 000031            4434#  4909
REPLOG  020730            5969   5979#
REPORT  020302            5894#  6700   7332
RERR  = 100000            4513#  8655
RESET = 000001            4504#  8228
RESFLG  006560            4737#  6787*  6799*  7091
RESTRT  024462            6791   6807   6816   6824   6828#
RHDMCC  002660            4606#  8297   8299   8320   8322   8329   8692   8693
```

```
RHDMSG  002656          4604#   8429    8805
RHLPEN  003046          4634#   5959
RHLPTB  003036          4630#   5957
RHLP0   013034          5220#   5899
RHLP1   013072          4630    5221#
RHLP2   013114          4631    5222#
RHLP3   013147          4632    5223#
RHLP4   013200          4633    5224#
RI    = 040000          4483#   4800
RINTEN= 000100          4500#   8431    8698    8809
RMDLOP= 000047          4448#   5053
RMSGCC  011462          5113#   8423*   8430*   8651*   8668*   8692*   8806*
RMSGPT  011460          5112#   8422*   8429*   8646    8648*   8669*   8691*   8805*
RNODE   011472          5117#   6878*   7603    8282    8419    8504
RNODM   035514          8977    8999#
RPASS   006572          4753#   5798    7123*   7129    7135    7337*   7394    7552*   8008    8010*
RPEXT = 000002          4460#   5921    5935    5965
RPHLP = 000001          4459#   5931    5933    5961
RPLOG = 000003          4461#   5937    5970
RPT     020776          5999#   6027    6034    6045    6051    6058    6066    6089
RPTAA   021046          6014#
RPTDCK  021512          4847    6061#
RPTDDE  021436          4850    6054#
RPTDER  021222          4845    6029#
RPTDLE  021512          4849    6060#
RPTDSP  007650          4841#   6020
RPTDVI  021316          4846    6036#
RPTEOP  021362          4851    6046#
RPTMSB  021666          6026    6085    6088    6101#
RPTMSC  021566          4848    6082#
RPTMSE  021652          6089#
RPTTXQ  021150          4841    4842    4843    4844    6022#
RPT0    021042          5995    6006    6010    6013#
RPT1    021030          6003    6009#
RSTARY= 002000          4502#   8638    8674
RTS   = 000004          4482#   4799    8866    8903
RUN   = 000004          4413#   4887    7166    7335
RXACT = 004000          4503#
RXBUF   004144          4674#   7611
RXC   = 000006          4343#   5786    5849
RXCSR   011416          5083#   6861*   8230    8238    8262*   8271    8278    8373    8431*   8528    8612    8698*   8809*
                        8866*   8869    8877    8903*
RXENA = 000020          4498#   8431    8698    8809
RXINEX  034624          8639    8652    8670    8675    8694    8700#
RXIN1   034272          8630    8633#
RXIN2   034220          8622    8624#
RXIN21  034304          8614    8616    8618    8638#
RXIN3   034464          8642    8674#
RXIN4   034332          8646#
RXIN5   034540          8654    8687#
RXIN6   034440          8657    8667#
RXIN7   034610          8685    8688    8696#
RXIN8   034616          8665    8698#
RXM   = 000040          4384#   8289    8418    8421    8687    8689
RXMTOT  006472          4700#   7591*   7638    7714    7897
RXONLY  030704          4757    7636#
```

```
RXON2   030704            7637#
RXPTR   006434            4683#   7101*   7589*   7612    7637    7688    7715    7802    7895
RXQ   = 000004            4342#   5782
R10$    020526            5930    5931#
R11$    020534            5931    5933#
R12$    020552            5933    5935#
R125$   020606            5940#
R13$    020570            5935    5937#
R30$    020604            5937    5939#
SCM     016275            5383#   5803
SCMD    016330            5386#   5811
SCML    016317            5385#   5807
SDVE    016264            5382#   5789
SDVI    016306            5384#   5794
SEOP    016341            5387#   5815
SETET = 000060            4457#   5011    7160    7482
SETEXP= 000010            4417#   4933    7182    7412    7477
SETTRN= 000011            4418#   4935    7415
SHF0    014211            5303#   6390
SHF1    014247            5310#   6428
SHMSG   013232            5226#   7388
SHOW  = 000002            4411#   4894    7180    7318    7345    7359
SHTAB   003066            4641#   7379    7385
SHTEND  003075            4642#   7382
SHTYP0  013266            4636    5227#
SHTYP1  013275            4636    5228#
SHTYP2  013302            4636    5229#
SHTYP3  013307            4636    5230#
SHTYP4  013314            4636    5231#
SHTYP5  013322            4636    5232#
SHTYP6  013327            4636    5233#
SHTYP7  013335            4636    5234#
SHTYTB  003046            4636#   7388
SHWOP   022564            6043    6379#   7131    7396
SIZE  = 000012            4419#   5025    7419    7425
SMSC    016352            5396#   5834
SQD   = 000040            4484#   4801
SRXQ    016253            5381#   5781
STADD   006504            4706#   6148    7399*
START   024262            6789    6799#
STATB = 000001            4315#   5851    6394    7534
STATUS= 000016            4423#   4964
STXC    016242            5380#   5776
STXQ    016231            5379#   5771
SVCGBL= 000000            4030#   4039#   4116    4136    4154    5165    5176    5503    5507    5511    5515    5534    5543
                          5707    6686    6727    6756    6923    6941    6980    7017    8611    8744    8944    9059#
SVCINS= 000001            4030#   4036#   4116    4136    4154    5165    5176    5504    5505    5508    5509    5512    5513
                          5516    5517    5535    5536    5544    5545    5547    5731    5859    5862    5899    5905    5913
                          5918    5958    5996    6014    6019    6025    6032    6033    6049    6057    6064    6083    6086
                          6101    6116    6150    6154    6156    6196    6390    6428    6570    6608    6718    6785    6788
                          6789    6790    6791    6792    6793    6795    6796    6803    6804    6809    6810    6818    6819
                          6823    6826    6843    6844    6880    6892    6893    6895    6896    6912    6932    6952    6953
                          6955    6971    6991    7007    7028    7044    7089    7126    7133    7134    7137    7145    7153
                          7157    7176    7187    7196    7216    7226    7322    7388    7429    7454    7525    7569    7582
                          7916    7930    7939    7979    8037    8066    8116    8134    8232    8242    8275    8277    8311
                          8333    8521    8532    8663    8684    8701    8772    8822    8862    8868    8881    8921    8944
```

```
                    8958    8975    8976    8977    8979    9059
SVCSUB= 000001      4030#   4038#
SVCTAG= 000001      4030#   4040#   4200    5505    5509    5513    5517    5536    5545    5731    5905    6718    6823
                    6912    6932    6971    7007    7044    7145    8066    8701    8772    8921    8979
SVCTST= 000001      4030#   4037#   7070
SYN   = 000226      4514#
SYNCC   011454      5110#   8307*   8516*   8745    8748*   8817*
SYNCW   011456      5111#   7601*   7606*   8747
S$LSYM= 010000      4030#   4200#   5505#   5509#   5513#   5517#   5536#   5545#   5731#   5905#   6718#   6823#   6912#
                    6932#   6971#   7007#   7044#   7145#   8066#   8701#   8772#   8921#   8979#
S1      024250      6793    6795#
S2      024324      6804    6809#
S3      024374      6810    6818#
S4      024442      6819    6826#
TABEX   014017      5299#   7196    7226
TAL   = 000005      4297#   7509    8139
TALCK   032242      4762    8060#   8088    8090
TALMOD= 000035      4438#   4918
TBMT  = 000004      4506#
TCURAD  006462      4695#   7107*   7110    7202    7207*   7366*
TDSR    011426      5087#   6870*   6871*   8747*   8750*   8753*   8758*   8762*
TEMP    006530      4720#   5772*   5777*   5782*   5786*   5790*   5795*   5804*   5808*   5812*   5816*   5835*   5849
                    5865    5866*   5867*   5868    6153*   6154    6198*   6199*   6200    6239*   6240*   6246    6381*
                    6390    6393*   6396*   6428    7387*   7388    7439*   7440*   7444    7447    7808*   7809*   7813
                    8860*   8861*   8863
TEMP1   006532      4721#   5771*   5776*   5781*   5789*   5794*   5803*   5807*   5811*   5815*   5834*   5862    6388*
                    6390    6397*   6400*   6428
TEMP2   006534      4722#   5796*   5797*   5871    6242*   6244*   6248    6389*   6390    6401*   6404*   6428    7770*
                    7780*   7791*   7797    7811*   7812    7832*   7901*   8004*   8075*   8084*   8118*   8129*   8237*
                    8270*   8328*   8527*   8629*   8632*   8660*   8679*   8876*   8899*   8900*
TEMP3   006536      4723#   5512    5535    5544    5798*   5872    6384*   6387*   6390    7772*   7782*   7792*   7798
                    7809    7810    7811    7833*   7903*   7913    8005*   8076*   8085*   8120*   8130*   8238*   8271*
                    8329*   8528*   8624*   8626    8658*   8680*   8877*
TEMP4   006540      4724#   5508    5535    5544    5799*   5846*   5873    7910*   7919*   7924*   7925    7936    8003*
                    8239*   8272*   8330*   8529*   8625*   8626    8659*   8681*   8878*
TEMP5   006542      4725#   6422*   6425*   6428
TEOM  = 001000      4511#   8750    8762
TERR  = 100000      4512#
TIMERS  006640      4779#   5725    5729*   8290*   8294    8312    8318    8519*   8522    8823
TIMER1  006634      4777#   5719    5721*   7084*   7085    8264*   8265    8867*   8871
TIMER2  006636      4778#   5722    5724*
TIMMIN  006626      4773#   5716*   5870    6828*
TIMSEC  006630      4774#   5713*   5714    5717*   5869    6829*
TIMTCK  006632      4775#   5710*   5712*   5727    5867    6830*
TINTEN= 000100      4509#   8308    8517    8754    8818
TM    = 000040      4485#   4802    8621    8623
TOINOT  034050      8523    8535#
TOTCC   006524      4718#   6190*   6191    6197*   6199    6201*   7096*   7184*   7185    7206    7213*   7214    7238
TRA   = 000001      4293#   7506
TRAMOD= 000034      4437#   4916
TRVACT  023206      6484    6495#   6511    6516    6521    6524    6544    6605    6623    6644    6668
TRVALN  024000      6473    6627#
TRVALP  023734      6472    6613#
TRVBIF  023312      6469    6524#
TRVBR   023302      6468    6521#
TRVBRC  023226      6482    6502#   6522    6527    6546    6610    6625    6646    6672
```

```
TRVDEC  023406            6475    6549#
TRVERR  023244            6466    6511#
TRVEXI  023264            6467    6516#
TRVNMA  023426            6550    6553#
TRVNOB  023236            6507#   6528    6545    6606    6624    6645
TRVNUM  023420            6471    6552#
TRVOCT  023420            6474    6551#
TRVSPA  023334            6470    6530#
TRVSTR  024066            6476    6650#
TSOM  = 000400            4510#   8750
TTL   = 000001            4301#   7138    8249
TTLL  = 000010            4507#   8248    8251
TTLLOP= 000044            4445#   5047
TTOTCC  006460            4694#   6350    709?*   7184    7189    7206*   7362*
TXACT = 000002            4505#
TXBUF   003144            4673#   6318    7107    7365
TXC   = 000002            4341#   5777
TXCSR   011424            5086#   6868*   6869*   8228*   8239    8248*   8251*   8272    8308*   8375    8517*   8529    8621
                          8754*   8818*   8878
TXENA = 000020            4508#   8308    8517    8818
TXINEX  035004            8749    8756    8761    8768    8771#
TXIN1   034650            8746    8750#
TXIN2   034712            8752    8757#
TXIN3   034776            8764    8769#
TXM   = 000020            4382#   8289    8502    8506    8763    8765
TXMTOT  006456            4693#   6338    7113*   7191*   7193    7208*   7357    7361*   7580    7662    7686    7712
TXONLY  030736            4758    7660#
TXON2   030744            7661#
TXPTR   006436            4684#   6331*   6333*   6334    6343*   6345    7100*   7111    7192*   7200*   7201    7205*   7363*
                          7364    7587*   7661    7687    7713
TXQ   = 000000            4340#   5772
T$ARGC= 000001            4116#   5504#   5508#   5512#   5516#   5535#   5544#   5859#   5862#   5899#   5913#   5918#   5958#
                          5996#   6014#   6019#   6025#   6032#   6033#   6049#   6057#   6064#   6083#   6086#   6101#   6116#
                          6150#   6154#   6156#   6196#   6390#   6428#   6570#   6608#   6826#   7089#   7126#   7153#   7157#
                          7187#   7196#   7216#   7226#   7322#   7388#   7429#   7454#   7525#   7569#   7582#   8037#   8116#
                          8134#
T$CODE= 006130            5905#   6823#   7145#   8066#   8958#   8975#   8976#   8977#
T$ERRN= 000006            4030#   7916#   7930#   7939#   7979#   8242#   8275#   8333#   8532#   8663#   8684#   8881#
T$EXCP= 000000            5905#   6823#   7145#   8066#   8975#   8976#
T$FLAG= 000040            5547#   6896#   6955#   6991#   7028#   7137#   7176#
T$GMAN= 000000            4030#   5905#   6823#   7145#   8066#
T$HILI= 000776            5905#   6823#   7145#   8066#   8975#   8976#
T$LAST= 000001            4030#   9059#
T$LOLI= 000300            5905#   6823#   7145#   8066#   8975#   8976#
T$LSYM= 010000            4030#   4200    5505    5509    5513    5517    5536    5545    5731    6718    6912    6932    6971
                          7007    7044    8701    8772    8921    8979
T$LTNO= 000001            9059#
T$NEST= 177777            4030#   4051#   4154#   4200#   5503#   5505#   5507#   5509#   5511#   5513#   5515#   5517#   5534#
                          5536#   5543#   5545#   5707#   5731#   6686#   6718#   6727#   6733#   6756#   6912#   6923#   6932#
                          6941#   6971#   6980#   7007#   7017#   7044#   7070#   8611#   8701#   8744#   8772#   8921#   8944#
                          8979#   9060#
T$NS0 = 000000            4051#   9060
T$NS1 = 000004            4154#   4200    5503#   5505    5507#   5509    5511#   5513    5515#   5517    5534#   5536    5543#
                          5545    5707#   5731    6686#   6718    6727#   6733    6756#   6912    6923#   6932    6941#   6971
                          6980#   7007    7017#   7044    7070#   8921    8944#   8979
T$NS2 = 000010            8611#   8701    8744#   8772
```

```
T$PTNU= 000000          4030#
T$SAVL= 177777          4030#
T$SEGL= 177777          4030#
T$SUBN= 000000          4030#   7070#
T$TAGL= 177777          4030#
T$TAGN= 010023          4030#   4154#   5503#   5507#   5511#   5515#   5534#   5543#   5707#   6686#   6727#   6756#   6923#
                        6941#   6980#   7017#   7070#   8611#   8744#   8944#
T$TEMP= 000000          4136#   4200#   5505#   5509#   5513#   5517#   5536#   5545#   5547#   5731#   5905#   6718#   6733#
                        6823#   6896#   6912#   6932#   6955#   6971#   6991#   7007#   7028#   7044#   7137#   7145#   7176#
                        8066#   8701#   8772#   8921#   8958#   8975#   8976#   8977#   8979#   9060#
T$TEST= 000001          4030#   7070#   9059
T$TSTM= 177777          4030#   5504    5505    5508    5509    5512    5513    5516    5517    5535    5536    5544    5545
                        5859    5862    5899    5905    5913    5918    5958    5996    6014    6019    6025    6032    6033
                        6049    6057    6064    6083    6086    6101    6116    6150    6154    6156    6196    6390    6428
                        6570    6608    6718    6785    6788    6790    6792    6795    6803    6809    6818    6823    6826
                        6843    6880    6892    6893    6895    6896    6912    6932    6952    6953    6955    6971    7007
                        7044    7089    7126    7133    7137    7145    7153    7157    7176    7187    7196    7216    7226
                        7322    7388    7429    7454    7525    7569    7582    7916    7930    7939    7979    8037    8066
                        8116    8134    8232    8242    8275    8277    8311    8333    8521    8532    8663    8684    8822
                        8862    8868    8881    8921
T$TSTS= 000001          4030#   7070#
T$$AU = 010016          7017#   7028    7044
T$$AUT= 010013          6923#   6932
T$$CLE= 010014          6941#   6955    6971
T$$DU = 010015          6980#   6991    7007
T$$HAR= 010022          8944#   8979
T$$HW = 010000          4154#   4200
T$$INI= 010012          6756#   6896    6912
T$$MSG= 010006          5503#   5505    5507#   5509    5511#   5513    5515#   5517    5534#   5536    5543#   5545    5547
T$$PRO= 010011          6727#
T$$RPT= 010010          6686#   6718
T$$SRV= 010021          5707#   5731    8611#   8701    8744#   8772
T$$TES= 010017          7070#   7137    7176    8921
T1      025072  G       4136    7070#
UAM   = 000200  G       4278#
UPTABL  031244          7794#
UPTA1   031332          7800    7807#
UPTA3   031330          7804    7806#
UPTA4   031270          7795    7799#
UPTEX   031402          7806    7818#
VECTOR  035461          8976    8998#
X$    = 000202          4054#   4878#   4879#   4880#   4881#   4882#   4883#   4884#   4885#   4886#   4887#   4888#   4889#
                        4890#   4891#   4892#   4893#   4894#   4895#   4896#   4897#   4898#   4902#   4903#   4904#   4905#
                        4906#   4907#   4908#   4909#   4910#   4911#   4912#   4913#   4914#   4915#   4916#   4917#   4918#
                        4919#   4923#   4924#   4925#   4926#   4927#   4932#   4933#   4934#   4935#   4936#   4939#   4940#
                        4941#   4942#   4943#   4944#   4945#   4946#   4949#   4950#   4951#   4952#   4953#   4954#   4964#
                        4965#   4966#   4967#   4983#   4984#   4987#   4988#   4990#   4991#   4994#   4997#   4998#   4999#
                        5000#   5001#   5002#   5003#   5004#   5005#   5006#   5007#   5008#   5009#   5010#   5011#   5012#
                        5014#   5015#   5016#   5017#   5018#   5019#   5020#   5023#   5024#   5025#   5026#   5027#   5028#
                        5031#   5032#   5033#   5036#   5047#   5048#   5049#   5050#   5051#   5052#   5053#   5054#   5057#
                        5058#   5059#   5064#   5930#   5931#   5932#   5933#   5934#   5935#   5936#   5937#   5938#   5939#
                        5940#
X$ALWA= 000000          4030#
X$FALS= 000040          4030#
X$OFFS= 000400          4030#
X$TRUE= 000020          4030#
```

```
$PATCH  035540          9049#
.     = 035624          4026#  4574#  4580#  4588#  4601#  4610#  4615#  4643#  4673#  4674#  4675#  4676#  4677#
                        4678#  4681#  4784#  4785#  4881#  4885#  4889#  4896#  4903#  4905#  4911#  4913#  4924#
                        4926#  4933#  4935#  4951#  4953#  4964#  4966#  4983#  4990#  4997#  4999#  5001#  5003#
                        5005#  5011#  5025#  5027#  5051#  5165#  5272#  5547   5933#  5935#  6896   6955   6991
                        7028   7137   7176   9050#
```

```
BCOMPL    15#  4030#  6789  6791  6796  7134
BERROR    19#  4030#
BGNAU     23#  4030#  7017
BGNAUT    31#  4030#  6923
BGNCLN    39#  4030#  6941
BGNDU     47#  4030#  6980
BGNHRD    55#  4030#  8944
BGNHW     66#  4030#  4154
BGNINI    77#  4030#  6756
BGNMOD    85#  4030#  4051
BGNMSG    98#  4030#  5503  5507  5511  5515  5534  5543
BGNPRO   106#  4030#  6727
BGNPTA   114#  4030#
BGNRPT   144#  4030#  6686
BGNSEG   152#  4030#
BGNSET   161#  4030#
BGNSFT   182#  4030#
BGNSRV   193#  4030#  5707  8611  8744
BGNSUB   201#  4030#
BGNSW    225#  4030#
BGNTST   236#  4030#  7070
BNCOMP   266#  4030#  6793  6804  6810  6819  6844
BNERRO   270#  4030#
BREAK    274#  4030#  8232  8277  8311  8521  8822  8862  8868
BRESET   278#  4030#  6953
CKLOOP   282#  4030#
CLI     4058#  4878   4879  4880  4881  4882  4883  4884  4885  4886  4887  4888  4889  4890  4891
        4892   4893   4894  4895  4896  4897  4898  4902  4903  4904  4905  4906  4907  4908  4909
        4910   4911   4912  4913  4914  4915  4916  4917  4918  4919  4923  4924  4925  4926  4927
        4932   4933   4934  4935  4936  4939  4940  4941  4942  4943  4944  4945  4946  4949  4950
        4951   4952   4953  4954  4964  4965  4966  4967  4983  4984  4987  4988  4990  4991  4994
        4997   4998   4999  5000  50C1  5002  5003  5004  5005  5006  5007  5008  5009  5010  5011
        5012   5014   5015  5016  5017  5018  5019  5020  5023  5024  5025  5026  5027  5028  5031
        5032   5033   5036  5047  5048  5049  5050  5051  5052  5053  5054  5057  5058  5059  5064
        5930   5931   5932  5933  5934  5935  5936  5937  5938  5939  5940
CLOCK    286#  4030#  6803  6809
CLOSE    292#  4030#
CLRVEC   296#  4030#
COMMEN   301#  4030#
DELAY    322#  4030#
DESCRI   317#  4030#  5176
DEVTYP   341#  4030#  5165
DISPAT   346#  4030#  4136
DISPLA   360#  4030#
DOCLN    376#  4030#  6785
DODU     380#  4030#
DORPT    385#  4030#
ENDAU    389#  4030#  7044
ENDAUT   401#  4030#  6932
ENDCLN   413#  4030#  6971
ENDCOM   425#  4030#
ENDDU    441#  4030#  7007
ENDHRD   453#  4030#  8979
ENDHW    465#  4030#  4200
ENDINI   475#  4030#  6912
ENDMOD   487#  4030#  9060
```

```
ENDMSG  500#  4030#  5505  5509  5513  5517  5536  5545
ENDPRO  512#  4030#  6733
ENDPTA  520#  4030#
ENDRPT  529#  4030#  6718
ENDSEG  541#  4030#
ENDSET  555#  4030#
ENDSFT  568#  4030#
ENDSRV  580#  4030#  5731  8701  8772
ENDSUB  596#  4030#
ENDSW   614#  4030#
ENDTST  624#  4030#  8921
EQUALS  642#  4030#  4278
ERRDF   714#  4030#
ERRHRD  718#  4030#
ERROR   722#  4030#
ERRSF   726#  4030#
ERRSOF  730#  4030#  7916  7930  7939  7979  8242  8275  8333  8532  8663  8684  8881
ERRTBL  734#  4030#
ESCAPE  744#  4030#
EXIT    771#  4030#  5547  6896  6955  6991  7028  7137  7176
FEQUAL  810#  4030#
GETBYT  824#  4030#
GETPRI  834#  4030#
GETWOR  829#  4030#
GMANIA  839#  4030#
GMANID  848#  4030#  5905  6823  7145  8066
GMANIL  859#  4030#
GPHARD  868#  4030#  6843
GPRMA   874#  4030#  8975  8976
GPRMD   903#  4030#  5905#  6823#  7145#  8066#
GPRML   934#  4030#  8958  8977
HEADER  954#  4030#  4116
INLOOP  962#  4030#
IOSETU  966#  4030#
IOSTAR  974#  4030#
KT11    982#  4030#
LASTAD  1147#  4030#  9059
MANUAL  1162#  4030#  7133
MEMORY  1166#  4030#
M$BYTE  2000#  4030#  4116#
M$CHEC  2118#  4030#  5547#  6896#  6955#  6991#  7028#  7137#  7176#
M$CNTO  2182#  4030#  5905#  6823#  7145#  8066#  8958#  8975#  8976#  8977#
M$COUN  2066#  4030#  5504#  5508#  5512#  5516#  5535#  5544#  5859#  5862#  5899#  5913#  5918#  5958#  5996#
        6014#  6019#  6025#  6032#  6033#  6049#  6057#  6064#  6083#  6086#  6101#  6116#  6150#  6154#  6156#
        6196#  6390#  6428#  6570#  6608#  6826#  7089#  7126#  7153#  7157#  7187#  7196#  7216#  7226#  7322#
        7388#  7429#  7454#  7525#  7569#  7582#  8037#  8116#  8134#
M$DATA  1867#  4030#  4116#  5165#  5176#
M$DECR  2029#  4030#  4200#  5505#  5509#  5513#  5517#  5536#  5545#  5731#  6718#  6733#  6912#  6932#  6971#
        7007#  7044#  8701#  8772#  8921#  8979#  9060#
M$DEFA  2170#  4030#  5905#  6823#  7145#  8066#  8958#  8975#  8976#  8977#
M$ENDE  2074#  4030#  4200#  5505#  5509#  5513#  5517#  5536#  5545#  5731#  6718#  6912#  6932#  6971#  7007#
        7044#  8701#  8772#  8921#  8979#  9060#
M$ERRI  1649#  4030#  7916#  7930#  7939#  7979#  8242#  8275#  8333#  8532#  8663#  8684#  8881#
M$ESCA  2006#  4030#
M$ESCS  2010#  4030#
M$EXCP  2101#  4030#  5905#  6823#  7145#  8066#  8975#  8976#
```

CROSS REFERENCE TABLE -- MACRO NAMES

```
M$EXIT   2014#  4030#  5547#  6896#  6955#  6991#  7028#  7137#  7176#
M$EXSE   2022#  4030#  5547#  6896#  6955#  6991#  7028#  7137#  7176#
M$EXTJ   2018#  4030#  5547#  6896#  6955#  6991#  7028#  7137#  7176#
M$GEN    2038#  4030#  4116#  4136#  4154#  4200#  5165#  5176#  5503#  5505#  5507#  5509#  5511#  5513#  5515#
         5517#  5534#  5536#  5543#  5545#  5707#  5731#  5905#  6686#  6718#  6727#  6756#  6823#  6912#  6923#
         6932#  6941#  6971#  6980#  7007#  7017#  7044#  7070#  7145#  8066#  8611#  8701#  8744#  8772#  8921#
         8944#  8979#  9059#
M$GENB   1938#  4030#  5905#  6823#  7145#  8066#
M$GETS   2035#  4030#  4200#  5505#  5509#  5513#  5517#  5536#  5545#  5731#  6718#  6733#  6912#  6932#  6971#
         7007#  7044#  8701#  8772#  8921#  8979#  9060#
M$GETT   1877#  4030#  5547#  6896#  6955#  6991#  7028#  7137#  7176#
M$GNGB   1902#  4030#  4051#  4116#  4136#  4154#  5165#  5176#  5503#  5507#  5511#  5515#  5534#  5543#  5707#
         6686#  6727#  6756#  6923#  6941#  6980#  7017#  8611#  8744#  8944#  9059#
M$GNIN   2049#  4030#  4116#  4136#  4154#  5165#  5176#  5504#  5505#  5508#  5509#  5512#  5513#  5516#  5517#
         5535#  5536#  5544#  5545#  5547#  5731#  5859#  5862#  5899#  5905#  5913#  5918#  5958#  5996#  6014#
         6019#  6025#  6032#  6033#  6049#  6057#  6064#  6083#  6086#  6101#  6116#  6150#  6154#  6156#  6196#
         6390#  6428#  6570#  6608#  6718#  6785#  6788#  6789#  6790#  6791#  6792#  6793#  6795#  6796#  6803#
         6804#  6809#  6810#  6818#  6819#  6823#  6826#  6843#  6844#  6880#  6892#  6893#  6895#  6896#  6912#
         6932#  6952#  6953#  6955#  6971#  6991#  7007#  7028#  7044#  7089#  7126#  7133#  7134#  7137#  7145#
         7153#  7157#  7176#  7187#  7196#  7216#  7226#  7322#  7388#  7429#  7454#  7525#  7569#  7582#  7916#
         7930#  7939#  7979#  8037#  8066#  8116#  8134#  8232#  8242#  8275#  8277#  8311#  8333#  8521#  8532#
         8663#  8684#  8701#  8772#  8822#  8862#  8868#  8881#  8921#  8944#  8958#  8975#  8976#  8977#  8979#
         9059#
M$GNLS   1913#  4030#  5905#  6823#  7145#  8066#
M$GNSU   1898#  4030#
M$GNTA   1890#  4030#  4200#  5505#  5509#  5513#  5517#  5536#  5545#  5731#  6718#  6912#  6932#  6971#  7007#
         7044#  8701#  8772#  8921#  8979#
M$GNTE   1894#  4030#  7070#
M$HAPT   1739#  4030#  4116#
M$HNAP   1824#  4030#  4116#
M$INCR   2026#  4030#  4051#  4154#  5503#  5504#  5505#  5507#  5508#  5509#  5511#  5512#  5513#  5515#  5516#
         5517#  5534#  5535#  5536#  5543#  5544#  5545#  5707#  5859#  5862#  5899#  5905#  5913#  5918#  5958#
         5996#  6014#  6019#  6025#  6032#  6033#  6049#  6057#  6064#  6083#  6086#  6101#  6116#  6150#  6154#
         6156#  6196#  6390#  6428#  6570#  6608#  6686#  6718#  6727#  6756#  6785#  6788#  6790#  6792#  6795#
         6803#  6809#  6818#  6823#  6826#  6843#  6880#  6892#  6893#  6895#  6896#  6912#  6923#  6932#  6941#
         6952#  6953#  6955#  6971#  6980#  7007#  7017#  7044#  7070#  7089#  7126#  7133#  7137#  7145#  7153#
         7157#  7176#  7187#  7196#  7216#  7226#  7322#  7388#  7429#  7454#  7525#  7569#  7582#  7916#  7930#
         7939#  7979#  8037#  8066#  8116#  8134#  8232#  8242#  8275#  8277#  8311#  8333#  8521#  8532#  8611#
         8663#  8684#  8744#  8822#  8862#  8868#  8881#  8921#  8944#
M$IOSE   1700#  4030#
M$LDR0   1942#  4030#  6788#  6790#  6792#  6795#  6803#  6809#  6843#  6895#  6952#
M$MASK   1671#  4030#
M$MCHI      4#  4030#
M$MCLO   1624#  4030#
M$MSK1   1677#  4030#
M$POP    1881#  4030#  4200#  5505#  5509#  5513#  5517#  5536#  5545#  5731#  6718#  6733#  6912#  6932#  6971#
         7007#  7044#  8701#  8772#  8921#  8979#  9060#
M$PRIN   1636#  4030#  5504#  5508#  5512#  5516#  5535#  5544#  5859#  5862#  5899#  5913#  5918#  5958#  5996#
         6014#  6019#  6025#  6032#  6033#  6049#  6057#  6064#  6083#  6086#  6101#  6116#  6150#  6154#  6156#
         6196#  6390#  6428#  6570#  6608#  6826#  7089#  7126#  7153#  7157#  7187#  7196#  7216#  7226#  7322#
         7388#  7429#  7454#  7525#  7569#  7582#  8037#  8116#  8134#
M$PUSH   1631#  4030#  4051#  4154#  5503#  5507#  5511#  5515#  5534#  5543#  5707#  6686#  6727#  6756#  6923#
         6941#  6980#  7017#  7070#  8611#  8744#  8944#
M$PUT    1972#  4030#  5504#  5508#  5512#  5516#  5535#  5544#  5859#  5862#  5899#  5913#  5918#  5958#  5996#
         6014#  6019#  6025#  6032#  6033#  6049#  6057#  6064#  6083#  6086#  6101#  6116#  6150#  6154#  6156#
         6196#  6390#  6428#  6570#  6608#  6826#  6880#  6892#  6893#  7089#  7126#  7153#  7157#  7187#  7196#
```

```
        7216#  7226#  7322#  7388#  7429#  7454#  7525#  7569#  7582#  8037#  8116#  8134#
M$PUT1  1981#  4030#  5504#  5508#  5512#  5516#  5535#  5544#  5859#  5862#  5899#  5913#  5918#  5958#  5996#
        6014#  6019#  6025#  6032#  6033#  6049#  6057#  6064#  6083#  6086#  6101#  6116#  6150#  6154#  6156#
        6196#  6390#  6428#  6570#  6608#  6826#  6880#  6892#  6893#  7089#  7126#  7153#  7157#  7187#  7196#
        7216#  7226#  7322#  7388#  7429#  7454#  7525#  7569#  7582#  8037#  8116#  8134#
M$RADI  2077#  4030#  5905#  6823#  7145#  8066#  8958#  8975#  8976#  8977#
M$RBR0  1952#  4030#
M$RNR0  1962#  4030#  6803#  6809#  6843#
M$SETS  2032#  4030#  4051#  4154#  5503#  5507#  5511#  5515#  5534#  5543#  5707#  6686#  6727#  6756#  6923#
        6941#  6980#  7017#  7070#  8611#  8744#  8944#
M$STAR  1733#  4030#
M$SVC   1933#  4030#  5504#  5505#  5508#  5509#  5512#  5513#  5516#  5517#  5535#  5536#  5544#  5545#  5547#
        5859#  5862#  5899#  5905#  5913#  5918#  5958#  5996#  6014#  6019#  6025#  6032#  6033#  6049#  6057#
        6064#  6083#  6086#  6101#  6116#  6150#  6154#  6156#  6196#  6390#  6428#  6570#  6608#  6718#  6785#
        6788#  6790#  6792#  6795#  6803#  6809#  6818#  6823#  6826#  6843#  6880#  6892#  6893#  6895#  6896#
        6912#  6932#  6952#  6953#  6955#  6971#  6991#  7007#  7028#  7044#  7089#  7126#  7133#  7137#  7145#
        7153#  7157#  7176#  7187#  7196#  7216#  7226#  7322#  7388#  7429#  7454#  7525#  7569#  7582#  7916
        7930   7939   7979   8037#  8066#  8116#  8134#  8232#  8242   8275   8277#  8311#  8333   8521#  8532
        8663   8684   8822#  8862#  8868#  8881   8921#
M$TLAB  1929#  4030#  5504#  5505#  5508#  5509#  5512#  5513#  5516#  5517#  5535#  5536#  5544#  5545#  5859#
        5862#  5899#  5905#  5913#  5918#  5958#  5996#  6014#  6019#  6025#  6032#  6033#  6049#  6057#  6064#
        6083#  6086#  6101#  6116#  6150#  6154#  6156#  6196#  6390#  6428#  6570#  6608#  6718#  6785#  6788#
        6790#  6792#  6795#  6803#  6809#  6818#  6823#  6826#  6843#  6880#  6892#  6893#  6895#  6896#  6912#
        6932#  6952#  6953#  6955#  6971#  7007#  7044#  7089#  7126#  7133#  7137#  7145#  7153#  7157#  7176#
        7187#  7196#  7216#  7226#  7322#  7388#  7429#  7454#  7525#  7569#  7582#  7916#  7930#  7939#  7979#
        8037#  8066#  8116#  8134#  8232#  8242#  8275#  8277#  8311#  8333#  8521#  8532#  8663#  8684#  8822#
        8862#  8868#  8881#  8921#
M$TSTL  1921#  4030#  5504#  5505#  5508#  5509#  5512#  5513#  5516#  5517#  5535#  5536#  5544#  5545#  5859#
        5862#  5899#  5905#  5913#  5918#  5958#  5996#  6014#  6019#  6025#  6032#  6033#  6049#  6057#  6064#
        6083#  6086#  6101#  6116#  6150#  6154#  6156#  6196#  6390#  6428#  6570#  6608#  6718#  6785#  6788#
        6790#  6792#  6795#  6803#  6809#  6818#  6823#  6826#  6843#  6880#  6892#  6893#  6895#  6896#  6912#
        6932#  6952#  6953#  6955#  6971#  7007#  7044#  7089#  7126#  7133#  7137#  7145#  7153#  7157#  7176#
        7187#  7196#  7216#  7226#  7322#  7388#  7429#  7454#  7525#  7569#  7582#  7916#  7930#  7939#  7979#
        8037#  8066#  8116#  8134#  8232#  8242#  8275#  8277#  8311#  8333#  8521#  8532#  8663#  8684#  8822#
        8862#  8868#  8881#  8921#
M$WORD  1994#  4030#  4116#  4136#  5547#  5905#  6823#  6896#  6955#  6991#  7028#  7137#  7145#  7176#  7916#
        7930#  7939#  7979#  8066#  8242#  8275#  8333#  8532#  8663#  8684#  8881#  8958#  8975#  8976#  8977#
        9059
M$XFER  1682#  4030#
NODCL   4063#  4878   4879   4880   4881   4882   4883   4884   4885   4886   4887   4888   4889   4890   4891
        4892   4893   4894   4895   4896   4897   4898   4902   4903   4904   4905   4906   4907   4908   4909
        4910   4911   4912   4913   4914   4915   4916   4917   4918   4919   4923   4924   4925   4926   4927
        4932   4933   4934   4935   4936   4939   4940   4941   4942   4943   4944   4945   4946   4949   4950
        4951   4952   4953   4954   4964   4965   4966   4967   4983   4984   4987   4988   4990   4991   4994
        4997   4998   4999   5000   5001   5002   5003   5004   5005   5006   5007   5008   5009   5010   5011
        5012   5014   5015   5016   5017   5018   5019   5020   5023   5024   5025   5026   5027   5028   5031
        5032   5033   5036   5047   5048   5049   5050   5051   5052   5053   5054   5057   5058   5059   5064
        5930   5931   5932   5933   5934   5935   5936   5937   5938   5939   5940
OPEN    1171#  4030#
POINTE  1176#  4030#  4088
PRINTB  1239#  4030#  5504   5508   5512   5516   5535   5544
PRINTF  1279#  4030#  5859   5862   5899   5913   5918   5958   6150   6154   6156   6196   6570   6608   6826
        7089   7126   7153   7157   7187   7196   7216   7226   7322   7388   7429   7454   7525   7569   7582
        8037   8116   8134
PRINTS  1319#  4030#  5996   6014   6019   6025   6032   6033   6049   6057   6064   6083   6086   6101   6116
        6390   6428
```

```
PRINTX   1359#  4030#
READBU   1399#  4030#  6818
READEF   1403#  4030#  6788  6790  6792  6795
RFLAGS   1408#  4030#
SETPRI   1413#  4030#  6895  6952
SETVEC   1418#  4030#  6880  6892  6893
SLASH    1424#  4030#
STARS    1438#  4030#
SVC      1452#  4029#  4030
XFER     1612#  4030#  5547#  6896#  6955#  6991#  7028#  7137#  7176#
XFERF    1616#  4030#
XFERT    1620#  4030#

. ABS.  035624     000

 ERRORS DETECTED:  0

 CVCLHB/I,CVCLHB.SEQ/DOC/CRF=SVC34R.MLB,CVCLHB.P11
 RUN-TIME: 22 28 3 SECONDS
 RUN-TIME RATIO: 71/54=1.3
 CORE USED:  19K  (37 PAGES)

 DOCUMENT PAGES:  188
```