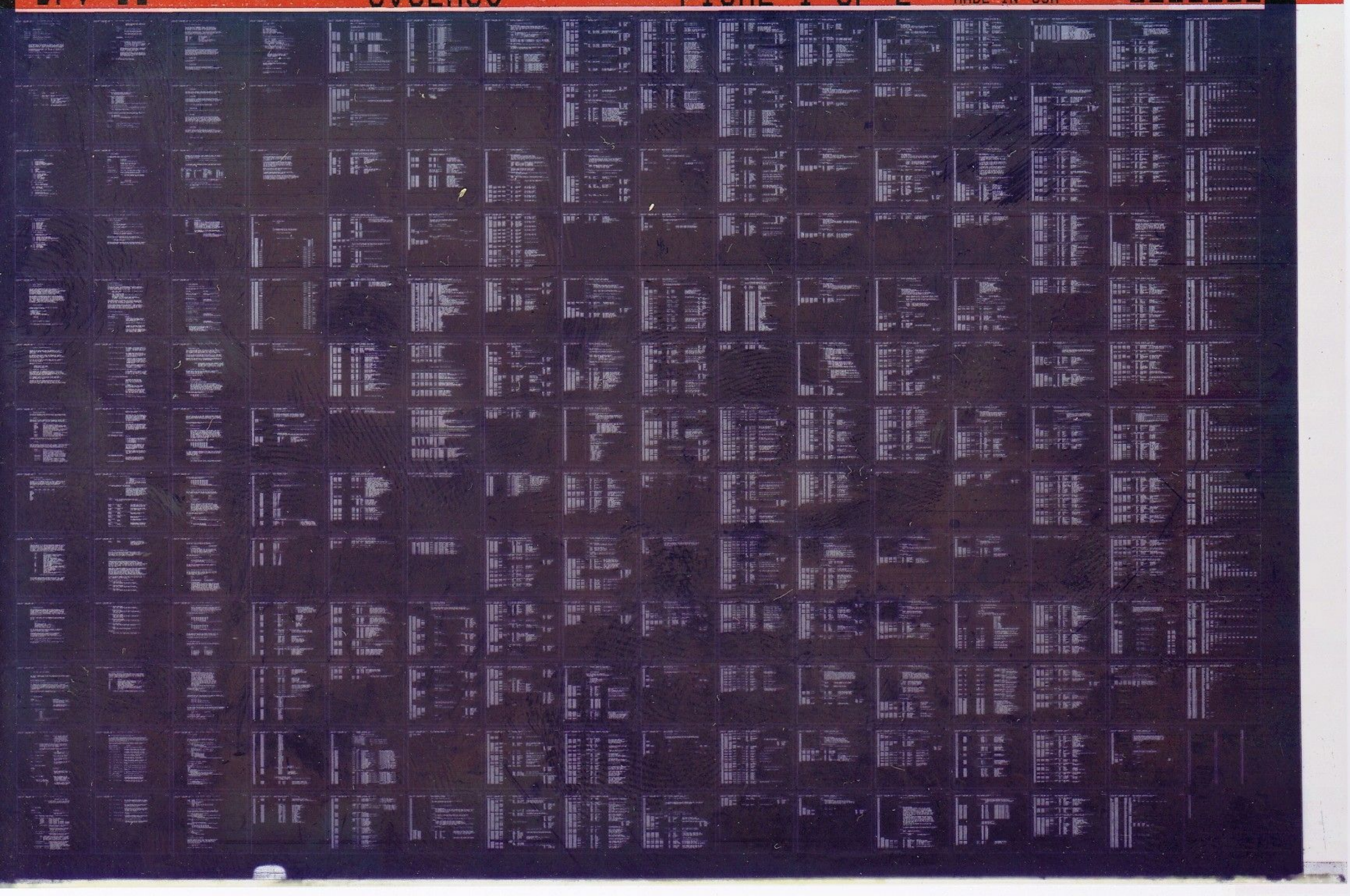
DPV-11

DPV-11 DCLT
CVCLHCO

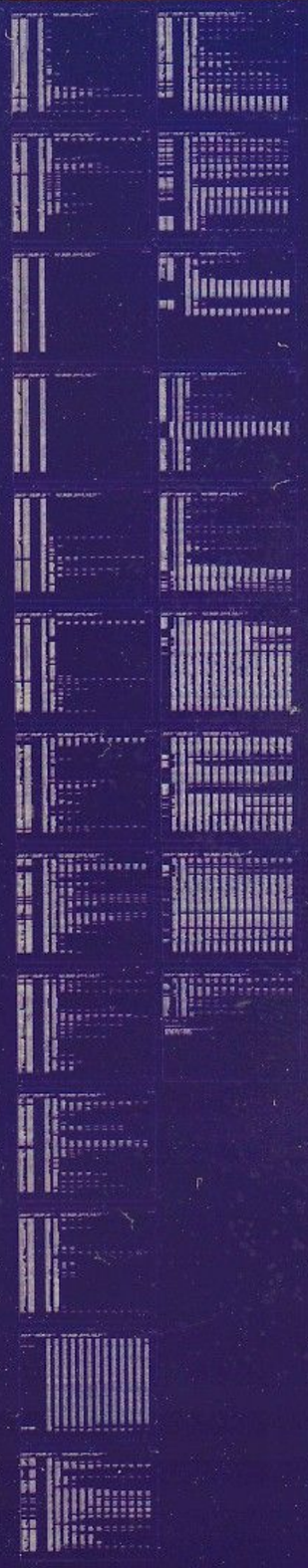
AH-F584C-MC
FICHE 1 OF 2



DPV-11

DPV-11 DCLT
CVCLHCO

AH-F584C-MC
FICHE 2 OF 2



```
1                               .TITLE  CVCLHC DPV-11 DATA COMM. LINK TEST
                                .REM &
```

IDENTIFICATION

PRODUCT CODE: AC-F582C-MC

PRODUCT NAME: CVCLHC0 DPV-11 DATA COMM. LINK TEST

PRODUCT DATE: MARCH 82

MAINTAINER: MERRIMACK DIAGNOSTIC ENGINEERING

AUTHOR: BRUCE RIBOLINI-BRUCE LUHRS

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
C	JUNE-82	ERNIE COOPER	ADD ^C ABORT MESSAGE TO EVENT LOG ADD DDCMP PROTOCOL LAYER

TABLE OF CONTENTS

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 COMMUNICATION LINKS. THIS PROGRAM ALLOWS THE DPV-11 TO COMMUNICATE WITH OTHER SYNCHRONOUS(INCLUDING DDCMP) DEVICES ON POINT TO POINT OR MULTIPOINT NETWORKS. 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 LOOK 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. WITH DDCMP PROTOCOL ENABLED THE HIGHEST SPEED TO BE EXPECTED(LIMITED BY CPU) IS 9.6KB.

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 ?
IS THIS A MULTIPOINT NETWORK: (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. PDP-11 RUNNING INTERPROCESSOR TEST PROGRAM(ITEP))
IF OTHER NODE IS ITEP THEN THE ABOVE "MULTIPOINT NETWORK" QUESTION WILL NOT APPEAR.

IF TO THE "MULTIPOINT NETWORK" QUESTION YOU RESPOND WITH "YES" THEN

```
ADDRESS THIS STATION: (D) A ?
```

WILL BE DISPLAYED. INPUT THE DECIMAL TRIBUTARY NUMBER (1-255) OF THIS DPV-11.

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 RELATED 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. IF "-EEEEEE" IS NOT SPECIFIED THEN THE CONTENTS OF "SSSSSS" IS PRINTED IN WORD FORMAT.
	WHERE "/B" IS OPTIONAL: DEFAULT IS PRINT WORDS "/B" CAUSES PRINT BYTES	

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 =REMOTEMODEM	NOT USED BY DPV..

/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.
/PROTOCOL	ENABLES SUBSET OF DDCMP PROTOCOL - THE DPV CAN NOW COMMUNICATE WITH "INTELLIGENT" SYNCHRONOUS DEVICES THAT SUPPORT DDCMP PROTOCOL IN THEIR MICROCODE. (DMR,DMC,DMV OR DMP)
/NOPROTOCOL	DISABLES DDCMP PROTOCOL - THE DPV NOW RUNS IN NON-PROTOCOL MODE. COMMUNICATION POSSIBLE ONLY BETWEEN DPV'S OR DUP TO DPV LINKS RUNNING DCLT. WHEN COMMUNICATING WITH AN ITEP PROGRAM ALWAYS SELECT NON-PROTOCOL MODE.

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/NOPROTOCOL
```

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/NOPROTOCOL
```

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

2.5.3 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
COUNTERS/FULL	PRINTS THE ENTIRE DDCMP STATISTICAL AND ERROR LOG. SEE SECTION 4.3
COUNTERS/ERRORS	PRINTS ONLY THE DDCMP ERROR LOCATIONS OF THE LOG.
COUNTERS/OFFSET=NN	PRINTS A SINGLE LOCATION OF THE LOG AS SPECIFIED BY THE OCTAL WORD OFFSET VALUE(NN).
	NOTE:: THE DDCMP COUNTERS WILL BE DISPLAYED ONLY WITH PROTOCOL ENABLED(/PROTOCOL).

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

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
RXCSR TXCSR
XXXXXX XXXXXXX
THIS MEANS THAT AFTER A MASTER RESET WAS ISSUED TO DPV THE RXCSR REGISTER WAS NON ZERO.

NO CLEAR TO SEND FROM MODEM
RXCSR TXCSR
XXXXXXX XXXXXXX
WHEN REQUEST TO SEND SIGNAL IS SET MODEM DOES NOT RESPOND WITH CLEAR TO SEND

TIME OUT WAITING FOR RX OR TX TO COMPLETE
RXCSR TXCSR
XXXXXXX XXXXXXX
THIS USUALLY MEANS AN OPEN COMMUNICATION LINK.

MODEM DID NOT RETURN MODEM READY
RXCSR TXCSR
XXXXXXX XXXXXXX
MODEM DID NOT RESPOND WITH MODEM READY(MR).

CRC IN ERROR
RDSR RXCSR
XXXXXX XXXXXX
CRC ERROR DETECTED BY HARDWARE IN INCOMING MESSAGE.

RECEIVER OVERRUN
RDSR RXCSR
XXXXXXX XXXXXXX
RECEIVER WASN'T SERVICED FAST ENOUGH(SOFTWARE) TO PREVENT A CHARACTER FROM BEING LOST.

TIMED OUT IN START,STACK ACK SEQ
THIS USUALLY MEANS THAT THE DPV

RDATA	SDATA	
XXXXXXX	XXXXXXX	IS UNABLE TO ESTABLISH A CONNECTION WITH THE OTHER DEVICE BEING TESTED. SEE DDCMP SPEC. FOR EXPLANATION OF STARTUP SEQUENCE.

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:
                ^C ABORT:
                        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

NOTE:: BECAUSE THE DPV IS AN INTERRUPT DRIVEN DEVICE, IT IS BEST TO DISABLE STATUS TO PREVENT OVERRUN ERRORS.(AT HIGH SPEEDS)

4.3 PRINTING DDCMP STATISTICAL AND ERROR LOG

IF YOU ARE RUNNING THIS PROGRAM WITH DDCMP PROTOCOL ENABLED,YOU CAN EXAMINE(VIA "RPT>" COMMAND)DDCMP STATISTICAL AND ERROR COUNTERS TO GET A BETTER UNDERSTANDING OF WHAT IS HAPPENING ON THE LINK. FOR A FULL DESCRIPTION OF THESE COUNTERS SEE (DIGITAL DATA COMMUNICATION MESSAGE PROTOCOL) SPECIFICATION VERSION 4.1.

BELOW IS A BRIEF DESCRIPTION OF EACH COUNTER. THE MOST IMPORTANT OF THESE ARE DATA MESSAGES SENT/RECEIVED AND DATA ERRORS IN/OUT.

OCTAL #	MESSAGE	MEANING
000000	STATUS FLAGS	USED ONLY IN SOFTWARE DEVELOPMENT.
000000	DATA MSGS TX	# MESSAGES TX'ED DURING THE TEST. RESET TO ZERO AT START OR RESTART. LATCHES AT -1.
000000	DATA MSGS RX	# MESSAGES RX'ED DURING THE TEST.RESET TO ZERO AT START OR RESTART. LATCHES AT -1.
000	HIGHEST MSG TX	MODULO 255 COUNTER. HIGHEST MESSAGE # SENT AND ACK'ED BY REMOTE STATION.
000	HIGHEST MSG ACK	MODULO 255 COUNTER. HIGHEST MESSAGE # RX'D BY REMOTE NODE.(WITH NO ERRORS)
000	NEXT MSG # TO TX	MODULO 255 COUNTER. ALWAYS 1 GREATER THEN CURRENT MESSAGE NUMBER BEING SENT.
000	LAST MSG # TX'ED	MODULO 255 COUNTER. ALWAYS SAME AS HIGHEST # SENT.
000	HIGHEST MSG# RX	NUMBER OF LAST MESSAGE RX'ED AND ACK'ED.
000	TRIB ADDR	IF MULTIPOINT THEN ADDRESS THIS STATION.
000	REMOTE TIME OUTS	MODULO 255 COUNTER. REPLYS RECEIVED AND ACK'ED.
000	GLOBAL CRC ERRORS	IF MULTIPOINT NETWORK-CRC ERRORS DETECTED.
000	NAK REASON	REASON FOR SENDING LAST NAK.
000	SEL THRESH ERRS	HALF/DUPLEX ONLY. SELECT TIME OUTS.
000	RX THRESH ERRS	INCREMENTED WHEN ERROR DETECTED IN INCOMING MESSAGE.(MODULO 8 COUNTER) RESET WHEN GOOD MESSAGE RECEIVED.
000	TX THRESH ERRS	INCREMENTED WHEN NAK RECEIVED. RESET WHEN ACK RECEIVED.(MODULO 8 COUNTER)

000	DATA ERRORS OUT	NAKS RECEIVED BECAUSE OF HEADER CRC ERROR OR DATA CRC ERRORS OR MESSAGE NOT RECEIVED AT ALL(REP). INDICATES NOISE ON TRANSMIT LINE.
000	DATA ERRORS IN	NAKS SENT BECAUSE HEADER CRC ERROR OR DATA CRC ERROR DETECTED IN INCOMING MESSAGE. MESSAGE TAKING NOISE HITS.
000	LOCAL BUFFER ERRS	EITHER NO BUFFER WAS AVAILABLE FOR INCOMING MESSAGE OR BUFFER THAT WAS AVAILABLE WAS TOO SMALL FOR INCOMING MESSAGE. USUALLY A SOFTWARE SPEED PROBLEM.
000	REMOTE BUFFER ERRS	SAME AS LOCAL BUT BUFFER PROBLEMS AT REMOTE STATION.
000	REMOTE STA ERRS	RX OVERRUN ERRORS(RX WASN'T SERVICED FAST ENOUGH) OR IF FORMAT ERROR A CRC EXISTED AND WASN'T DETECTED BY HARDWARE.
000	LOCAL STA ERRS	SAME AS REMOTE STATION ERRORS.
000	TX / RX THRESH ERR	OVERFLOW FROM RX OR TX THRESHOLD COUNTERS. INDICATES A PERSISTENT LINK PROBLEM THAT ISN'T CORRECTED AFTER 7 RETRIES.

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] PT-PT =0  MULTIPOINT = 1
.WORD   1               ;[12] TRIB ADDRESS THIS STATION
.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. ALL MESSAGES ARE PRECEDED BY SYNC
CHARACTERS(ITEP = 026)(PROTOCOL = 226). ITEP MESSAGES CONTAIN
NO CRC CHARACTERS. NON-ITEP MESSAGES ARE ENCLOSED IN A DDCMP
ENVELOPE. WITH PROTOCOL ENABLED CONTROL MESSAGES(ACK,NAK,...)
ALSO APPEAR ON THE LINK.

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

IF PROTOCOL IS SELECTED THE HALF DUPLEX START COLUMN CAN BE IGNORED.

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/NPR
NOTE: DUP ITEP SENDS ONLY 56 CHARS	
FOR ONE WAY IN.....	
SET SWITCHES TO 1222	RUN MODE=TRA/STATUS/NPR
FOR EXTERNAL LOOPBACK....	
SET SWITCHES.....1224	CLEAR EXPECTED SET EXP=ITEP/S=56 RUN MODE=ACTIVE/STATUS/CHECK/NPR
FOR INTERNAL LOOPBACK.....	
SET SWTICHES.....1260	CLEAR EXPECTED SET EXP=ITEP/S=56 RUN MODE=ACTIVE/STATUS/CHECK/NPR

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.
REMEMBER WHEN YOU ARE COMMUNICATING WITH AN "ITEP" NODE
ALWAYS DISABLE('/NOPROTOCOL').

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.

EVEN IF YOU ARE CHECKING OUT DPV TO DPV LINKS, IT IS A GOOD IDEA
TO ENABLE PROTOCOL. BY EXAMINING THE DDCMP STATISTICAL AND ERROR
LOG, YOU CAN GET A BETTER PICTURE OF WHAT IS HAPPENING ON THE LINK.

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/NOPROTOCOL
          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/NOPROTOCOL
```

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/NOPROTOCOL
        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/NOPROTOCOL
        DCLT> (A)  ?

FOR NODE B:
        INI RXQ EOP RXQ EOP RXQ EOP
             MODE=RECEIVE/PASS=00000
                        /STATUS/NOCHECK/NOECHO/NOMODEM/NOPROTOCOL
        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/NOPROTOCOL
        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/NOPROTOCOL
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/NOPROTOCOL
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/NOPROTOCOL
        DCLT> (A)  ?


A GOOD VARIATION THAT COULD BE TRIED IS TO LOAD THE TRANSMIT LIST
AND EXPECT LIST WITH A LARGE MESSAGE(512 CHARACTERS),ENABLE PROTOCOL
AND RUN MANY PASSES. SET BOTH ENDS THE SAME.

        DCLT> (A) ? CL T
        DCLT> (A) ? CL E
        DCLT> (A) ? SE T=CCITT/SIZE=512
        DCLT> (A) ? SE E=T
        DCLT> (A) ? R M=A/PA=255/PR/CH/NST


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/NOPROTOCOL
                            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/NOPROTOCOL
                            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 "OPR SPCD=A-Z,SP,TAB,0-9 IN QUOTES"
                     RUN MODE=MTYP/LOOP=LTYP/CHECK,PROTOCOL,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/NOPROTOCOL
  DCLT> (A)  ?
IF YOU WERE TO EXECUTE THE RUN COMMAND
        C E
        C T
        R M=A/LO=I/ST/CH/PAS=3/PR
WHAT YOU WOULD SEE (IF USING DEFAULT TRANSMIT AND EXPECT
MESSAGES) IS
        INI RXQ TXQ TXC CMP EOP RXQ TXQ
        TXC CMP EOP RXQ TXQ TXC CMP EOP
        MODE=ACTIVE/LOOP=INTERNAL/PASS=0000
                /STATUS/CHECK/NOECHO/NOMODEM/PROTOCOL
  DCLT> (A)  ?

 IF YOU USE THE EXIT COMMAND
        EXIT
 WHAT YOU WOULD SEE IS
        CVCLH 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 DCLT> 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
                    COUNTER/SW - PRINT DDCMP COUNTERS
                    WHERE /SW=FULL, /ERRORS, /OFFSET=NN(O)

                    RPT> (A)  ? LOG

                    THE LOG COMMAND PRODUCES THE FOLLOWING

                    >>> DCLT EVENT LOG ENTRY >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
                        0:01:17     ^C ABORT
                       PASS = 12   ERRORS = 0    STRT-TO = 0
                    >>> DCLT EVENT LOG ENTRY >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
                        0:01:16   TRANSMIT MSG QUEUED
                       ADDR OF MSG=003150   BYTE COUNT= 58
                            MODEM STATUS: CTS DSR DCD RTS RI SQD TM
                                          0   1   0   0   1  1   0
                    >>> DCLT EVENT LOG ENTRY >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
                        0:01:16   RECEIVE SPACE QUEUED
                       ADDR OF MSG=004150   BYTE COUNT= 512
                            MODEM STATUS: CTS DSR DCD RTS RI SQD TM
                                          0   1   0   0   1  1   0
                                         .....
                                         .....
                                         .....

                            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. ALSO DISABLE DATACHECKING('/NOCHECK').

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.

IF YOU ARE RUNNING HALF-DUPLEX IT IS BEST TO USE THE "/NOMODEM" SWITCH BECAUSE EVERY TIME THE LINE IS TURNED-AROUND A MODEM CHANGE WILL BE REPORTED.

IF YOU ARE RUNNING WITH PROTOCOL SELECTED THE MODEM STATUS AS REPORTED IN THE EVENT LOG MAY NOT INDICATE THE TRUE CONDITION OF THE MODEM SIGNALS. THIS IS BECAUSE THE EVENT IS LOGGED BEFORE THE MESSAGE IS PASSED TO THE DDCMP PROTOCOL LAYER WHERE THE RX,TX AND MODEM SIGNALS ARE MANIPULATED.

```
1940
1941
1942
1943
1944                                    .SBTTL  PROGRAM HEADER
1945
1946  002000                                    BGNMOD
1947
1948
1949
1950
1951
1952                                    ;++
1953                                    ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1954                                    ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1955                                    ;--
1956
1957  002000                                    POINTER BGNRPT,BGNAU,BGNDU
1958
1959
1960
1961
1962
1963  002000                            HEADER  CVCLH,C,0,1800.,0,#PRI07
1964  002000
1965  002000  103                               LS$NAME::
1966  002001  126                                       .ASCII  /C/
1967  002002  103                                       .ASCII  /V/
1968  002003  114                                       .ASCII  /C/
1969  002004  110                                       .ASCII  /L/
1970  002005  000                                       .ASCII  /H/
1971  002006  000                                       .BYTE   0
1972  002007  000                                       .BYTE   0
1973  002010                                            .BYTE   0
1974  002010  103                               LS$REV::
1975  002011                                            .ASCII  /C/
1976  002011  060                               LS$DEPO::
1977  002012                                            .ASCII  /0/
1978  002012  000000                            LS$UNIT::
1979  002014                                            .WORD   0
1980  002014  003410                            LS$TIML::
1981  002016                                            .WORD   1800.
1982  002016  046250                            LS$HPCP::
1983  002020                                            .WORD   LS$HARD
1984  002020  000000                            LS$SPCP::
1985  002022                                            .WORD   0
1986  002022  002130                            LS$HPTP::
1987  002024                                            .WORD   LS$HW
1988  002024  000000                            LS$SPTP::
1989  002026                                            .WORD   0
1990  002026  046636                            LS$LADP::
1991  002030                                            .WORD   LS$LAST
1992  002030  000000                            LS$STA::
1993  002032                                            .WORD   0
1994  002032  000000                            LS$CO::
1995  002034                                            .WORD   0
                                                LS$DTYP::
```

```
1996  002034  000000
1997  002036
1998  002036  000000
1999  002040
2000  002040  002124
2001  002042
2002  002042  000340
2003  002044
2004  002044  000000
2005  002046
2006  002046  000000
2007  002050
2008  002050     003
2009  002051     003
2010  002052
2011  002052  000000
2012  002054  000000
2013  002056
2014  002056  000000
2015  002060
2016  002060  011526
2017  002062
2018  002062  025334
2019  002064
2020  002064  000000
2021  002066
2022  002066  000000
2023  002070
2024  002070  026332
2025  002072
2026  002072  026324
2027  002074
2028  002074  000000
2029  002076
2030  002076  011536
2031  002100
2032  002100  104035
2033  002102
2034  002102  000000
2035  002104
2036  002104  025350
2037  002106
2038  002106  026236
2039  002110
2040  002110  026234
2041  002112
2042  002112  025342
2043  002114
2044  002114  000000
2045  002116
2046  002116  000000
2047  002120
2048  002120  000000
2049
```

```
                .WORD   0
L$APT::
                .WORD   0
L$DTP::
                .WORD   L$DISPATCH
L$PRIO::
                .WORD   #PRI07
L$ENVI::
                .WORD   0
L$EXP1::
                .WORD   0
L$MREV::
                .BYTE   C$REVISION
                .BYTE   C$EDIT
L$EF::
                .WORD   0
                .WORD   0
L$SPC::
                .WORD   0
L$DEVP::
                .WORD   L$DVTYP
L$REPP::
                .WORD   L$RPT
L$EXP4::
                .WORD   0
L$EXP5::
                .WORD   0
L$AUT::
                .WORD   L$AU
L$DUT::
                .WORD   L$DU
L$LUN::
                .WORD   0
L$DESP::
                .WORD   L$DESC
L$LOAD::
                EMT     E$LOAD
L$ETP::
                .WORD   0
L$ICP::
                .WORD   L$INIT
L$CCP::
                .WORD   L$CLEAN
L$ACP::
                .WORD   L$AUTO
L$PRT::
                .WORD   L$PROT
L$TEST::
                .WORD   0
L$DLY::
                .WORD   0
L$HIME::
                .WORD   0
```

```
 CVCLHC DPV-11 DATA COMM. LINK TEST      MACY11 30A(1052)  23-MAR-82  16:43  PAGE 46
CVCLHC.P11    22-MAR-82 11:09           DISPATCH TABLE

                                        .SBTTL  DISPATCH TABLE

                                        ;++
                                        ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
                                        ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
                                        ;--

                                                DISPATCH 1

002122                                                                  .WORD   1
002122  000001                                                 L$DISPATCH::
002124                                                                  .WORD   T1
002124  026340
```

```
                                .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
002126  000010                                                                 .WORD   L10000-L$HW/2
002130                                                                  L$HW::
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] SPARE
002140  000000                          .WORD   0              ;[10] PT-PT = 0  MULTIPOINT = 1
002142  000001                          .WORD   1              ;[12] TRIB ADDRESS THIS STATION
002144  000000                          .WORD   0              ;[14] OTHER NODE "ITEP"
002146  000000                          .WORD   0              ;[16] SPARE


002150                                  ENDHW
002150                                                                  L10000:
```

```
2101
2102
2103
2104
2105                                    .SBTTL  GLOBAL EQUATES SECTION
2106
2107
2108
2109
2110                                    ;++
2111                                    ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
2112                                    ; ARE USED IN MORE THAN ONE TEST.
2113                                    ;--
2114
2115
2116  002150                                    EQUALS
2117                                    ;
2118                                    ; BIT DIFINITIONS
2119                                    ;
2120            100000                  BIT15== 100000
2121            040000                  BIT14== 40000
2122            020000                  BIT13== 20000
2123            010000                  BIT12== 10000
2124            004000                  BIT11== 4000
2125            002000                  BIT10== 2000
2126            001000                  BIT09== 1000
2127            000400                  BIT08== 400
2128            000200                  BIT07== 200
2129            000100                  BIT06== 100
2130            000040                  BIT05== 40
2131            000020                  BIT04== 20
2132            000010                  BIT03== 10
2133            000004                  BIT02== 4
2134            000002                  BIT01== 2
2135            000001                  BIT00== 1
2136                                    ;
2137            001000                  BIT9==  BIT09
2138            000400                  BIT8==  BIT08
2139            000200                  BIT7==  BIT07
2140            000100                  BIT6==  BIT06
2141            000040                  BIT5==  BIT05
2142            000020                  BIT4==  BIT04
2143            000010                  BIT3==  BIT03
2144            000004                  BIT2==  BIT02
2145            000002                  BIT1==  BIT01
2146            000001                  BIT0==  BIT00
2147                                    ;
2148                                    ; EVENT FLAG DEFINITIONS
2149                                    ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
2150                                    ;
2151            000040                  EF.START==      32.                     ; START COMMAND WAS ISSUED
2152            000037                  EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
2153            000036                  EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
2154            000035                  EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
2155            000034                  EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
2156                                    ;
```

```
2157                                ;
2158                                ; PRIORITY LEVEL DEFINITIONS
2159                                ;
2160            000340              PRI07== 340
2161            000300              PRI06== 300
2162            000240              PRI05== 240
2163            000200              PRI04== 200
2164            000140              PRI03== 140
2165            000100              PRI02== 100
2166            000040              PRI01== 40
2167            000000              PRI00== 0
2168                                ;
2169                                ;OPERATOR FLAG BITS
2170                                ;
2171            000004              EVL==         4
2172            000010              LOT==        10
2173            000020              ADR==        20
2174            000040              IDU==        40
2175            000100              ISR==       100
2176            000200              UAM==       200
2177            000400              BOE==       400
2178            001000              PNT==      1000
2179            002000              PRI==      2000
2180            004000              IXE==      4000
2181            010000              IBE==     10000
2182            020000              IER==     20000
2183            040000              LOE==     40000
2184            100000              HOE==    100000
2185
```

```
2186                                    ;***** INDEPENDENT EQUATES
2187
2188    001000                                  BUFLIM=512.             ;MAX BUFFER SIZE IN BYTES
2189                                                                    ;  APPLIES TO TX,RX AND CMP BUFFS
2190    000017                                  MSGLIM=15.              ;MAX NO. OF MESSAGES PER BUFFER
2191                                                                    ;   (FOR EACH INCREMENT (+1) TO MSGLIM,
2192                                                                    ;    ADD 6 WORDS TO THE POINTER TABLE
2193                                                                    ;    (PTRTAB:) SINCE THIS MEANS 2 MORE
2194                                                                    ;    "POINTER" WORDS PER BUFFER.
2195
2196                                    ;MODE OF OPERATION EQUATES
2197    000000                                  REC=0                   ;RECEIVE MODE
2198    000001                                  TRA=1                   ;TRANSMIT MODE
2199    000002                                  PAS=2                   ;PASSIVE MODE
2200    000003                                  ACT=3                   ;ACTIVE MODE
2201    000004                                  DOW=4                   ;DOWN-LINE-LOAD MODE
2202    000005                                  TAL=5                   ;TALK MODE
2203    000006                                  LIS=6                   ;LISTEN MODE
2204                                    ;MAINT LOOP TYPE EQUATES
2205    000000                                  NONE=   0               ;NO LOOP
2206    000001                                  TTL=    1               ;INTERNAL TTL
2207    000002                                  CABLE=  2               ;CABLE LOOP
2208    000003                                  MODLOC= 3               ;MODEM LOCAL
2209    000004                                  MODREM= 4               ;MODEM REMOTE
2210    000005                                  MOP=    5               ;MOP
2211
2212
2213                                    ;CLOCK ENABLE VALUES TO BE LOADED IN CLK'S CSR
2214    000100                                  LCLKEN= 100             ;L-CLOCK CSR VALUE TO ENABLE THE CLOCK
2215    000111                                  PCLKEN= 111             ;P-CLOCK CSR VALUE TO ENABLE THE CLOCK
2216    001600                                  PCLKCT= 1600            ;P-CLOCK COUNT SET REGISTER FOR COUNTER
2217
2218                                    ;PARAM WORD EQUATES
2219
2220    000001                                  STATB=  BIT0            ;OPERATOR AWAKE ASKED FOR
2221    000002                                  DATCKB= BIT1            ;DATA CHECK BIT
2222    000004                                  ECHOB=  BIT2            ;ECHO BIT
2223    000010                                  MODCHK= BIT3            ;MODEM STATUS CHECK BIT
2224    000020                                  CRCB=   BIT4            ;CRC CALCUALTE ASKED FOR
2225    000040                                  PROTOB= BIT5            ;PROTOCOL PROCESSING ASKED FOR
2226    000100                                  PRORUN= BIT6            ;DDCMP PROTOCOL RUNNING
2227    000200                                  ABORT=  BIT7            ;FATAL PROTOCOL ERROR
2228
2229                                    ;OPTION TYPE EQUATES
2230
2231
2232    000000                                  DPV=    0       ;CODE FOR DPV CHAR MODE
2233
2234                                    ;EVENT LOG MESSAGE TYPES (USED TO LOCATE EVENT DESCRIPTION IN EVENT TABLE
2235                                    ;  AND DISPATCHING TO SEPERATE SECTIONS OF THE EVENT REPORTING SECTION)
2236    000000                                  TXQ=    0               ;TRANSMIT MESSAGE QUEUED
2237    000002                                  TXC=    2               ;TRANSMIT COMPLETE
2238    000004                                  RXQ=    4               ;RECEIVE BUFFER QUEUED
2239    000006                                  RXC=    6               ;RECEIVE COMPLETE
2240    000010                                  DER=    10              ;DEVICE INFORMATION
2241    000012                                  DVI=    12              ;DEVICE ABOUT TO INIT
```

```
                                DCK=    14              ;DATA COMPARISON RESULTS
                                MSC=    16              ;MODEM STATUS CHANGE
                                DLE=    20              ;DATA COMPARISON LENGH ERROR
                                DDE=    22              ;DATA COMPARISON DATA ERROR
                                EOP=    24              ;END OF PASS
                                ABO=    26              ;^C ABORT

                            ;EQUATES FOR FLAG WORD

000001                          ININT=  BIT0            ;INPUT INT. REC.
000002                          OTINT=  BIT1            ;OUTPUT INT REC
000004                          QRX=    BIT2            ;RX QUED /COMPL
000010                          QTX=    BIT3            ;TX QUED/COMPL
000100                          ERX=    BIT6            ;EXPECT TO GET A RX COMPLED
000200                          ETX=    BIT7            ;EXPECT TO GET A TX COMPLETED

000020                          TXM=    BIT4            ;INDICATES TO TX INTERRUPT ROUTINE
                                                        ;THAT IT IS TIME TO TRANSMIT BODY OF MSG.
000040                          RXM=    BIT5            ;INDICATES TO RX INTERUPPT ROUTINE
                                                        ;THAT IT IS TIME TO REC MSG BODY
000400                          BCC=    BIT8            ;TIME FOR CRC CHECK.

001000                          PAD=    BIT9            ;INDICATES THAT PAD MUST BE SENT

002000                          INOVR=  BIT10           ;INIT OVER

004000                          FIRST=  BIT11           ;FIRST  TIME FOR CTS

                            ; SPECIAL CLI CODES FOR "CHAR" ARGUMENT IN CLI CALLS
                            ;  (COMMAND LINE INTERPRETER DEFINITIONS)
000000                          CLIERR= 0
000001                          CLIEXI= 1
000002                          CLIBR=  2
000003                          CLIBIF= 3
000004                          CLISPA= 4
000005                          CLINUM= 5
000006                          CLIALP= 6
000007                          CLIALN= 7
000010                          CLIOCT= 8.
000011                          CLIDEC= 9.
000012                          CLISTR= 10.

                            ; DEFS FOR COMMAND LINE INTERPRETATION ACTION VALUES
000000                          NULL=0
000001                          CLEAR=1
000002                          SHOW=2
000003                          CHECK=3
000004                          RUN=4
000005                          HLP=5
000006                          CSHEXP=6
000007                          CSHTRN=7
000010                          SETEXP=10
000011                          SETTRN=11
000012                          SIZE=12
000013                          QCOPY=13
```

```
2298    000014                  NUM=14
2299    000015                  OPRMSG=15
2300    000016                  STATUS=16
2301    000017                  ENDQO=17
2302    000020                  CMSG0=20
2303    000021                  CMSG1=21
2304    000022                  CMSG2=22
2305    000023                  CMSG3=23
2306    000024                  CMSG4=24
2307    000025                  CMSG5=25
2308    000026                  CMSG6=26
2309    000027                  ATVMOD=27
2310    000030                  PASMOD=30
2311    000031                  RECMOD=31
2312    000032                  LISMOD=32
2313    000033                  DLLMOD=33
2314    000034                  TRAMOD=34
2315    000035                  TALMOD=35
2316    000036                  NO=36
2317    000037                  ECHO=37
2318    000040                  CRC=40
2319    000041                  PROTO=41
2320    000042                  PASC=42
2321    000043                  MOP=43
2322    000044                  TTLLOP=44
2323    000045                  CBLLOP=45
2324    000046                  LMDLOP=46
2325    000047                  RMDLOP=47
2326    000050                  NOTNUF=50
2327    000051                  BADCHR=51
2328    000052                  DMPS=52
2329    000053                  DMPE=53
2330    000054                  DMPQ=54
2331    000055                  PRNT=55
2332    000056                  MOSC=56
2333    000057                  EXIT=57         ;REV B BY EC
2334    000060                  SETET=60        ;REV B BY EC
2335                    ;FOLLOWING EQUATES USED IN REPORT CLI ; REV B EC
2336    000001                  RPHLP=1
2337    000002                  RPEXT=2
2338    000003                  RPLOG=3
2339    000004                  RPERR=4         ;'COUNTER/ERROR'
2340    000005                  RPFUL=5         ;'COUNTER/FULL'
2341    000006                  RNOTNF=6        ;MORE COMMAND NEEDED
2342    000007                  RPSWO=7         ;VALIDATE OFFSET
2343
2344
2345                    ;***** DEVICE DEPENDENT EQUATES
2346                    ; MODEM SIGNAL BIT DEFINITONS
2347                    ;       IF SIGNAL AVAILABLE IN DEVICE, EQUATE NAME TO BIT POSITION,
2348                    ;       ELSE EQUATE IT TO = 0
2349
2350    020000                  CTS=    BIT13           ;CLEAR TO SEND (CIRCUIT CB)
2351    001000                  DSR=    BIT9            ;DATA SET READY (CIRCUIT CC)
2352    010000                  DCD=    BIT12           ;DATA CARRIER DETECT (CIRCUIT CF)
2353    000004                  RTS=    BIT2            ;REQUEST TO SEND (CIRCUIT CA)
```

```
2354            040000                  RI=     BIT14           ;RING INDICATOR (CIRCUIT CE)
2355            000040                  SQD=    BIT5            ;SIGNAL QUALITY DETECT (CIRCUIT CG)
2356            000040                  TM=     BIT5            ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)
2357
2358
2359
2360                                    ; DEVICE SIGNALS
2361            000002                          DTR=    BIT1            ;DATA TERMINAL READY
2362            000020                          RXENA=  BIT4            ;RECEIVER ENABLE
2363            000040                          DSITEN= BIT5            ;DATA SET CHANGE ENABLE
2364            000100                          RINTEN= BIT6            ;REC INT. ENABLE
2365            000200                          RDATRY= BIT7            ;REC DATA READY
2366            002000                          RSTARY= BIT10           ;REC STATUS READY
2367            004000                          RXACT=  BIT11           ;REC ACTIVE
2368            000001                          RESET=  BIT0            ;MASTER RESET
2369            000002                          TXACT=  BIT1            ;TX ACTIVE
2370            000004                          TBMT=   BIT2            ;TX BUFFER EMPTY
2371            000010                          TTLL=   BIT3            ;TTL LOOP BIT
2372            000020                          TXENA=  BIT4            ;TX ENABLE
2373            000100                          TINTEN= BIT6            ;TX INT ENABLE
2374            000400                          TSOM=   BIT8            ;TX START OF MSG.
2375            001000                          TEOM=   BIT9            ;TX END OF MSG.
2376            100000                          TERR=   BIT15           ;TX ERROR
2377            100000                          RERR=   BIT15           ;REC OVER RUN
2378            000226                          SYN=    226             ;SYNC WORD
2379            100000                          CRCOK=  BIT15           ;CRC CHECKED GOOD
2380            004000                          RXOVER= BIT11           ;RECEIVER OVERRUN ERROR
2381
```

```
                                .SBTTL  GLOBAL DATA SECTION
                                .SBTTL          DEFAULT MESSAGE DEFINITIONS AND TABLES

                                ;++
                                ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
                                ; IN MORE THAN ONE TEST.
                                ;--

                                ;MESSAGE BYTE COUNT TABLE

002150                          DMSGCT:
002150  000001                  MSG0C:  .WORD   EMSG0-MSG0      ;BYTE COUNT OF MESSAGE #0
002152  000001                  MSG1C:  .WORD   EMSG1-MSG1      ;BYTE COUNT OF MESSAGE #1
002154  000001                  MSG2C:  .WORD   EMSG2-MSG2      ;BYTE COUNT OF MESSAGE #2
002156  000001                  MSG3C:  .WORD   EMSG3-MSG3      ;BYTE COUNT OF MESSAGE #3
002160  000100                  MSG4C:  .WORD   EMSG4-MSG4      ;BYTE COUNT OF MESSAGE #4
002162  000072                  MSG5C:  .WORD   EMSG5-MSG5      ;BYTE COUNT OF MESSAGE #5
002164  000101                  MSG6C:  .WORD   EMSG6-MSG6      ;BYTE COUNT OF MESSAGE #6
002166  000000                  OPCNT:  .WORD   0               ;BYTE COUNT FOR OPERATOR SPEC'D MSG.
002170  000001                  MSG8C:  .WORD   EMSG8-MSG8      ;BYTE COUNT OF RECEIVE BUFFER FILL PATTERN

                                ;MESSAGE ADDRESS TABLE

002172                          DMSGAD:
002172  002214                          MSG0                    ;ADDRESS OF MESSAGE #0
002174  002215                          MSG1                    ;ADDRESS OF MESSAGE #1
002176  002216                          MSG2                    ;ADDRESS OF MESSAGE #2
002200  002217                          MSG3                    ;ADDRESS OF MESSAGE #3
002202  002220                          MSG4                    ;ADDRESS OF MESSAGE #4
002204  002320                          MSG5                    ;ADDRESS OF MESSAGE #5
002206  002412                          MSG6                    ;ADDRESS OF MESSAGE #6
002210  002520                          OPBUF                   ;ADDRESS OF OPERATOR SPEC'D MSG.
002212  002642                          MSG8                    ;ADDRESS OF RECEIVE BUFFER FILL PATTERN

002214  000                     MSG0:   .BYTE   000             ;MESSAGE OF ALL 0'S
002215                          EMSG0:
002215  377                     MSG1:   .BYTE   377             ;MESSAGE OF ALL 1'S
002216                          EMSG1:
002216  252                     MSG2:   .BYTE   252             ;MESSAGE OF ALTERNATING 1'S
002217                          EMSG2:
002217  125                     MSG3:   .BYTE   125             ;MESSAGE OF ALTERNATING 0'S
002220                          EMSG3:
002220                          MSG4:                           ;"CCITT" 512-BIT (VS. 511 BITS) TEST PATTERN
002220  177603  157427  031011          .WORD   177603,157427,031011,047321,163715,105221,143325,142304
002226  047321  163715  105221
002234  143325  142304
002240  040041  014116  052606          .WORD   040041,014116,052606,172334,105025,123754,111337,111523
002246  172334  105025  123754
002254  111337  111523
002260  030030  145064  137642          .WORD   030030,145064,137642,143531,063617,135075,066730,026575
002266  143531  063617  135075
002274  066730  026575
002300  052012  053627  070071          .WORD   052012,053627,070071,151172,165044,031605,166632,016741
002306  151172  165044  031605
002314  166632  016741
002320                          EMSG4:
```

```
2438  002320                          MSG5:                           ;"INTERPROCESSOR TEST PROGRAM'S (ITEP)" MESSAGE
2439                                                                  ; #1, (DP1:)
2440  002320  077577  040444  052040          .ASCII  <177><177>/$A THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG./
2441  002326  042510  050440  044525
2442  002334  045503  041040  047522
2443  002342  047127  043040  054117
2444  002350  045040  046525  042520
2445  002356  020104  053117  051105
2446  002364  052040  042510  046040
2447  002372  055101  020131  047504
2448  002400  027107
2449  002402  005015  077401  077577          .ASCIZ  <15><12><001><177><177><177><177>
2450  002410  000177
2451  002412                          EMSG5:
2452  002412                          MSG6:                           ;ALPHA-NUMERICS (OR FUTURE COMM TURNAROUND MSG)
2453  002412  022043  021041  023040          .ASCII  /#$!" &'()*+,-.0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ/
2454  002420  024047  025051  026053
2455  002426  027055  030460  031462
2456  002434  032464  033466  034470
2457  002442  035472  036474  037476
2458  002450  040500  041502  042504
2459  002456  043506  044510  045512
2460  002464  046514  047516  050520
2461  002472  051522  052524  053526
2462  002500  054530     132
2463  002503     057  056133  057135          .ASCIZ  ?/[\]^_%?
2464  002510  022537     000
2465  002513                          EMSG6:
2466        002514                            .EVEN
2467
2468                                  ; ***************************************************************************
2469                                  ;THESE THREE STORAGE AREAS MUST NOT BE SEPERATED !!!!
2470
2471  002514  047045  040445          OPBFPT: .ASCII  /%N%A/
2472  002520  000122                  OPBUF:  .BLKB   82.              ;BUFFER FOR OPERATOR SPEC'D MESSAGES
2473  002642                          OPEND:
2474
2475                                  ; THE ABOVE THREE LINES MUST BE KEPT TOGETHER
2476                                  ; ***************************************************************************
2477
2478  002642     033                  MSG8:   .BYTE   33              ;RECEIVE BUFFER FILL PATTERN
2479  002643                          EMSG8:
2480        002644                            .EVEN
```

```
                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                ; THE FOLLWING IS THE AREA USED TO TRANSMIT AND REC THE ;
                                ; HEADER MSGS. AND THE START,STACK ACK SEQUENCES.       ;
                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                ;; THE TRANSMIT HEADER MESSAGE WILL BE STORED HERE
 002644     000                 HDMSG:  .BYTE 0         ;FILLER
 002645     201                 HDMID:  .BYTE 201       ;MESSAGE TYPE STORED HERE
 002646                         HDMTYP:                 ;IF CONTROL MESSAGE, TYPE IS STORED HERE
 002646  000000                 HDMCC:  .WORD   0       ;CHAR COUNT GOES HERE
 002650     001                 HDMREP: .BYTE   1       ;RESPONSE NUMBER
 002651     001                 HDMNUM: .BYTE   1       ;MSG. NUMBER
 002652     001                 HDMADR: .BYTE   1       ;ADDR TO.
 002653                         HSMSE:
         002654                         .EVEN
 002654  000006                 HDMC:   .WORD   6
                                ;; RECEIVED HEADER WILL BE STORED HERE
 002656     000                 RHDMSG: .BYTE 0
 002657     201                 RHDMID: .BYTE 201       ;MESSAGE TYPE GOES IN HERE
 002660                         RHDTYP:                 ;CONTROL MESSAGE TYPE GOES HERE
 002660  000000                 RHDMCC: .WORD           ;BYTE COUNT GOES HERE
 002662     001                 RHDREP: .BYTE   1       ;RESP NUM
 002663     001                 RHDNUM: .BYTE   1       ;MSG NUM
 002664     001                 RHDADR: .BYTE   1       ;ADDR TO.
         002666                         .EVEN
```

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

                                        CMDBUF: .BLKB   82.             ;BUFFER FOR OPERATOR COMMANDS
                                        KEYWD1: .WORD   0               ;THIS LOC WILL =1 IF CLEAR TYPED, 2 FOR SHOW,
                                                                        ; A 4 IF RUN WAS TYPED, 5 IF HELP WAS TYPED
                                        QUALFG: .WORD   0               ;THIS LOC HOLDS QUALIFIER VALUE (SIZE OR COPY)
                                        QUALVL: .WORD   0
                                        HLPTAB: .WORD   HLP1
                                                .WORD   HLP2
                                                .WORD   HLP3
                                                .WORD   HLP3A
                                                .WORD   HLP4
                                                .WORD   HLP4A
                                                .WORD   HLP5
                                                .WORD   HLP6
                                        HLPEND:
                                        ;INDEX TABLE FOR REPORT 'RPT>' HELP MESSAGES REV B EC
                                        RHLPTB: .WORD   RHLP1
                                                .WORD   RHLP2
                                                .WORD   RHLP3
                                                .WORD   RHLP4
                                                .WORD   RHLP5
                                                .WORD   RHLP6
                                        RHLPEN:

                                        SHTYTB: .WORD   SHTYP0,SHTYP1,SHTYP2,SHTYP3,SHTYP4,SHTYP5,SHTYP6,SHTYP7

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

003072  000  377  252  SHTAB:  .BYTE   0,377,252,125,203,177,043
003075  125  203  177
003100  043
003101                  SHTEND:
003102                  .EVEN

                        MODES:  .WORD   M00     ;ADDRESSES OF MODE TYPES IN ASCII
                                .WORD   M01
                                .WORD   M02
                                .WORD   M03
                                .WORD   M04
                                .WORD   M05
                                .WORD   M06

                        LOOPS:  .WORD   LP0     ;ADDRESSES OF LOOP TYPES IN ASCII
                                .WORD   LP1
                                .WORD   LP2
                                .WORD   LP3
                                .WORD   LP4

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

                                        P$BUFA: .WORD   0               ;LOC. TO HOLD ADDR. OF CMD LINE BUFFER
                                        P$TREE: .WORD   0               ;LOC. TO HOLD ADDR. OF PARSING TREE
```

```
 2563  003136  000000          P$ACT:   .WORD   0          ;LOC. TO HOLD ADDR. OF ACTION ROUTINE
 2564  003140  000000          P$CNT:   .WORD   0          ;LOC. TO BE A COUNTER LOCATION
 2565  003142  000000          P$NUM:   .WORD   0          ;LOC. TO HOLD NUMERIC VALUE FROM PARSE
 2566  003144  000000          P$RADX:  .WORD   0          ;LOC. TO HOLD RADIX USED(LO) AND +/-(HI BYTE)
 2567  003146     000          P$NNUF:  .BYTE   0          ;RETURN =0 IF ENOUGH OF COMMAND FOUND
 2568  003147     000          P$GDBD:  .BYTE   0          ;RETURN CODE 0 IF NO ERROR FOUND
 2569
```

```
2570                                    .SBTTL          MESSAGE BUFFERS AND POINTER TABLES
2571
2572  003150  001000                    TXBUF:  .BLKB   BUFLIM  ;TRANSMITTER BUFFERS
2573  004150  001000                    RXBUF:  .BLKB   BUFLIM  ;RECEIVER BUFFERS
2574  005150  001000                    CMPBUF: .BLKB   BUFLIM  ;COMPARISON BUFFERS
2575  006150  000036                    PTRTAB: .BLKW   MSGLIM*2        ;TABLE FOR MESSAGE ADDRS. & BYTE COUNTS
2576  006244  000036                    PTR13:  .BLKW   MSGLIM*2
2577  006340  000036                    PTR23:  .BLKW   MSGLIM*2
2578  006434                            PTREND:                 ; END OF MSG. PTR. TABLE
2579
2580  006434  000002                            .BLKW   2       ;FILLER FOR OVERFLOW OF RX POINTER TABLE
2581
2582  006440  000000                    RXPTR:  .WORD   0       ;RECEIVER MESSAGE POINTER
2583  006442  000000                    TXPTR:  .WORD   0       ;TRANMITTER BUFFER POINTER
2584  006444  000000                    CMPPTR: .WORD   0       ;COMPARISON BUFFER POINTER
2585  006446  000000                    CMPTOT: .WORD   0       ;CMP MSG TOTAL
2586  006450  000000                    CTOTCC: .WORD   0       ;COMPARE BUFFER CHAR. COUNT
2587  006452  000000                    CCURAD: .WORD   0       ;CURRENT ADDR OF CMP BUFF TO ADD AT
2588
2589  006454  000000                    DVTXA:  .WORD   0       ;DEVICE TX ADDR
2590  006456  000000                    DVTCC:  .WORD   0       ;DEVICE TX CHAR COUNT
2591  006460  000000                    DVTCT:  .WORD   0       ;DEVICE TX MESSAGE COUNT
2592  006462  000000                    TXMTOT: .WORD   0       ;TX MSG TOTAL
2593  006464  000000                    TTOTCC: .WORD   0       ;TX BUFFER CHAR. COUNT
2594  006466  000000                    TCURAD: .WORD   0       ;CURRENT ADDR. OF TX BUFF TO ADD AT
2595
2596  006470  000000                    DVRXA:  .WORD   0       ;DEVICE RX ADDR
2597  006472  000000                    DVRCC:  .WORD   0       ;DEVICE RX CHAR COUNT
2598  006474  000000                    DVRCT:  .WORD   0       ;DEVICE RX MESSAGE COUNT
2599  006476  000000                    RXMTOT: .WORD   0       ;RX MSG TOTAL
2600
2601  006500  000000                    LNCNT:  .WORD   0       ;NUMBER OF OPERATOR AWAKE MSGS
2602  006502  000000                    OPVAR:  .WORD   0       ;OPTTIONAL VARIABLE LOCATION
2603  006504  000000                    PSCNT:  .WORD   0       ;PASS COUNTER
2604  006506  000000                    ERRCNT: .WORD   0       ;ERROR COUNTER
2605  006510  000000                    STADD:  .WORD   0       ;START ADDR.
2606  006512  000000                    ENADD:  .WORD   0       ;END ADDR. FOR DUMP
2607  006514  000000                    BYTBIT: .WORD   0       ;BYTE BIT FOR DUMP ROUTINE
2608
2609                                    ;OTHER MESSSAGE RELATED STORAGE LOCATIONS
2610
2611  006516  000000                    MSGTYP: .WORD   0       ;TYPE OF DATA 0=0'S,1=1'S,2=10'S,3=01'S
2612                                                            ;4=CCITT,5=QUICK FOX,6=ALPHA/NUM,7=OPER
2613  006520  000000                    CURCC:  .WORD   0       ;TX/RX/CMP CHAR COUNT
2614  006522  000000                    CPTRR:  .WORD   0       ;CURRENT RX POINTER
2615  006524  000000                    CPTR:   .WORD   0       ;CURRENT POINTER
2616  006526  000000                    CURADD: .WORD   0       ;CURRENT TX/RX/CMP START ADDD
2617  006530  000000                    TOTCC:  .WORD   0       ;TOTAL CHAR COUNT NOT MORE THEN 'BUFLIM'
2618  006532  000000                    OFSET:  .WORD   0       ;OFFSET COUNT
2619  006534  000000                    TEMP:   .WORD   0       ;TEMPORARY LOCATIONS (USED A LOT)
2620  006536  000000                    TEMP1:  .WORD   0
2621  006540  000000                    TEMP2:  .WORD   0
2622  006542  000000                    TEMP3:  .WORD   0
2623  006544  000000                    TEMP4:  .WORD   0
2624  006546  000000                    TEMP5:  .WORD   0
2625  006550  000000                    CONOTM: .WORD   0       ;CONTROL OUT ERROR MSG. ADDRESS
```

```
 2626  006552  000000  CONTIN: .WORD   0       ;WORD FOR CONTROL IN
 2627  006554     000  GOOD:   .BYTE   0       ;BYTE TO HOLD EXPECTED MESSAGE DATA BYTE FOR ERR REPORT
 2628  006555     000  BAD:    .BYTE   0       ;BYTE TO HOLD RECEIVED MESSAGE DATA BYTE FOR ERR REPORT
 2629
```

```
2630                                    ;MORE INDEPENDENT CODE STORAGE LOCATIONS
2631
2632  006556  000000                    LOGUNT: .WORD   0       ;LOC. TO HOLD LOGICAL UNIT NUMBER
2633  006560  000000                    PCADD:  .WORD   0       ;LOC. HOLD PC OF CALLING ROUTINE
2634  006562  000000                    DCLFLG: .WORD   0       ;CLEANUP & EXIT FLAG -1 = EXIT TEST
2635  006564  000000                    RESFLG: .WORD   0       ;LOC TO HOLD FLAG (-1) THAT A RESTART WAS GIVEN
2636  006566  000000                    MODTYP: .WORD   0       ;DCLT MODE OF OPERATION TYPE
2637                                                            ;  (0=REC-ONLY, 1=TX-ONLY, 2=PASSIVE-LOOPBK,
2638                                                            ;   3=ACTIVE-LOOPBK, 4=DOWN L.L., 5=TALK, 6=LISTEN)
2639  006570  000000                    MLTYP:  .WORD   0       ;MAINTENANCE LOOP TYPE (0=NONE, 1=INTERNAL TTL,
2640                                                            ;  2=CABLE, 3=MODEM-ANALOG LOOPBK (LOCAL),
2641                                                            ;  4=MODEM-DIGITAL LOOPBK (REMOTE), 5=MOP)
2642  006572  000000                    FHDPLX: .WORD   0       ;FULL OR HALF DUPLEX FLAG (1=FULL FROM P-TABLE)
2643  006574  000002                    PARAM:  .WORD   2       ;PROGRAM PARAMETERS
2644                                                            ; BIT0= STATUS MSGS TO OPR PRINTED (1=YES)
2645                                                            ; BIT1= DATA CHECKING DONE ON RCVD MSGS (1=YES)
2646                                                            ; BIT2= ECHO (TRANSMIT) RCV'D MSG.(PASSIVE)(1=YES)
2647                                                            ; BIT3= MODEM STATUS CHECK (1=YES)
2648                                                            ; BIT4= CRC CALC./CHECK DONE (1=YES)
2649                                                            ; BIT5= PROTOCOL EMULATION (1=YES)
2650                                                            ; BIT6= PROTOCOL IS RUNNING
2651                                                            ; BIT7= ABORT PROTOCOL
2652  006576  000000                    RPASS:  .WORD   0       ;PASS NUMBER FROM RUN COMMAND
2653  006600  000000                    FLAG:   .WORD   0       ;DEVICE FLAG WORD
2654
2655                                    ;MODE DISPATCH TABLE
2656  006602  032206                    MODE:   .WORD   RXONLY  ;RX ONLY DISPATCH
2657  006604  032240                            .WORD   TXONLY  ;TX ONLY DISPATCH
2658  006606  032300                            .WORD   PLCK    ;PASSIVE LOOP BACK DISP
2659  006610  032334                            .WORD   ALCK    ;ACTIVE LOOP BACK DISP
2660  006612  033566                            .WORD   DLL     ;DOWN LINE LOAD DISP
2661  006614  033612                            .WORD   TALCK   ;TALK MODE DISPATCH
2662  006616  034056                            .WORD   LISCK   ;LISTEN MODE DISPATCH
2663
2664
2665                                    .SBTTL          CLOCK TABLES, EVENT LOG AND POINTERS
2666  006620  000000                    CLKCSR: .WORD   0       ;CLOCK CSR ADDRESS
2667  006622  000000                    CLKBR:  .WORD   0       ;CLOCK INTERRUPT LEVEL
2668  006624  000000                    CLKVEC: .WORD   0       ;CLOCK INTERRUPT VECTOR
2669  006626  000074                    CLKHZ:  .WORD   60.     ;CLOCK'S HERTZ RATE
2670  006630  000000                    CLKEN:  .WORD   0       ;CLOCK'S CSR VALUE TO INTRPT. ENABLE IT
2671
2672  006632  000000                    TIMMIN: .WORD   0       ;PLACE TO KEEP TIME-SINCE-START
2673  006634  000000                    TIMSEC: .WORD   0
2674  006636  000000                    TIMTCK: .WORD   0       ;PLACE TO KEEP # OF TICKS/SEC
2675
2676  006640  000000                    TIMER1: .WORD   0       ;EVENT TIMER #1 (TICKS)
2677  006642  000000                    TIMER2: .WORD   0       ;EVENT TIMER #2 (TICKS)
2678  006644  000000                    TIMERS: .WORD   0       ;EVENT TIMER #3 (SECONDS)
2679
```

```
 2680
 2681  006646  006650                ;EVENT LOG TABLE AND ITS NEXT ENTRY POINTER
 2682  006650  000341  EVTPTR: .WORD  EVTLOG  ;POINTER TO NEXT FREE SPACE IN EVENT LOG
 2683  007552  000001  EVTLOG: .BLKW  225.    ;EVENT LOG BUFFER
 2684                  EVTEND: .BLKW  1.      ;APPROXIMATE END OF EVENT TABLE (ALLOWS CIRCULAR QUE)
 2685
 2686                  .SBTTL          MODEM DATA SECTION
 2687  007554  000000
 2688                  MODS:   .WORD   0       ;MODEM STATUS
 2689
```

```
                                ;TABLE OF MODEM SIGNAL BIT DEFINITIONS

007556  020000  MOBITS: .WORD   CTS             ;CLEAR TO SEND (CIRCUIT CB)
007560  001000          .WORD   DSR             ;DATA SET READY (CIRCUIT CC)
007562  010000          .WORD   DCD             ;DATA CARRIER DETECT (CIRCUIT CF)
007564  000004          .WORD   RTS             ;REQUEST TO SEND (CIRCUIT CA)
007566  040000          .WORD   RI              ;RING INDICATOR (CIRCUIT CE)
007570  000040          .WORD   SQD             ;SIGNAL QUALITY DETECT (CIRCUIT CG)
007572  000040          .WORD   TM              ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)
007574          MOBITE:

                                ;TABLE OF ADDRESSES OF MODEM SIGNAL MESSAGE POSITIONS

007574  016477  MOMSGS: .WORD   EVMCTS          ;CLEAR TO SEND (CIRCUIT CB)
007576  016503          .WORD   EVMDSR          ;DATA SET READY (CIRCUIT CC)
007600  016507          .WORD   EVMDCD          ;DATA CARRIER DETECT (CIRCUIT CF)
007602  016513          .WORD   EVMRTS          ;REQUEST TO SEND (CIRCUIT CA)
007604  016517          .WORD   EVMRI           ;RING INDICATOR (CIRCUIT CE)
007606  016523          .WORD   EVMSQD          ;SIGNAL QUALITY DETECT (CIRCUIT CG)
007610  016527          .WORD   EVMTM           ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)

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

007612  015101  EVTLST: .WORD   EDTXQ   ;TRANSMIT MESSAGE QUEUED
007614  015125          .WORD   EDTXC   ;TRANSMIT OF MESSAGE COMPLETE
007616  015154          .WORD   EDRXQ   ;RECEIVE MESSAGE SPACE QUEUED
007620  015201          .WORD   EDRXC   ;MESSAGE RECEIVED - RECEIVE COMPLETE
007622  015227          .WORD   EDDER   ;DEVICE INFORMATION
007624  015274          .WORD   EDDVI   ;DEVICE INITIALIZE STARTED
007626  015244          .WORD   EDDCK   ;DATA COMPARISON DONE
007630  015426          .WORD   EDMOS   ;MODEM STATUS CHANGE
007632  015322          .WORD   EDDLE   ;DATA COMPARE LENGTH ERROR
007634  015357          .WORD   EDDDE   ;DATA COMPARE DATA ERROR
007636  015412          .WORD   EDEOP   ;END OF PASS
007640  015452          .WORD   EDABO   ;^ C ABORT

                                ;LOCATIONS USED DURING EVENT REPORTING

007642  000000  EVTSEC: .WORD   0       ;TEMPORARY LOCS TO KEEP EVENT TIME WHILE REPORTING
007644  000000  EVTMIN: .WORD   0
007646  000000  EVTTCK: .WORD   0
007650  000000  EVTADD: .WORD   0       ;TEMP. LOC. TO HOLD ADDRESS DURING EVENT REPORTING
007652  000000  EVTBCT: .WORD   0       ;       "       "  BYTE COUNT   "       "       "
007654  000000  EVTTMP: .WORD   0       ;       "       "  OTHER DATA   "       "       "

                                ;REPORT CODING DISPATCH TABLE

007656  022276  RPTDSP: .WORD   RPTTXQ  ;TRANSMIT QUEUED ENTRY DECODING
007660  022276          .WORD   RPTTXQ  ;TRANSMIT COMPLETE ENTRY DECODING
007662  022276          .WORD   RPTTXQ  ;RECEIVER QUEUED ENTRY DECODING
007664  022276          .WORD   RPTTXQ  ;RECEIVER COMPLETE ENTRY DECODING
007666  022350          .WORD   RPTDER  ;DEVICE ERROR ENTRY DECODING
007670  022444          .WORD   RPTDVI  ;DEVICE INIT ENTRY DECODING
007672  022640          .WORD   RPTDCK  ;DATA COMPARISON ENTRY DECODING
```

```
2746  007674  022714                    .WORD   RPTMSC  ;REPORT MODEM STATUS CHANGE
2747  007676  022640                    .WORD   RPTDLE  ;DATA COMPARISON LENGH ERROR
2748  007700  022564                    .WORD   RPTDDE  ;DATA COMPARISON DATA ERROR
2749  007702  022510                    .WORD   RPTEOP  ;END OF PASS
2750  007704  022510                    .WORD   RPTABO  ;^C ABORT
2751
2752
2753  007706  000000            DEV1:   .WORD   0               ;TEMP LOCS TO HOLD DATA FOR EVENT REPORTING
2754  007710  000000            DEV2:   .WORD   0               ;  AND SHOW MODE,... SUBROUTINE
2755  007712  000000            DEV3:   .WORD   0
2756  007714  000000            DEV4:   .WORD   0
2757
```

```
                        .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) !
                        ; ----------------------

007716                  CLITRE:

                        ;FIRST KEYWORD
007716                          CLI     CLISPA,0,N10$                   ;SKIP ANY LEADING SPACES
007722                  N10$:   CLI     <'?>,HLP,N42$                   ;IS THE FIRST NON-SP CHAR A "?"
007726                          CLI     CLIEXI,0                        ;  IF YES DO "HLP" AND EXIT
007730                  N42$:   CLI     CLISTR,HLP,N43$,<'HELP'>        ;ELSE, IS FIRST WORD A "HELP"
007744                          CLI     CLIEXI,0                        ;  IF YES DO "HLP" AND EXIT
007746                  N43$:   CLI     CLISTR,PRNT,N44$,<'PRINT'>      ;ELSE, IS FIRST WORD A "PRINT"
007762                          CLI     CLIEXI,0                        ;  IF YES DO "PRINT" AND EXIT
007764                  N44$:   CLI     CLISTR,EXIT,N45$,<'EXIT'>       ;ELSE, IS FIRST WORD 'EXIT' ;REV B BY EC
010000                          CLI     CLIEXI,0                        ;  IF YES DO "EXIT" AND EXIT
010002                  N45$:   CLI     CLISTR,RUN,N46$,<'RUN'>         ;ELSE, IS FIRST WORD A "RUN"
010014                          CLI     CLIBR,0,N80$                    ;  IF YES DO "RUN" & GOTO N80$
010020                  N46$:   CLI     CLISTR,NOTNUF,N40$,<'DUMP'>     ;ELSE, IS FIRST WORD A "DUMP"
010034                          CLI     CLIBR,0,N50$                    ;  IF YES GOTO N80$
010040                  N40$:   CLI     CLISTR,CLEAR,N20$,<'CLEAR'>     ;ELSE, IS FIRST WORD A "CLEAR"
010054                          CLI     CLIBR,NOTNUF,N100$              ;  IF YES DO "CLR" & GOTO N100$
010060                  N20$:   CLI     <'S>,NOTNUF,N30$                ;ELSE, IS FIRST CHAR. A "S"
010064                          CLI     CLISTR,SHOW,N25$,<'HOW'>        ;  IF YES IS REST OF WORD "HOW"
010076                          CLI     CLIBR,0,N100$                   ;    IF YES, DO "SHOW",BR N100$
010102                  N25$:   CLI     CLISTR,0,N30$,<'ET'>            ;  ELSE, IS REST OF WORD "ET"
010114                          CLI     CLIBR,0,N110$                   ;    IF YES, DO "SET", BR N110$
010120                  N30$:   CLI     CLIERR,0                        ;OTHERWISE "ILL CMD" - EXIT

                        ;SECOND KEYWORD (MODE=) FOR RUN COMMAND

010122                  N80$:   CLI     CLISPA,0,N30$                   ;SKIP LEADING SPS, IF NONE-ERR
010126                  N81$:   CLI     CLISTR,NOTNUF,N30$,<'MODE'>     ;IS NEXT WORD "MODE="
010142                          CLI     <'=>,0,N30$                     ;  IF NO, IT'S WRONG -ERR -EXIT
010146                          CLI     CLISTR,ATVMOD,N82$,<'ACTIVE'>   ;IS NEXT WORD "ACTIVE"
010164                          CLI     CLIBR,0,N115$                   ;  IF YES, DO "ACTIVE",BR N115$
010170                  N82$:   CLI     CLISTR,PASMOD,N83$,<'PASSIVE'>  ;IS NEXT WORD "PASSIVE"
010206                          CLI     CLIBR,0,N115$                   ;  IF YES, DO "PASSVE",BR N115$
010212                  N83$:   CLI     CLISTR,RECMOD,N84$,<'RECEIVE'>  ;IS NEXT WORD "RECEIVE"
010230                          CLI     CLIBR,0,N115$                   ;  IF YES, DO "RECVE",BR N115$
010234                  N84$:   CLI     CLISTR,LISMOD,N85$,<'LISTEN'>   ;IS NEXT WORD "LISTEN"
010252                          CLI     CLIBR,0,N115$                   ;  IF YES, DO "LISTEN",BR N115$
010256                  N85$:   CLI     CLISTR,DLLMOD,N86$,<'DOWNLINELOAD'>     ;IS NEXT WORD "DOW..."
010302                          CLI     CLIBR,0,N115$                   ;  IF YES, DO "DWNLL",BR N115$
010306                  N86$:   CLI     <'T>,0,N30$                     ;IS NEXT CHAR A "T"
```

```
2814  010312                        CLI     CLISTR,TRAMOD,N87$,<'RANSMIT'>  ; IS REST OF WORD 'RANSMIT'
2815  010330                        CLI     CLIBR,0,N115$                   ;  IF YES, DO 'TRANSM',BR N115$
2816  010334                N87$:   CLI     CLISTR,TALMOD,N30$,<'ALK'>      ; IS REST OF WORD 'ALK'
2817  010346                        CLI     CLIBR,0,N115$                   ;  IF YES, DO 'TALK',BR N115$
2818                                                                        ; IF NO, ERROR - EXIT
2819
2820                                ;SECOND KEYWORD (FOR CLEAR OR SHOW)
2821  010352                N100$:  CLI     CLISPA,0,N30$                   ;SKIP LEADING SPACES, NONE=ERR
2822  010356                N102$:  CLI     CLISTR,CSHEXP,N104$,<'EXPECTBUFF'>      ;IS NEXT WORD 'EXPE...'
2823  010400                        CLI     CLIEXI,0                        ; IF YES, DO CLR-EXP,EXIT
2824  010402                N104$:  CLI     CLISTR,CSHTRN,N30$,<'TRANSMITBUFF'>     ;IS NEXT WORD 'TRANS..'
2825  010426                        CLI     CLIEXI,0                        ; IF YES, DO CLR-TRN,EXIT
2826                                                                        ;IF NO - ERROR - EXIT
2827
2828
2829                                ;SECOND KEYWORD (FOR SET)
2830  010430                N110$:  CLI     CLISPA,0,N30$
2831  010434                N111$:  CLI     CLISTR,SETEXP,N112$,<'EXPECT'>
2832  010452                        CLI     CLIBR,0,N120$
2833  010456                N112$:  CLI     CLISTR,SETTRN,N30$,<'TRANSMIT'>
2834  010476                        CLI     CLIBR,0,N120$
2835
2836                                ;GET ADDRESSES FOR DUMP COMMAND
2837  010502                N50$:   CLI     CLIALP,0,N51$
2838  010506                N51$:   CLI     CLISPA,0,N52$
2839  010512                N52$:   CLI     CLIOCT,DMPS,N30$
2840  010516                        CLI     <'->,NOTNUF,N125$
2841  010522                        CLI     CLIOCT,DMPE,N30$
2842  010526                        CLI     <'/>,NOTNUF,N125$
2843  010532                        CLI     <'B>,DMPQ,N30$
2844  010536                        CLI     CLIBR,0,N125$
2845
2846                                ;QUALIFIERS FOR THE RUN COMMAND
2847  010542                N115$:  CLI     CLIALP,0,N114$
2848  010546                N114$:  CLI     <'/>,NOTNUF,N125$
2849  010552                        CLI     CLISTR,NO,N116$,<'NO'>
2850  010564                N116$:  CLI     <'C>,0,N117$
2851  010570                        CLI     CLISTR,CHECK,N117$,<'HECK'>
2852  010604                        CLI     CLIBR,0,N115$
2853
2854
2855  010610                N117$:  CLI     CLISTR,STATUS,N118$,<'STATUS'>
2856  010626                        CLI     CLIBR,0,N115$
2857  010632                N118$:  CLI     CLISTR,ECHO,N119$,<'ECHO'>
2858  010646                        CLI     CLIBR,0,N115$
2859
2860  010652                N119$:  CLI     <'P>,0,N132$
2861  010656                        CLI     CLISTR,PROTO,N130$,<'ROTOCOL'>
2862  010674                        CLI     CLIBR,0,N115$
2863  010700                N130$:  CLI     CLISTR,0,N30$,<'ASS'>
2864  010712                        CLI     CLIBR,0,N150$
2865
2866  010716                N132$:  CLI     CLISTR,MOSC,N131$,<'MODEM'>
2867  010732                        CLI     CLIBR,0,N115$
2868
2869  010736                N131$:  CLI     CLISTR,0,N30$,<'LOOP'>
```

```
 2870  010752                          CLI     CLIBR,0,N140$
 2871
 2872
 2873  010756                          ;GET MESSAGE TYPE FOR SET MESSAGE COMMANDS
                                       N120$:  CLI     <'=>,0,N30$
 2874
 2875                                  ;   LOOK FOR DEFAULT MESSAGE NAME
 2876  010762                          N60$:   CLI     CLISTR,CMSG1,N61$,<'ONES'>
 2877  010776                                  CLI     CLIBR,0,N121$
 2878  011002                          N61$:   CLI     CLISTR,CMSG0,N62$,<'ZEROES'>
 2879  011020                                  CLI     CLIBR,0,N121$
 2880  011024                          N62$:   CLI     CLISTR,CMSG2,N63$,<'1ALT'>
 2881  011040                                  CLI     CLIBR,0,N121$
 2882  011044                          N63$:   CLI     CLISTR,CMSG3,N64$,<'0ALT'>
 2883  011060                                  CLI     CLIBR,0,N121$
 2884  011064                          N64$:   CLI     CLISTR,CMSG5,N65$,<'ITEP'>
 2885  011100                                  CLI     CLIBR,0,N121$
 2886  011104                          N65$:   CLI     CLISTR,CMSG4,N66$,<'CCITT'>
 2887  011120                                  CLI     CLIBR,0,N121$
 2888  011124                          N66$:   CLI     CLISTR,CMSG6,N67$,<'ALPHA'>
 2889  011140                                  CLI     CLIBR,0,N121$
 2890  011144                          N67$:   CLI     CLISTR,SETET,N68$,<'TRANSMIT'> ;REV B BY EC
 2891  011164                                  CLI     CLIBR,0,N125$
 2892                                  ;   LOOK FOR QUOTED MESSAGE
 2893  011170                          N68$:   CLI     <''>,OPRMSG,N30$
 2894  011174                          N70$:   CLI     <''>,ENDQO,N71$
 2895  011200                                  CLI     CLIBR,0,N121$
 2896  011204                          N71$:   CLI     CLISPA,0,N72$
 2897  011210                          N72$:   CLI     CLIALN,0,N73$           ;ONLY A-Z,SP,TAB, OR 0-9 BETWEEN '"S
 2898  011214                                  CLI     CLIBR,0,N70$
 2899  011220                          N73$:   CLI     CLIERR,BADCHR           ;PRINT ERROR IF NONE LEGAL CHAR FOR '"S
 2900
 2901                                  ;GET QUALIFIERS (SIZE OR COPY) FOR SET MESSAGE COMMANDS
 2902  011222                          N121$:  CLI     CLIALP,0,N123$
 2903  011226                          N123$:  CLI     <'/>,NOTNUF,N125$
 2904  011232                                  CLI     CLISTR,SIZE,N122$,<'SIZE'>
 2905  011246                                  CLI     CLIBR,0,N126$
 2906  011252                          N122$:  CLI     CLISTR,QCOPY,N30$,<'COPY'>
 2907  011266                                  CLI     CLIBR,0,N126$
 2908
 2909                                  ;NUMER FOR SIZE OR COPY
 2910  011272                          N126$:  CLI     <'=>,0,N30$
 2911  011276                                  CLI     CLIDEC,NUM,N30$
 2912  011302                                  CLI     CLIBR,0,N121$
 2913
 2914                                  ;GET MAINTENANCE LOOP TYPE FOR RUN "LOOP" QUIALIFIER
 2915  011306                          N140$:  CLI     <'=>,0,N30$
 2916
 2917
 2918  011312                          N141$:  CLI     CLISTR,TTLLOP,N142$,<'INTERNALTTL'>
 2919  011334                                  CLI     CLIBR,0,N115$
 2920  011340                          N142$:  CLI     CLISTR,CBLLOP,N143$,<'CABLE'>
 2921  011354                                  CLI     CLIBR,0,N115$
 2922  011360                          N143$:  CLI     CLISTR,LMDLOP,N144$,<'LOCALMODEM'>
 2923  011402                                  CLI     CLIBR,0,N115$
 2924  011406                          N144$:  CLI     CLISTR,RMDLOP,N30$,<'REMOTEMODEM'>
 2925  011430                                  CLI     CLIBR,0,N115$
```

```
 2926
 2927                                   ;GET LINE NUMBER FOR "PASS" RUN QUALIFIER
 2928  011434                           N150$:  CLI     <'=>,0,N30$
 2929  011440                                   CLI     CLIDEC,PASC,N30$
 2930  011444                                   CLI     CLIBR,0,N115$
 2931
 2932
 2933
 2934                                   ;END-OF-LINE
 2935  011450                           N125$:  CLI     CLIEXI,0
 2936
```

```
2937
2938
2939                                    ;DEVICE DEPENDENT STORAGE LOCATIONS FOR
2940                                    ;  CURRENT DEVICE PARAMTERS
2941
2942
2943  011452  000000                    RXCSR:  .WORD   0               ;REC CONTROL AND STATUS
2944  011454  000000                    PCSAR:  .WORD   0               ;STATUS REGISTIER
2945  011456  000000                    RDSR:   .WORD   0               ;REC DATA AND STATUS REG
2946  011460  000000                    TXCSR:  .WORD   0               ;TRANSMIT AND REC. CONTROL
2947  011462  000000                    TDSR:   .WORD   0               ;TRANSMIT DATA AND STATUS REG
2948
2949
2950  011464  000000                    INVEC:  .WORD   0               ;INPUT INTERRUPT VECTOR ADDRESS
2951  011466  000000                    OUTVEC: .WORD   0               ;OUTPUT INTERRUPT VECTOR ADDRESS
2952  011470  000000                    INTPRI: .WORD   0               ;INTERRUPT PRIORITY
2953  011472  065626                    DPVP1:  .WORD   065626          ;THIS WORD IS BROKEN DOWN AS FOLLOWS
2954                                                                    ;BITS 0-7 =SYNC WORD
2955                                                                    ;BITS 8-10=ERR DET SELECTED
2956                                    ;THIS IS SET TO SYNC 262        ;BIT11 = IDLE
2957                                    ;CRC 16 INIT TO 1               ;BIT12 = SEC ADDR. MODE
2958                                    ;STRIP SYNC AND BCP MODE        ;BIT13 = STRIP SYNC
2959                                    ;IDLE SET TO MARK               ;BIT14 = PORTO TYPE SEL(1=BCP 0=BOP)
2960                                                                    ;BIT15 = ALL PARTIES ADDRESS..
2961
2962
2963  011474  000000                    CMODS:  .WORD   0               ;CURRENT MODEM
2964  011476  000000                    IRXCSR: .WORD   0               ;IMAGE OF RXCSR
2965  011500  000000                    IRDSR:  .WORD   0               ;IMAGE OR RDSR
2966  011502  000000                    MSGPTR: .WORD   0               ;MSG PTR.FOR HEADER OR CONTROL
2967  011504  000000                    MSGCC:  .WORD   0               ;MSG COUNTER OR CC
2968  011506  000000                    SYNCC:  .WORD   0               ;SYNC CHAR COUNT.
2969  011510  000000                    SYNCW:  .WORD   0               ;SYNC WORD.PLUS TSOM BIT.
2970  011512  000000                    RMSGPT: .WORD   0               ;MSG PTR FOR REC
2971  011514  000000                    RMSGCC: .WORD   0               ;CHAR COUNTER FOR REC
2972  011516  000000                    BCCW:   .WORD   0               ;CRC HOLDING LOC.
2973  011520  000000                    MGLCNT: .WORD   0               ;COUNT OF GLITCH ERRORS
2974  011522  000000                    MHRCNT: .WORD   0               ;COUNT OF HARD ERRORS
2975  011524  000000                    RNODE:  .WORD   0               ;1=REMOTE NODE ITEP,0=NON ITEP
2976
2977
2978                                    ;       ERRTBL
```

```
                        .SBTTL  GLOBAL TEXT SECTION

                        ;++
                        ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
                        ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
                        ; MORE THAN ONE TEST.
                        ;--

                        .SBTTL          DEVICE SUPPORTED
                        ;
                        ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                        ;

                              DEVTYP  <DPV-11>
011526                                                  L$DVTYP::
011526 050104 026526 030461                                     .ASCIZ  /DPV-11/
011534    000
          011536                                                .EVEN

                        .SBTTL          PROGRAM IDENTIFICATION
                        ; TEST DESCRIPTION
                        ;
                              DESCRIPT        <DPV-11 DATA COMM LINK TEST >
011536                                                  L$DESC::
011536 050104 026526 030461                                     .ASCIZ  /DPV-11 DATA COM
011544 042040 052101 020101
011552 047503 046515 046040
011560 047111 020113 042524
011566 052123 000040
                                                                .EVEN
                              .EVEN
```

```
3016                                    .SBTTL          GLOBAL FORMAT STATEMENTS, MESSAGES, AND ASCII INFO
3017
3018  011572  041504  052114  000076  CLI$PM: .ASCIZ  /DCLT>/
      011600  050122  037124     000  CLI$RP: .ASCIZ  /RPT>/          ;REV B BY EC
      011605     045  022516  037501  CLIERM: .ASCIZ  /%N%A?ILL CMD-BAD SYNTX?/
      011635     045  022516  037501  CLINUF: .ASCIZ  /%N%A?INCMPLTE CMD?/
      011660  047045  040445  047077  CLINBG: .ASCIZ  /%N%A?NUM TOO BIG?/
      011702  047045  040445  041077  CLIBRX: .ASCIZ  /%N%A?BAD RADIX?/
      011722  047045  040445  021077  CLIBDL: .ASCIZ  /%N%A?''LOOP'' VALID ONLY IN ACTIVE?/
      011764  047045  040445  021077  CLINPS: .ASCIZ  /%N%A?''ECHO'' VALID ONLY IN PASSIVE?/
      012027     045  022516  037501  CLIBCR: .ASCIZ  /%N%A?ILL CHR- ''A-Z,0-9,SP,TAB'' ONLY?/
      012074  047045  040445  021077  CLISE0: .ASCIZ  /%N%A?''SIZE=0'' NOT VALID?/
      012125     045  022516  037501  CLIPW:  .ASCIZ  /%N%A?TRANSMIT & EXPECT LIST MUST BE IDENTICAL FOR LOOP?/;REV B EC
      012215     045  022516  052101  HLP0:   .ASCIZ  /%N%ATHIS IS DCLT. TYPE ''H'' OR ''?'' FOR DETAILS/
      012273     045  022516  000124  HLPF:   .ASCIZ  /%N%T/
      012300  041504  052114  041440  HLP1:   .ASCIZ  /DCLT CMDS:/
      012313     040  046103  040505  HLP2:   .ASCII  / CLEAR OR SHOW  EXPECTLIST OR TRANSMITLIST/<15><12>
      012367     040  051120  047111          .ASCII  / PRINT/<15><12>
      012377     040  054105  052111          .ASCII  / EXIT/<15><12> ;REV B EC
      012406  042040  046525  020120          .ASCIZ  ? DUMP START-END/B?
      012430  051440  052105  042440  HLP3:   .ASCIZ  ? SET EXPECTMSG OR TRANSMITMSG=TYPE/SIZE=N OR /COPY=N?
      012515     040  042523  020124  HLP3A:  .ASCIZ  / SET EXPECT=TRANSMIT/   ;REV B EC
      012542  020040  052040  050131  HLP4:   .ASCIZ  ?   TYPE=ONES,ZEROES,1ALT,0ALT,ITEP,CCITT,ALPHA?
      012621     040  020040  020040  HLP4A:  .ASCIZ  /      OR ''OPR SPCD=A-Z,SP,TAB,0-9 IN QUOTES''/
      012677     040  052522  020116  HLP5:   .ASCIZ  ? RUN MODE=MTYP/LOOP=LTYP/CHECK,STATUS,ECHO,MODEM,PASS=N?
      012767     040  020040  052115  HLP6:   .ASCII  /   MTYP=TRAN,REC,ACT,PAS,TAL,LIS,DOWN/<15><12>
      013036  020040  046040  054524          .ASCIZ  /   LTYP=INT,CAB,LOC,REM/

      013066  047045  040445  054524  RHLP0:  .ASCIZ  /%N%ATYPE ''H'' OR ''?'' FOR HELP!/ ;REV B EC
      013124  041504  052114  051040  RHLP1:  .ASCIZ  /DCLT REPORT CMDS:/ ;REV B EC
      013146  047514  020107  020055  RHLP2:  .ASCIZ  /LOG - PRINT DCLT EVENT LOG/  ;REV B EC
      013201     105  044530  020124  RHLP3:  .ASCIZ  /EXIT - EXIT REPORT LEVEL/  ;REV B EC
      013232  042510  050114  026440  RHLP4:  .ASCIZ  /HELP - PRINT THIS MESSAGE/  ;REV B EC
      013264  047503  047125  042524  RHLP5:  .ASCIZ  ?COUNTERS/SW - PRINT DDCMP COUNTERS?
      013327     127  042510  042522  RHLP6:  .ASCIZ  ?WHERE /SW=FULL, /ERRORS, /OFFSET=NN(O)?
      013376  047045  040445  043117  RPTIV:  .ASCIZ  /%N%AOFFSET INVALID/
      013421     045  022516  042101  RPTNV:  .ASCIZ  /%N%ADDCMP COUNTERS VALID ONLY WITH PROTOCOL SELECTED./
      013507     045  022516  046501  SHMSG:  .ASCIZ  ?%N%AMSG: TYPE=%T%A/SIZE=%D3?
      013543     132  051105  042517  SHTYP0: .ASCIZ  /ZEROES/
      013552  047117  051505     000  SHTYP1: .ASCIZ  /ONES/
      013557     061  046101  000124  SHTYP2: .ASCIZ  /1ALT/
      013564  040460  052114     000  SHTYP3: .ASCIZ  /0ALT/
      013571     103  044503  052124  SHTYP4: .ASCIZ  /CCITT/
      013577     111  042524  000120  SHTYP5: .ASCIZ  /ITEP/
      013604  046101  044120  000101  SHTYP6: .ASCIZ  /ALPHA/
      013612  050117  020122  050123  SHTYP7: .ASCIZ  /OPR SPEC/
      013623     122  041505  044505  MO0:    .ASCIZ  /RECEIVE/
      013633     124  040522  051516  MO1:    .ASCIZ  /TRANSMIT/
      013644  040520  051523  053111  MO2:    .ASCIZ  /PASSIVE/
      013654  041501  044524  042526  MO3:    .ASCIZ  /ACTIVE/
      013663     104  053517  046116  MO4:    .ASCIZ  /DOWNLINELOAD/
      013700  040524  045514     000  MO5:    .ASCIZ  /TALK/
      013705     114  051511  042524  MO6:    .ASCIZ  /LISTEN/
      013714     000                  LP0:    .ASCIZ  //
      013715     057  047514  050117  LP00:   .ASCIZ  ?/LOOP=?
      013724  047111  042524  047122  LP1:    .ASCIZ  ?INTERNAL?
```

```
013735      103  041101  042514  LP2:    .ASCIZ  ?CABLE?
013743      114  041517  046101  LP3:    .ASCIZ  ?LOCALMODEM?
013756  042522  047515  042524  LP4:    .ASCIZ  ?REMOTEMODEM?
013772  047516                  PNST:   .ASCII  /NO/
013774  052123  052101  051525  PST:    .ASCIZ  /STATUS/
014003      116      117        PNCK:   .ASCII  /NO/
014005      103  042510  045503  PCK:    .ASCIZ  /CHECK/
014013      116      117        PNEC:   .ASCII  /NO/
014015      105  044103  000117  PEC:    .ASCIZ  /ECHO/
014022  047516                  PNMS:   .ASCII  /NO/
014024  047515  042504  000115  PMS:    .ASCIZ  /MODEM/
014032  047516                  PNPR:   .ASCII  /NO/
014034  051120  052117  041517  PPR:    .ASCIZ  /PROTOCOL/

014045      045  022516  046101  LISP:   .ASCIZ  /%N%ALIS>/
014056  046124  037113      000 OPRMM:  .ASCIZ  /TLK>/
014063      124  044510  020123  L5060:  .ASCIZ  /THIS A 50. OR 60. HZ. LSI-11:/
        014122                          .EVEN


                                ;
                                ; FORMAT STATEMENTS USED IN PRINT CALLS
                                ;

014122  047045  040445  047504  DLLCM:  .ASCIZ  /%N%ADOWN LINE LOAD NOT SUPPORTED BY THIS DEVICE/

014202  047045  040445  046103  BDCLK:  .ASCIZ  /%N%ACLOCK NOT FOUND/
014226  047045  040445  040502  NOCLK:  .ASCIZ  /%N%ABAD CLOCK - PROGRAM WILL HANG ON ''TIMEOUT''!!/
014307      115  054101  020056  TABEX:  .ASCIZ  /MAX. CHAR. MSG COUNT EXCEEDED -/
014347      102  043125  042506  BUFEX:  .ASCIZ  /BUFFER FULL -/
014365      045  022516  022524  MSGTRN: .ASCIZ  /%N%T%A MSG. NOT BUILT !!/
014416  047045  040445  044103  MSGTRU: .ASCIZ  /%N%ACHAR. COUNT EXCEEDS BUFF LIMIT - MSG TRUNCATED/
014501      045  022516  032523  SHF0:   .ASCIZ  ?%N%S5%AMODE=%T%T%T%A/PASS=%Z5?

014537      045  022516  032523  SHF1:   .ASCIZ  ?%N%S5%S5%S5%S5%A/%T%A/%T%A/%T%A/%T%A/%T%A/%T?

014604  051445  022465  052101  EFM2:   .ASCIZ  /%S5%ATOTAL MISMATCHES IN MSG = %D5/
014647      045  022516  031523  PCPM:   .ASCIZ  /%N%S3%ACALLED FROM PC=%O6/
014701      045  032523  040445  EFM11:  .ASCIZ  /%S5%ACOMPARE COUNT=%D5%S3%ARECEIVE COUNT=%D5/
014756  047515  042504  020115  MSCMS:  .ASCIZ  /MODEM STATUS CHANGES FOR THIS PASS WERE../
015030  051445  022465  044101  EFM13:  .ASCIZ  /%S5%AHARD CHANGES=%D5%A%S3%AGLITCHES=%D5/

                                ;EVENT DESCRIPTION MESSAGES

015101      124  040522  051516  EDTXQ:  .ASCIZ  /TRANSMIT MSG QUEUED/
015125      124  040522  051516  EDTXC:  .ASCIZ  /TRANSMIT MSG COMPLETED/
015154  042522  042503  053111  EDRXQ:  .ASCIZ  /RECEIVE SPACE QUEUED/
015201      122  041505  044505  EDRXC:  .ASCIZ  /RECEIVE MSG COMPLETED/
015227      104  053105  041511  EDDER:  .ASCIZ  /DEVICE ERROR/
015244  040504  040524  041440  EDDCK:  .ASCIZ  /DATA COMPARISON STARTED/
015274  042504  044526  042503  EDDVI:  .ASCIZ  /DEVICE INIT AND SETUP/
```

```
015322  040504  040524  041440  EDDLE:  .ASCIZ  /DATA COMPARISON LENGTH ERROR/
015357     104  052101  020101  EDDDE:  .ASCIZ  /DATA COMPARISON DATA ERROR/
015412  047105  020104  043117  EDEOP:  .ASCIZ  /END OF PASS/
015426  047515  042504  020115  EDMOS:  .ASCIZ  /MODEM STATUS CHANGE/
015452  041536  040440  047502  EDABO:  .ASCIZ  /^C ABORT/

                                ;EVENT REPORTING MESSAGES
015463     045  031523  047445  BASM3:  .ASCIZ  /%S3%03/
015472  051445  022463  033117  BASM2:  .ASCIZ  /%S3%06/
015501     045  022516  033117  BASM1:  .ASCIZ  /%N%06/
015507     045  022516  052101  NULEVT: .ASCIZ  /%N%ATHE DCLT EVENT LOG IS EMPTY/
015547     045  022516  037101  EVTF0:  .ASCIZ  /%N%A>>> DCLT EVENT LOG ENTRY <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<</
015643     045  022516  032504  EVTF1:  .ASCIZ  /%N%D5%A:%Z2%A:%Z2%S3%T/
015672  047045  051445  022463  EVTF2:  .ASCIZ  /%N%S3%AADDR OF MSG=%06%S3%ABYTE COUNT=%D5/
015744  047045  051445  022463  EVTF3:  .ASCIZ  /%N%S3%T%N/
015756  051445  022463  033117  EVTF3C: .ASCIZ  /%S3%06%S3%06/
015773     045  031523  047445  EVTF3D: .ASCIZ  /%S3%06%S3%06%S3%T/
016015     045  022516  031523  EVTF4:  .ASCIZ  /%N%S3%AADDR OF MSG=%06%S3%ABYTE COUNT=%D5%S3%ANO. OF CMP ERRS=%D5/
016117     045  022516  031523  EVTF4A: .ASCIZ  /%N%S3%AADDR OF MSG=%06%S3%ARX BYTES=%D5%S3%ACOMPARE BYTES=%D5/

016215     045  022516  031523  EVTF4B: .ASCIZ  /%N%S3%APASS=%D5%S3%AERRORS=%D5%S3%ASTRT-TO=%D5/

016274  051445  022465  041101  EVTF5A: .ASCIZ  /%S5%ABYTE# IN MSG.=%D5%S3%AEXPTD=%03%S3%ARECVD=%03/

016357     045  022516  034523  EVMOCG: .ASCIZ  /%N%S9%ACHANGED TO:/

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

016402  047045  051445  022470  EVMOHD: .ASCIZ  /%N%S8%AMODEM STATUS: CTS DSR DCD RTS RI  SQD TM/
016452  047045  051445  022471  EVMOST: .ASCII  /%N%S9%S9%S5%A/
016477     130     040     040  EVMCTS: .BYTE   'X,40,40,40
016503     130     040     040  EVMDSR: .BYTE   'X,40,40,40
016507     130     040     040  EVMDCD: .BYTE   'X,40,40,40
016513     130     040     040  EVMRTS: .BYTE   'X,40,40,40
016517     130     040     040  EVMRI:  .BYTE   'X,40,40,40
016523     130     040     040  EVMSQD: .BYTE   'X,40,40,40
016527     130     040     040  EVMTM:  .BYTE   'X,40,40,40
016533     000                          .BYTE   0
                                        .EVEN

                                ;EXECUTION STATUS MESSAGES TO BE PRINTED TO KEEP OPERATOR AWAKE
016534  047045     000          CR:     .ASCIZ  /%N/            ;CR FOR LINES IN A ROW
016537     045  031523  040445  STXQ:   .ASCIZ  /%S3%ATXQ/      ;ABOUT TO TRANSMIT
016550  051445  022463  052101  STXC:   .ASCIZ  /%S3%ATXC/      ;TX COMPLETED
016561     045  031523  040445  SRXQ:   .ASCIZ  /%S3%ARXQ/      ;ABOUT TO RECEIVE
016572  051445  022463  042501  SDVE:   .ASCIZ  /%S3%AERR/      ;DEVICE ERROR
016603     045  031523  040445  SCM:    .ASCIZ  /%S3%ACMP/      ;ABOUT TO DO DATA CHECKING OF RECVD VS. EXPTD
016614  051445  022463  044501  SDVI:   .ASCIZ  /%S3%AINI/      ;DEVICE ABOUT TO BE INITIALIZED
016625     045  031523  040445  SCML:   .ASCIZ  /%S3%ACML/      ;COMPARE LENGTH ERROR
016636  051445  022463  041501  SCMD:   .ASCIZ  /%S3%ACMD/      ;COMPARE DATA ERROR
016647     045  031523  040445  SEOP:   .ASCIZ  /%S3%AEOP/      ;END  OF PASS
                                        .EVEN

016660  051445  022463  046501  SMSC:   .ASCIZ  /%S3%AMSC/      ;MODEM STATUS CHANGE.
```

```
016671     115  042117  046505  GLMSG:  .ASCIZ  /MODEM STATUS GLITCHED/
016717     115  042117  046505  HRDMSG: .ASCIZ  /MODEM STATUS HARD ERROR/
```

```
016747     115  051501  042524  DVEM0:  .ASCII  /MASTER RESET DID NOT WORK/
017000  005015  020040  051040          .ASCIZ  <15><12>/   RXCSR     TXCSR   /
017026  047516  041440  042514  DVEM1:  .ASCII  /NO CLEAR TO SEND FROM MODEM /
017062  005015  020040  051040          .ASCIZ  <15><12>/   RXCSR     TXCSR   /
017110  044524  042515  047440  DVEM2:  .ASCII  /TIME OUT WAITING FOR RX OR TX TO COMPLETE/
017161     015  020012  020040          .ASCIZ  <15><12>/   RXCSR     TXCSR/
017205     103  041522  044440  DVEM3:  .ASCII  /CRC IN ERROR/
017221     015  020012  020040          .ASCIZ  <15><12>/   RDSR      RXCSR/
017245     122  041505  044505  DVEM4:  .ASCII  /RECEIVER OVERRUN/
017265     015  020012  020040          .ASCIZ  <15><12>/   RDSR      RXCSR/
017311     124  046511  042105  DVEM5:  .ASCII  /TIMED OUT IN START,STACK,ACK SEQ/
017351     015  020012  020040          .ASCIZ  <15><12>/   RDATA     SDATA/
017375     115  042117  046505  DVEM6:  .ASCII  /MODEM DID NOT RETURN MODEM READY/
017435     015  020012  020040          .ASCIZ  <15><12>/   RXCSR      TXCSR/

                                        .EVEN
```

```
                                .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.
                                ;--


                                .LIST BEX

017462                                  BGNMSG  ERR1
017462                                                                  ERR1::
017462                                  PRINTB  #EVTF5A,OFSET,<B,GOOD>,<B,BAD>  ;INDIVIDUAL DATA COMPARE ERROR
017462  005046                                                                  CLR     -(SP)
017464  153716  006555                                                          BISB    BAD,(SP)
017470  005046                                                                  CLR     -(SP)
017472  153716  006554                                                          BISB    GOOD,(SP)
017476  013746  006532                                                          MOV     OFSET,-(SP)
017502  012746  016274                                                          MOV     #EVTF5A,-(SP)
017506  012746  000004                                                          MOV     #4,-(SP)
017512  010600                                                                  MOV     SP,R0
017514  104414                                                                  TRAP    C$PNTB
017516  062706  000012                                                          ADD     #12,SP
017522                                  ENDMSG
017522                                                                  L10001:
017522  104423                                                                  TRAP    C$MSG

017524                                  BGNMSG  ERR2
017524                                                                  ERR2::
017524                                  PRINTB  #EFM2,TEMP4                     ;TOTAL DATA COMPARE FAILS ERROR
017524  013746  006544                                                          MOV     TEMP4,-(SP)
017530  012746  014604                                                          MOV     #EFM2,-(SP)
017534  012746  000002                                                          MOV     #2,-(SP)
017540  010600                                                                  MOV     SP,R0
017542  104414                                                                  TRAP    C$PNTB
017544  062706  000006                                                          ADD     #6,SP
017550                                  ENDMSG
017550                                                                  L10002:
017550  104423                                                                  TRAP    C$MSG

017552                                  BGNMSG  ERR10
017552                                                                  ERR10::
017552                                  PRINTB  #EFM11,R4,TEMP3                 ;LENGH COMPARISON ERROR
017552  013746  006542                                                          MOV     TEMP3,-(SP)
017556  010446                                                                  MOV     R4,-(SP)
017560  012746  014701                                                          MOV     #EFM11,-(SP)
017564  012746  000003                                                          MOV     #3,-(SP)
017570  010600                                                                  MOV     SP,R0
017572  104414                                                                  TRAP    C$PNTB
017574  062706  000010                                                          ADD     #10,SP
017600                                  ENDMSG
017600                                                                  L10003:
017600  104423                                                                  TRAP    C$MSG
```

```
      017602                            BGNMSG  ERR4
      017602                                                                  ERR4::
      017602                            PRINTB  #EFM13,MHRCNT,MGLCNT       ;MODEM STATUS CHANGE
      017602  013746  011520                                                          MOV     MGLCNT,-(SP)
      017606  013746  011522                                                          MOV     MHRCNT,-(SP)
      017612  012746  015030                                                          MOV     #EFM13,-(SP)
      017616  012746  000003                                                          MOV     #3,-(SP)
      017622  010600                                                                  MOV     SP,R0
      017624  104414                                                                  TRAP    C$PNTB
      017626  062706  000010                                                          ADD     #10,SP
      017632                            ENDMSG
      017632                                                                  L10004:
      017632  104423                                                                  TRAP    C$MSG



                                        ;
                                        ;PRINT THE 2 OCTAL #'S IN TEMP3/4
                                        ;

      017634                            BGNMSG  ERR13
      017634                                                                  ERR13::
      017634                            PRINTB  #EVTF3C,TEMP3,TEMP4
      017634  013746  006544                                                          MOV     TEMP4,-(SP)
      017640  013746  006542                                                          MOV     TEMP3,-(SP)
      017644  012746  015756                                                          MOV     #EVTF3C,-(SP)
      017650  012746  000003                                                          MOV     #3,-(SP)
      017654  010600                                                                  MOV     SP,R0
      017656  104414                                                                  TRAP    C$PNTB
      017660  062706  000010                                                          ADD     #10,SP
      017664                            ENDMSG
      017664                                                                  L10005:
      017664  104423                                                                  TRAP    C$MSG


                                        ;
                                        ;PRINT THE 2 OCTAL #'S IN TEMP3/4
                                        ; AND THE MESG. WHOSE ADDR. IS IN CONOTM
                                        ;

      017666                            BGNMSG  ERR14
      017666                                                                  ERR14::
      017666                            PRINTB  #EVTF3D,TEMP3,TEMP4,CONOTM
      017666  013746  006550                                                          MOV     CONOTM,-(SP)
      017672  013746  006544                                                          MOV     TEMP4,-(SP)
      017676  013746  006542                                                          MOV     TEMP3,-(SP)
      017702  012746  015773                                                          MOV     #EVTF3D,-(SP)
      017706  012746  000004                                                          MOV     #4,-(SP)
      017712  010600                                                                  MOV     SP,R0
      017714  104414                                                                  TRAP    C$PNTB
      017716  062706  000012                                                          ADD     #12,SP
      017722                            ENDMSG
      017722                                                                  L10006:
      017722  104423                                                                  TRAP    C$MSG

      017724                            EXIT    MSG
```

```
  3119  017724  000167                                  .WORD   J$JMP
  3120  017726  177772                                  .WORD   L10006-2-.
  3121
  3122
```

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

                                ;++
                                ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
                                ; THAT ARE USED IN MORE THAN ONE TEST.
                                ;--


                                .SBTTL          CLOCK SETUP SUBROUTINE

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       THIS SUBROUTINE SETS UP THE CLOCK INFORMATION TABLE FOLLOWING A "CLOCK"
                                ;       CALL EXECUTED IN THE INITIALIZATION CODE. BUT SINCE THE "CLOCK" CALL
                                ;       SAYS NOTHING ABOUT AN LSI-11'S CLOCK, THIS ROUTINE IS ONLY USED IF A
                                ;       LINE OR P-CLOCK IS FOUND.
                                ;
                                ; INPUTS:
                                ;       R1= POINTS TO SUPERVISOR SPACE WHERE CLOCK INFO WAS RETURNED
                                ;       R2= POINTS TO "CLK" TABLE WHERE CLOCK INFO WILL BE KEPT
                                ;
                                ; IMPLICIT INPUTS:
                                ;       THE SUPERVISOR  SPACE WHERE CLOCK INFO WAS RETURNED BY THE "CLOCK" CALL
                                ;
                                ; OUTPUTS:
                                ;       "CLKCSR" GETS LOADED WITH THE CLOCK'S CSR ADDRESS
                                ;       "CLKBR" GETS LOADED WITH THE CLOCK'S INTERRUPT LEVEL
                                ;       "CLKVEC" GETS LOADED WITH THE CLOCK'S INTERRUPT VECTOR
                                ;       "CLKHZ" GETS LOADED WITH THE LINE FREQ. (HERTZ RATE) WHICH DETERMINES
                                ;               THE NUMBER OF TICKS IN A SECOND
                                ;
                                ; CALLING SEQUENCE:
                                ;       JSR     PC,CLKSET               ;CALL CLOCK SETUP WITH R1 & R2 SETUP
                                ;--

017730                          CLKSET:
017730  012122                          MOV     (R1)+,(R2)+             ;LOAD CLOCK'S CSR ADDR. INTO "CLKCSR"
017732  012112                          MOV     (R1)+,(R2)              ;LOAD CLOCK'S INT. LEVEL INTO "CLKBR"
017734  006312                          ASL     (R2)                    ;ADJUST THE INT. LEVEL FOR LOADING INTO
017736  006312                          ASL     (R2)                    ;   THE PSW WITH A "SETVEC" CALL
017740  006312                          ASL     (R2)
017742  006312                          ASL     (R2)
017744  006322                          ASL     (R2)+
017746  012122                          MOV     (R1)+,(R2)+             ;LOAD CLOCK'S INT. VECTOR INTO "CLKVEC"
017750  012122                          MOV     (R1)+,(R2)+             ;LOAD CLOCK'S HERTZ RATE INTO "CLKHZ"
017752  000207                          RTS     PC
```

```
3170
3171                                    .SBTTL          CLOCK INTERRUPT SERVICE ROUTINE
3172                                    ;++
3173                                    ; FUNCTIONAL DESCRIPTION:
3174                                    ;       THIS IS THE CLOCK INTERRUPT SERVICE ROUTINE WHICH TAKES CARE OF
3175                                    ;       KEEPING THE "TIME-SINCE-START" AND COUNTING DOWN ANY OF THE
3176                                    ;       "EVENT" TIMERS. THE TIMERS ARE USED TO TIME COMPLETION OF  DEVICE
3177                                    ;       REQUESTS. THE "TIME-SINCE-START" IS USED TO BE LOGGED WITH EACH ENTRY
3178                                    ;       INTO THE EVENT LOG.
3179                                    ;
3180                                    ; IMPLICIT INPUTS:
3181                                    ;       TIMTCK: THE CURRENT NO. OF TICKS LEFT TO BE COUNTED UNTIL A SECOND
3182                                    ;               HAS BEEN COUNTED OFF
3183                                    ;       CLKHZ:  THE NO. OF TICKS IN A SECOND, DETERMINED BY THE SYS. LINE FREQ.
3184                                    ;       TIMMIN & TIMSEC: CURRENT VALUE OF "TIME-SINCE-START"
3185                                    ;                               IN MINUTES & SECONDS
3186                                    ;       TIMER 1,2, & S: CURRENT VALUES OF THE "EVENT TIMERS"
3187                                    ;
3188                                    ; IMPLICIT OUTPUTS:
3189                                    ;       NEW VALUE OF EVENT TIMER "1" DECREMENTED BY 1 TICK IF IT WAS NON-ZERO
3190                                    ;       NEW VALUE OF EVENT TIMER "2" DECREMENTED BY 1 TICK IF IT WAS NON-ZERO
3191                                    ;       NEW VALUE OF EVENT TIMER "S" DECREMENTED BY 1 SECOND IF IT WAS NON-ZERO
3192                                    ;
3193                                    ; FUNCTIONAL SIDE EFFECTS:
3194                                    ;       THE CLOCK IS DISABLED UPON ENTRY AND REENABLED WHEN LEAVING
3195                                    ;
3196                                    ; CALLING SEQUENCE:
3197                                    ;       THIS ROUTINE IS CALLED WHEN THE CLOCK INTERRUPTS THRU "CLKVEC".
3198                                    ;       THE ADDRESS OF THIS ROUTINE WAS LOADED INTO THE CLOCK'S INTERRUPT
3199                                    ;       VECTOR WITH A SUPERVISOR "SETVEC" CALL.
3200                                    ;--
3201
3202  017754                            BGNSRV  CLKINT
3203  017754                                                                    CLKINT::
3204
3205  017754  005077  166640            CLR     @CLKCSR         ;DISABLE THE CLOCK FORM INTERRUPTING
3206  017760  005337  006636            DEC     TIMTCK          ;DECREMENT THE # OF TICKS/SEC.
3207  017764  001015                    BNE     1$              ;GO CHECK TIMERS (1&2-TICKS, 3-SECONDS)
3208  017766  013737  006626  006636    MOV     CLKHZ,TIMTCK    ;RESET THE # OF TICKS/SEC.
3209  017774  005237  006634            INC     TIMSEC          ;INC # OF SECS-SINCE-START
3210  020000  022737  000074  006634    CMP     #60.,TIMSEC     ;SEE IF WE'VE COUNTED 60 SECS. YET
3211  020006  001004                    BNE     1$              ;IF NOT, GO CHECK TIMERS
3212  020010  005237  006632            INC     TIMMIN          ; ELSE INC MINUTES-SINCE-START
3213  020014  005037  006634            CLR     TIMSEC          ; AND RESTART SECOND COUNTER
3214
3215  020020  005737  006640    1$:     TST     TIMER1          ;SEE IF TIMER #1, TIMING ANYTHING
3216  020024  001402                    BEQ     2$              ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
3217  020026  005337  006640            DEC     TIMER1          ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
3218  020032  005737  006642    2$:     TST     TIMER2          ;SEE IF TIMER #2, TIMING ANYTHING
3219  020036  001402                    BEQ     3$              ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
3220  020040  005337  006642            DEC     TIMER2          ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
3221  020044  005737  006644    3$:     TST     TIMERS          ;SEE IF TIMER #3, TIMING ANYTHING
3222  020050  001406                    BEQ     4$              ; IF=0, NOTHING BEING TIMED, LEAVE
3223  020052  023737  006626  006636    CMP     CLKHZ,TIMTCK    ;SEE IF A SECOND HAS BEEN COUNTED OFF
3224  020060  001002                    BNE     4$              ; BR IF NO
3225  020062  005337  006644            DEC     TIMERS          ; ELSE DECREMENT THE TIMER VALUE (BY 1 SEC.)
```

```
3226  020066  013777  006630  166524  4$:     MOV     CLKEN,@CLKCSR   ;REENABLE THE CLOCK TO INTERRUPT
3227  020074                                  ENDSRV
3228  020074                                                                          L10007:
3229  020074  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
                                ;--
                                
020076                          LOGTXQ:
020076  012737  016537  006536          MOV     #STXQ,TEMP1     ;SET UP MSG. TO PRINT
020104  012737  000000  006534          MOV     #TXQ,TEMP       ;SET UP EVENT TYPE
020112  000517                          BR      LOGS1           ;GO LOG EVENT AND TIME

020114                          LOGTXC:
020114  012737  016550  006536          MOV     #STXC,TEMP1     ;SET UP MSG. TO PRINT
020122  012737  000002  006534          MOV     #TXC,TEMP       ;SET UP EVENT TYPE
020130  000510                          BR      LOGS1           ;GO LOG EVENT AND TIME

020132                          LOGRXQ:
020132  012737  016561  006536          MOV     #SRXQ,TEMP1     ;SET UP MSG. TO PRINT
020140  012737  000004  006534          MOV     #RXQ,TEMP       ;SET UP EVENT TYPE
020146  000501                          BR      LOGS1           ;GO LOG EVENT AND TIME

020150                          LOGRXC:
020150  012737  000006  006534          MOV     #RXC,TEMP       ;SET UP EVENT TYPE
020156  000475                          BR      LOGS1           ;GO LOG EVENT AND TIME
020160                          LGDVE:
```

```
3286  020160  012737  016572  006536          MOV     #SDVE,TEMP1     ;SET UP MSG. TO PRINT
3287  020166  012737  000010  006534          MOV     #DER,TEMP       ;SET UP EVENT TYPE
3288  020174  000503                          BR      LOGS3           ;GO LOG EVENT AND TIME
3289
3290  020176                          LOGDVI:
3291  020176  012737  016614  006536          MOV     #SDVI,TEMP1     ;SET UP MSG. TO PRINT
3292  020204  012737  000012  006534          MOV     #DVI,TEMP       ;SET UP EVENT TYPE
3293  020212  113737  006566  006540          MOVB    MODTYP,TEMP2
3294  020220  113737  006570  006541          MOVB    MLTYP,TEMP2+1
3295  020226  013737  006576  006542          MOV     RPASS,TEMP3
3296  020234  013737  006574  006544          MOV     PARAM,TEMP4     ;SET UP EVNT ENTRIES
3297  020242  000460                          BR      LOGS3           ;GO LOG EVENT AND TIME
3298
3299  020244                          LOGCMP:
3300  020244  012737  016603  006536          MOV     #SCM,TEMP1      ;SET UP MSG. TO PRINT
3301  020252  012737  000014  006534          MOV     #DCK,TEMP       ;SET UP EVENT TYPE
3302  020260  000451                          BR      LOGS3
3303  020262                          LOGCML:
3304  020262  012737  016625  006536          MOV     #SCML,TEMP1
3305  020270  012737  000020  006534          MOV     #DLE,TEMP       ;SET UP MSG. AND TYPE
3306  020276  000442                          BR      LOGS3           ;GO LOG EVENT AND TIME
3307  020300                          LOGCMD:
3308  020300  012737  016636  006536          MOV     #SCMD,TEMP1
3309  020306  012737  000022  006534          MOV     #DDE,TEMP
3310  020314  000433                          BR      LOGS3           ;GO LOG MSG TYPE AND TIME
3311  020316                          LOGEOP:
3312  020316  012737  016647  006536          MOV     #SEOP,TEMP1
3313  020324  012737  000024  006534          MOV     #EOP,TEMP
3314  020332  000424                          BR      LOGS3           ;GO LOG MSG TYPE AND TIME
3315
3316
3317  020334                          LOGMSC:
3318  020334  012737  016660  006536          MOV     #SMSC,TEMP1
3319  020342  012737  000016  006534          MOV     #MSC,TEMP
3320  020350  000415                          BR      LOGS3
3321
3322
3323  020352  013746  006506          LOGS1:  MOV     ERRCNT,-(SP)    ;SAVE CURRENT ERROR COUNT
3324  020356  004737  035124                  JSR     PC,DVMODS       ; GO GET MODEM STATUS
3325  020362  012604                          MOV     (SP)+,R4        ;GET SAVED ERRCNT VALUE
3326  020364  020437  006506                  CMP     R4,ERRCNT       ;WHERE ANY ERRORS FOUND
3327  020370  001402                          BEQ     1$              ; BR IF NONE
3328  020372  000137  020606                  JMP     LOGEX           ; ELSE, LEAVE WITHOUT LOGGING ANYTHING
3329                                                                  ;   BUT THE DEVICE ERROR FROM 'DVMODS'
3330  020376  013737  007554  006544  1$:     MOV     MODS,TEMP4      ;AND PUT IT IN TEMP4
3331
3332  020404                          LOGS3:
3333  020404  022737  000006  006534          CMP     #RXC,TEMP
3334  020412  001434                          BEQ     LOGS5           ;IF RXC DONT PRINT
3335  020414  032737  000001  006574          BIT     #STATB,PARAM
3336  020422  001430                          BEQ     LOGS5           ;IF NO STATUS SELECTED
3337                                                                  ;GO TO 5
3338
3339  020424  022737  000010  006500          CMP     #10,LNCNT       ;HAVE WE DONE 10?
3340  020432  001012                          BNE     LOGS4           ;IF NOT GO TO 4
3341  020434  005037  006500                  CLR     LNCNT           ;ESLE CLEAR IT
```

```
3342
3343   020440
3344   020440  012746  016534          PRINTF  #CR             ;ELSE PRINT CR
3345   020444  012746  000001                                          MOV     #CR,-(SP)
3346   020450  010600                                                 MOV     #1,-(SP)
3347   020452  104417                                                 MOV     SP,R0
3348   020454  062706  000004                                         TRAP    C$PNTF
3349   020460                                                         ADD     #4,SP
3350   020460  005237  006500  LOGS4:  INC     LNCNT           ;INC COUNTER OF # OF AWAKE MSGS
3351   020464                  PRINTF  TEMP1           ;PRINT OPERATOR AWAKE MSG.
3352   020464  013746  006536                                         MOV     TEMP1,-(SP)
3353   020470  012746  000001                                         MOV     #1,-(SP)
3354   020474  010600                                                 MOV     SP,R0
3355   020476  104417                                                 TRAP    C$PNTF
3356   020500  062706  000004                                         ADD     #4,SP
3357   020504  010346  LOGS5:  MOV     R3,-(SP)        ;SAVE R3 ON THE STACK
3358   020506  013703  006646          MOV     EVTPTR,R3
3359   020512  113723  006534          MOVB    TEMP,(R3)+      ;LOG EVENT
3360   020516  013737  006626  006534  MOV     CLKHZ,TEMP
3361   020524  163737  006636  006534  SUB     TIMTCK,TEMP
3362   020532  113723  006534          MOVB    TEMP,(R3)+      ;LOG TIME SINCE START
3363   020536  113723  006634          MOVB    TIMSEC,(R3)+
3364   020542  113723  006632          MOVB    TIMMIN,(R3)+    ;TICKS,SECS AND MINS.
3365   020546  013723  006540          MOV     TEMP2,(R3)+     ;LOG EVNT ENTRY 3
3366   020552  013723  006542          MOV     TEMP3,(R3)+     ;LOG EVNT ENTRY 4
3367   020556  013723  006544          MOV     TEMP4,(R3)+     ;LOG EVNT ENTRY 5
3368   020562  020327  007552          CMP     R3,#EVTEND
3369   020566  103404                  BLO     LOGS2           ;IF EVENT LOG FULL GO
3370                                                          ;CONTINUE;ELSE GO TO 2
3371   020570  012713  177777          MOV     #-1,(R3)        ;LOG A TABLE END
3372   020574  012703  006650          MOV     #EVTLOG,R3      ;PUT R3 TO START OF TABLE
3373   020600  010337  006646  LOGS2:  MOV     R3,EVTPTR       ;RESTORE POINTER
3374   020604  012603                  MOV     (SP)+,R3        ;RESTORE R3
3375   020606  000207  LOGEX:  RTS     PC
3376
```

```
                        .SBTTL  REPORT EVENT LOG
                                ;;DPV DCLT PROGRAM
                                ;;RPT> LOG
                                ;;     HELP
                                ;;     EXIT
                                ;;     COUNTER/FULL,ERROR,OFFSET=NN(0)

                        REPORT: MOV     R2,-(SP)        ;SAVE R2,R3,R4 ON THE STACK
                                MOV     R3,-(SP)
                                MOV     R4,-(SP)

                                ;PRINT HELP MESSAGE
                                PRINTF  #RHLP0          ;BASIC HELP MESSAGE
                                                                        MOV     #RHLP0,-(SP)
                                                                        MOV     #1,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTF
                                                                        ADD     #4,SP

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

                                ;PRINT PROMPT 'RPT>'
                                GMANID  CLI$RP,CMDBUF,A,-1,1,72.,NO
                                                                        TRAP    C$GMAN
                                                                        BR      10000$
                                                                        .WORD   CMDBUF
                                                                        .WORD   T$CODE
                                                                        .WORD   CLI$RP
                                                                        .WORD   -1
                                                                        .WORD   T$LOLIM
                                                                        .WORD   T$HILIM
                                                                10000$:
020666 012737 002666 003132             MOV     #CMDBUF,P$BUFA  ;INPUT BUFFER
020674 012737 021030 003134             MOV     #CLIRT,P$TREE   ;REPORT CLI TREE
020702 012737 021232 003136             MOV     #CLIRAC,P$ACT   ;ACTION ROUTINES
020710 005037 003012                    CLR     QUALFG
020714 004737 024244                    JSR     PC,P$TRV        ;GO PARSE COMMAND LINE
020720 105737 003147                    TSTB    P$GDBD          ;COMMAND OK ?
020724 001412                           BEQ     1$              ;YES,BRANCH
020726                                  PRINTF  #CLIERM         ;PRINT INVALID INPUT MESSAGE
                                                                        MOV     #CLIERM,-(SP)
                                                                        MOV     #1,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTF
                                                                        ADD     #4,SP
020746 000137 020636                    JMP     GETRCL          ;TRY AGAIN

020752 105737 003146            1$:     TSTB    P$NNUF          ;MORE COMMAND NEEDED ?
020756 001412                           BEQ     10$             ;NO,BRANCH
020760                                  PRINTF  #CLINUF         ;INCOMPLETE MESSAGE
                                                                        MOV     #CLINUF,-(SP)
                                                                        MOV     #1,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTF
                                                                        ADD     #4,SP
```

```
 3433  021000  000137  020636                 JMP     GETRCL                  ;TRY AGAIN
 3434
 3435  021004  023727  003010  000002  10$:   CMP     KEYWD1,#RPEXT           ;EXIT COMMAND ?
 3436  021012  001402                         BEQ     20$                     ;YES,BRANCH
 3437  021014  000137  020636                 JMP     GETRCL                  ;GET ANOTHER COMMAND
 3438  021020  012604                 20$:    MOV     (SP)+,R4                ;RESTORE R4
 3439  021022  012603                         MOV     (SP)+,R3                ;RESTORE R3
 3440  021024  012602                         MOV     (SP)+,R2                ;RESTORE R2
 3441  021026  000207                         RTS     PC                      ;RETURN
```

```
 3442
 3443  021030  .SBTTL  COMMAND LINE PARSING TREE FOR REPORT
 3444  021034  CLIRT:  CLI CLISPA,0,R10$                ;SKIP SPACES IN COMMAND LINE
 3445  021040  R10$:   CLI <'?>,RPHLP,R11$             ;IF INPUT = ? THEN PRINT HELP MESSAGE
                       CLI CLIEXI,0                    ;;AND EXIT PARSER
 3446  021042  R11$:   CLI CLISTR,RPHLP,R12$,<'HELP'>  ;IF INPUT = 'HELP' THEN PRINT HELP
 3447  021056           CLI CLIEXI,0                    ;;MESSAGE AND EXIT PARSER
 3448  021060  R12$:   CLI CLISTR,RPEXT,R13$,<'EXIT'>  ;IF INPUT = 'EXIT' THEN SET KEYWORD =
 3449  021074           CLI CLIEXI,0                    ;;RPEXT AND EXIT PARSER
 3450  021076  R13$:   CLI CLISTR,RPLOG,R14$,<'LOG'>   ;IF INPUT = 'LOG' THEN GO PRINT EVENT
 3451  021110           CLI CLIEXI,0                    ;;LOG AND EXIT PARSER
 3452  021112  R14$:   CLI CLISTR,RNOTNF,R30$,<'COUNTERS'>;;IF INPUT = COUNTERS
 3453  021132           CLI CLIBR,0,R20$                ;;THEN GET SWITCH
 3454  021136  R20$:   CLI <'/>,RNOTNF,R30$            ;;
 3455  021142           CLI CLISTR,RPERR,R21$,<'ERROR'> ;; REPORT ERROR COUNTERS
 3456  021156           CLI CLIEXI,0
 3457  021160  R21$:   CLI CLISTR,RPFUL,R22$,<'FULL'>  ;;REPORT ALL STATUS
 3458  021174           CLI CLIEXI,0                    ;
 3459  021176  R22$:   CLI CLISTR,RNOTNF,R30$,<'OFFSET'>;; REPORT ONE LOCATION
 3460  021214           CLI <'=>,0,R30$
 3461  021220           CLI CLIOCT,RPSWO,R30$           ;;
 3462  021224           CLI CLIEXI,0
 3463  021226  R30$:   CLI CLIERR,0
 3464  021230  R125$:  CLI CLIEXI,0
```

```
                        .SBTTL  CLI ACTION DISPATCHER AND ROUTINES
021232  006302          CLIRAC: ASL     R2              ;SET UP INDEX
021234  016202  021250          MOV     10$(R2),R2      ;
021240  062702  021250          ADD     #10$,R2         ;
021244  004712                  JSR     PC,(R2)         ;GO DO ACTION
021246  000207                  RTS     PC              ;RETURN
021250  000026          10$:    .WORD   ACTRNL-10$      ;NULL
021252  000030                  .WORD   ACTRHL-10$      ;HELP ROUTINE
021254  000074                  .WORD   ACTREX-10$      ;EXIT ROUTINE
021256  000104                  .WORD   ACTRLG-10$      ;REPORT EVENT LOG ROUTINE
021260  000142                  .WORD   ACTERR-10$      ;ERROR COUNTERS ONLY
021262  000120                  .WORD   ACTFUL-10$      ;ALL COUNTERS
021264  000020                  .WORD   ACTRNF-10$      ;MORE COMMAND NEEDED
021266  000164                  .WORD   ACTRSO-10$      ;VALIDATE OFFSET

                        ;;;;;ACTION ROUTINES FOR REPORT;;;;;
021270  113737  177777  003146 ACTRNF: MOVB    -1,P$NNUF       ;SET 'MORE COMMAND'' NEEDED
021276  000207         ACTRNL: RTS     PC              ;NULL

                                ;PRINT HELP MESSAGE
021300  012702  003036  ACTRHL: MOV     #RHLPTB,R2      ;INDEX FOR HELP MESSAGES
021304                  1$:     PRINTF  #HLPF,(R2)+     ;PRINT IT
021304  012246                                                  MOV     (R2)+,-(SP)
021306  012746  012273                                          MOV     #HLPF,-(SP)
021312  012746  000002                                          MOV     #2,-(SP)
021316  010600                                                  MOV     SP,R0
021320  104417                                                  TRAP    C$PNTF
021322  062706  000006                                          ADD     #6,SP
021326  020227  003052                  CMP     R2,#RHLPEN      ;LAST MESSAGE ?
021332  001364                  BNE     1$              ;NO,BRANCH
021334  012737  000001  003010        MOV     #RPHLP,KEYWD1   ;SET KEYWORD
021342  000207                  RTS     PC              ;RETURN

                                ;EXIT REPORT LEVEL
021344  012737  000002  003010 ACTREX: MOV     #RPEXT,KEYWD1   ;SET KEYWORD AND RETURN
021352  000207                  RTS     PC

                                ;PRINT ERROR LOG
021354  004737  022056  ACTRLG: JSR     PC,REPLOG       ;GO PRINT EVENT LOG
021360  012737  000003  003010        MOV     #RPLOG,KEYWD1   ;SET KEYWORD
021366  000207                  RTS     PC              ;RETURN

                                ;;REPORT ALL MESSAGE AND ERROR COUNTERS
021370  012737  000000  037250 ACTFUL: MOV     #0,FIR          ;STARTING INDEX
021376  012737  000036  037246        MOV     #36,LAST        ;LAST INDEX
021404  004737  021540                JSR     PC,STAPRI       ;GO PRINT IT
021410  000207                  RETURN

                                ;;PRINT ONLY DDCMP ERROR COUNTERS
021412  012737  000014  037250 ACTERR: MOV     #14,FIR         ;FIRST ERROR
021420  012737  000036  037246        MOV     #36,LAST        ;LAST ERROR
021426  004737  021540                JSR     PC,STAPRI       ;GO PRINT IT
021432  000207                  RETURN


                                ;;VERIFY OFFSET VALUE
```

```
3521  021434  105037  003146                ACTRSO: CLRB    P$NNUF          ;CLEAR 'NOT ENOUGH FLAG'
3522  021440  032737  000001  003142              BIT     #BIT0,P$NUM     ;IS IT ODD ?
3523  021446  001020                            BNE     20$             ;YES,BRANCH
3524  021450  005737  003142                    TST     P$NUM           ;NEGATIVE # ?
3525  021454  100415                            BMI     20$             ;YES,BRANCH
3526  021456  023727  003142  000036            CMP     P$NUM,#36       ;INDEX LARGER THEN 36 ?
3527  021464  003011                            BGT     20$             ;YES,BRANCH
3528  021466  013737  003142  037250            MOV     P$NUM,FIR       ;STARTING INDEX
3529  021474  013737  003142  037246            MOV     P$NUM,LAST      ;LAST LOCATION
3530  021502  004737  021540                    JSR     PC,STAPRI       ;PRINT SINGLE LOCATION
3531  021506  000413                            BR      30$             ;EXIT
3532  021510                          20$:      PRINTS  #RPTIV          ;INVALID
3533  021510  012746  013376                                                    MOV     #RPTIV,-(SP)
3534  021514  012746  000001                                                    MOV     #1,-(SP)
3535  021520  010600                                                          MOV     SP,R0
3536  021522  104416                                                          TRAP    C$PNTS
3537  021524  062706  000004                                                    ADD     #4,SP
3538  021530  112737  177777  003147            MOVB    #-1,P$GDBD      ;SET BAD DATA FLAG
3539  021536  000207                  30$:      RETURN                  ;OFFSET OK - EXIT
3540
3541
3542                                  ;; PRINT ROUTINES
3543  021540  010146                  STAPRI: MOV     R1,-(SP)        ;SAVE R1
3544  021542  032737  000040  006574            BIT     #PROTOB,PARAM   ;"/PROTOCOL" SELECTED?
3545  021550  001011                            BNE     5$              ;YES,BRANCH
3546  021552                                    PRINTF  #RPTNV          ;"VALID ONLY WITH PROTOCOL SELECTED"
3547  021552  012746  013421                                                    MOV     #RPTNV,-(SP)
3548  021556  012746  000001                                                    MOV     #1,-(SP)
3549  021562  010600                                                          MOV     SP,R0
3550  021564  104417                                                          TRAP    C$PNTF
3551  021566  062706  000004                                                    ADD     #4,SP
3552  021572  000420                            BR      20$             ;EXIT
3553  021574  013701  037250          5$:       MOV     FIR,R1  ;FIRST INDEX
3554  021600  016137  037146  037252  10$:      MOV     STALST(R1),MES  ;MESSAGE ADDRESS
3555  021606  016137  037046  037254            MOV     PRSTAT(R1),MESDATA ;MESSAGE DATA
3556  021614  004771  037206                    JSR     PC,@STAIND(R1)  ;JUMP TO PROPER PRINT ROUTINE
3557  021620  062701  000002                    ADD     #2,R1           ;BUMP INDEX
3558  021624  020137  037246                    CMP     R1,LAST         ;ALL MESSAGES PRINTED
3559  021630  003001                            BGT     20$             ;YES,BRANCH
3560  021632  000762                            BR      10$             ;PRINT NEXT MESSAGE
3561  021634  012601                  20$:      MOV     (SP)+,R1        ;RESTORE R1
3562  021636  000207                            RETURN                  ;EXIT
3563
3564
3565                                  ;; PRINT WORD LOCATION
3566  021640                          PRIW:   PRINTS  MES,MESDATA     ;PRINT WORD LOCATION
3567  021640  013746  037254                                                    MOV     MESDATA,-(SP)
3568  021644  013746  037252                                                    MOV     MES,-(SP)
3569  021650  012746  000002                                                    MOV     #2,-(SP)
3570  021654  010600                                                          MOV     SP,R0
3571  021656  104416                                                          TRAP    C$PNTS
3572  021660  062706  000006                                                    ADD     #6,SP
3573  021664  000207                            RETURN
3574
3575
3576                                  ;; PRINT TWO BYTES OF DATA
```

```
                                        PRIBB:  PRINTS  MES,<B,MESDATA>,<B,MESDATA+1>
021666
021666  005046                                                          CLR     -(SP)
021670  153716  037255                                                  BISB    MESDATA+1,(SP)
021674  005046                                                          CLR     -(SP)
021676  153716  037254                                                  BISB    MESDATA,(SP)
021702  013746  037252                                                  MOV     MES,-(SP)
021706  012746  000003                                                  MOV     #3,-(SP)
021712  010600                                                          MOV     SP,R0
021714  104416                                                          TRAP    C$PNTS
021716  062706  000010                                                  ADD     #10,SP
021722  000207                  RETURN
                                ;; PRINT SPECIAL BYTE MASK
021724  005037  006536          PRIBS:  CLR     TEMP1
021730  005037  006540                  CLR     TEMP2
021734  005037  006542                  CLR     TEMP3
021740  132737  000001  037255          BITB    #BIT0,MESDATA+1 ;BIT 0 = 1 ?
021746  001402                          BEQ     10$             ;NO,BRANCH
021750  005237  006536                  INC     TEMP1           ;SET IT
021754  132737  000002  037255  10$:    BITB    #BIT1,MESDATA+1 ;BIT 1 =1 ?
021762  001402                          BEQ     20$             ;NO,BRANCH
021764  005237  006540                  INC     TEMP2           ;SET IT
021770  132737  000004  037255  20$:    BITB    #BIT2,MESDATA+1 ;BIT2 = 1 ?
021776  001402                          BEQ     30$             ;NO,BRANCH
022000  005237  006542                  INC     TEMP3           ;SET IT
022004                          30$:    PRINTS  MES,<B,MESDATA>,<B,TEMP1>,<B,TEMP2>,<B,TEMP3>
022004  005046                                                          CLR     -(SP)
022006  153716  006542                                                  BISB    TEMP3,(SP)
022012  005046                                                          CLR     -(SP)
022014  153716  006540                                                  BISB    TEMP2,(SP)
022020  005046                                                          CLR     -(SP)
022022  153716  006536                                                  BISB    TEMP1,(SP)
022026  005046                                                          CLR     -(SP)
022030  153716  037254                                                  BISB    MESDATA,(SP)
022034  013746  037252                                                  MOV     MES,-(SP)
022040  012746  000005                                                  MOV     #5,-(SP)
022044  010600                                                          MOV     SP,R0
022046  104416                                                          TRAP    C$PNTS
022050  062706  000014                                                  ADD     #14,SP
022054  000207                  RETURN
```

```
                                        .SBTTL          DUMP EVENT LOG

                                REPLOG: MOV     R2,-(SP)        ;SAVE R2,R3,R4 ON THE STACK
                                        MOV     R3,-(SP)
                                        MOV     R4,-(SP)

                                        MOV     EVTPTR,R2       ;MAKE R2 A POINTER TO EVENT TABLE
                                        CMP     EVTLOG,#-1      ;SEE IF EVENT TABLE IS EMPTY
                                        BNE     RPT0            ;BR IF NO
                                        PRINTS  #NULEVT         ;IF EMPTY TELL OPRERATOR.
                                                                        MOV     #NULEVT,-(SP)
                                                                        MOV     #1,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTS
                                                                        ADD     #4,SP
                                        JMP     ENDEVT          ;AND END

                                RPT:    SUB     #12,R2          ;NOW POINT BACK TO TOP OF ENTRY U
                                                                ;JUST PRINTED

                                        CMP     R2,#EVTLOG      ;POINTING TO TOP OF EVNT LOG QUEUE?
                                        BNE     RPT1            ; BR IF NO
                                        MOV     #EVTEND,R2      ;SET R2 TO POINT TO BOTTOM OF LOG
                                        CMP     -2(R2),#-1
                                        BNE     RPT0            ;IF END OF LOG IS NOT EMPTY
                                        JMP     ENDEVT          ;CONTINUE...ELSE EXIT

                                RPT1:   CMP     R2,EVTPTR       ;ARE WE BACK TO POINTER?
                                        BNE     RPT0            ;IF NOT CONTINUE
                                        JMP     ENDEVT          ;IF SO EXIT....

                                RPT0:   SUB     #12,R2          ;POINT R2 TO START OF ENTRY
                                RPTAA:  PRINTS  #EVTF0          ;PRINT EVENT ENTRY HEADER
                                                                        MOV     #EVTF0,-(SP)
                                                                        MOV     #1,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTS
                                                                        ADD     #4,SP
                                        MOVB    (R2)+,R3        ;PUT EVENT TYPE INTO R3
                                        MOVB    (R2)+,EVTTCK
                                        MOVB    (R2)+,EVTSEC    ;PUT EVENT TIME (TICKS,SECS,MINS IN TEMP LOC.S)
                                        MOVB    (R2)+,EVTMIN
                                        PRINTS  #EVTF1,EVTMIN,EVTSEC,EVTTCK,EVTLST(R3)  ;PRINT EVENT TIME AND DESCRIPT.
                                                                        MOV     EVTLST(R3),-(SP)
                                                                        MOV     EVTTCK,-(SP)
                                                                        MOV     EVTSEC,-(SP)
                                                                        MOV     EVTMIN,-(SP)
                                                                        MOV     #EVTF1,-(SP)
                                                                        MOV     #5,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTS
                                                                        ADD     #14,SP
```

```
3676  022272  000173  007656                  JMP     @RPTDSP(R3)     ;DISPATCH TO DECODING SECTION FOR SPECIFIC TYPE
3677
3678  022276  012237  007650          RPTTXQ: MOV     (R2)+,EVTADD    ;STORE MESSAGE ADDRESS FOR PRINTING
3679  022302  012237  007652                  MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT FOR PRINTING
3680  022306  012203                          MOV     (R2)+,R3        ;STORE MODEM STATUS FOR PRINTING
3681  022310                                  PRINTS  #EVTF2,EVTADD,EVTBCT    ;PRINT ADDR,BYTE CNT
3682  022310  013746  007652                                                          MOV     EVTBCT,-(SP)
3683  022314  013746  007650                                                          MOV     EVTADD,-(SP)
3684  022320  012746  015672                                                          MOV     #EVTF2,-(SP)
3685  022324  012746  000003                                                          MOV     #3,-(SP)
3686  022330  010600                                                                  MOV     SP,R0
3687  022332  104416                                                                  TRAP    C$PNTS
3688  022334  062706  000010                                                          ADD     #10,SP
3689  022340  004737  023014                  JSR     PC,RPTMSB       ;GO PRINT MODEM STATUS
3690  022344  000137  022124                  JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
3691
3692  022350  012237  007654          RPTDER: MOV     (R2)+,EVTTMP    ;GET ADDRESS OF DEVICE INFO MESSAGE
3693  022354  012237  007706                  MOV     (R2)+,DEV1      ;STORE DEVICE REG CONTENTS FOR PRINTING
3694  022360  012237  007710                  MOV     (R2)+,DEV2
3695  022364                                  PRINTS  #EVTF3,EVTTMP   ;PRINT DEVICE REG CONTENTS.
3696  022364  013746  007654                                                          MOV     EVTTMP,-(SP)
3697  022370  012746  015744                                                          MOV     #EVTF3,-(SP)
3698  022374  012746  000002                                                          MOV     #2,-(SP)
3699  022400  010600                                                                  MOV     SP,R0
3700  022402  104416                                                                  TRAP    C$PNTS
3701  022404  062706  000006                                                          ADD     #6,SP
3702  022410                                  PRINTS  #EVTF3C,DEV1,DEV2
3703  022410  013746  007710                                                          MOV     DEV2,-(SP)
3704  022414  013746  007706                                                          MOV     DEV1,-(SP)
3705  022420  012746  015756                                                          MOV     #EVTF3C,-(SP)
3706  022424  012746  000003                                                          MOV     #3,-(SP)
3707  022430  010600                                                                  MOV     SP,R0
3708  022432  104416                                                                  TRAP    C$PNTS
3709  022434  062706  000010                                                          ADD     #10,SP
3710  022440  000137  022124                  JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
3711
3712  022444  005037  007706          RPTDVI: CLR     DEV1
3713  022450  005037  007710                  CLR     DEV2            ;CLEAR UPPER BYTES OF DEV1 & DEV2 BEFORE USE
3714  022454  112237  007706                  MOVB    (R2)+,DEV1      ;STORE SETUP OPERATION PARAMETERS FOR PRINTING
3715  022460  112237  007710                  MOVB    (R2)+,DEV2
3716  022464  012237  007712                  MOV     (R2)+,DEV3
3717  022470  012237  007714                  MOV     (R2)+,DEV4
3718  022474  010246                          MOV     R2,-(SP)        ;SAVE R2 ON THE STACK
3719  022476  004737  023712                  JSR     PC,SHWOP        ;GO PRINT MODE, MAINT-LOOP TYPE, PARAMTERS.
3720  022502  012602                          MOV     (SP)+,R2        ;RESTORE R2
3721  022504  000137  022124                  JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
3722
3723
                      ;;REPORT END OF PASS OR ^C ABORT
3724  022510          RPTABO:
3725  022510  012237  007650          RPTEOP: MOV     (R2)+,EVTADD
3726  022514  012237  007652                  MOV     (R2)+,EVTBCT
3727  022520  012237  007654                  MOV     (R2)+,EVTTMP
3728  022524                                  PRINTS  #EVTF4B,EVTADD,EVTBCT,EVTTMP    ;PRINT ADDR,RXBYTES,CMPBYTES.
3729  022524  013746  007654                                                          MOV     EVTTMP,-(SP)
3730  022530  013746  007652                                                          MOV     EVTBCT,-(SP)
3731  022534  013746  007650                                                          MOV     EVTADD,-(SP)
```

```
3732  022540  012746  016215                                                                  MOV     #EVTF4B,-(SP)
3733  022544  012746  000004                                                                  MOV     #4,-(SP)
3734  022550  010600                                                                          MOV     SP,R0
3735  022552  104416                                                                          TRAP    C$PNTS
3736  022554  062706  000012                                                                  ADD     #12,SP
3737
3738  022560  000137  022124                  JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY
3739
3740
3741  022564  012237  007650          RPTDDE: MOV     (R2)+,EVTADD    ;STORE MESSAGE ADDRESS FOR PRINTING
3742  022570  012237  007652                  MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT FOR PRINTING
3743  022574  012237  007654                  MOV     (R2)+,EVTTMP    ;STORE TOTAL # OF CMP ERRORS
3744  022600                                  PRINTS  #EVTF4,EVTADD,EVTBCT,EVTTMP     ;PRINT ADDR, BYTE CNT, # CMP ERRS
3745  022600  013746  007654                                                                  MOV     EVTTMP,-(SP)
3746  022604  013746  007652                                                                  MOV     EVTBCT,-(SP)
3747  022610  013746  007650                                                                  MOV     EVTADD,-(SP)
3748  022614  012746  016015                                                                  MOV     #EVTF4,-(SP)
3749  022620  012746  000004                                                                  MOV     #4,-(SP)
3750  022624  010600                                                                          MOV     SP,R0
3751  022626  104416                                                                          TRAP    C$PNTS
3752  022630  062706  000012                                                                  ADD     #12,SP
3753  022634  000137  022124                  JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY
3754
3755  022640                          RPTDLE:
3756  022640  012237  007650          RPTDCK: MOV     (R2)+,EVTADD    ;STORE MSG ADDR FOR PRINT
3757  022644  012237  007652                  MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT
3758  022650  012237  007654                  MOV     (R2)+,EVTTMP    ;STORE BYTE COUNT COMP
3759  022654                                  PRINTS  #EVTF4A,EVTADD,EVTBCT,EVTTMP    ;PRINT ADDR,RXBYTES,CMPBYTES.
3760  022654  013746  007654                                                                  MOV     EVTTMP,-(SP)
3761  022660  013746  007652                                                                  MOV     EVTBCT,-(SP)
3762  022664  013746  007650                                                                  MOV     EVTADD,-(SP)
3763  022670  012746  016117                                                                  MOV     #EVTF4A,-(SP)
3764  022674  012746  000004                                                                  MOV     #4,-(SP)
3765  022700  010600                                                                          MOV     SP,R0
3766  022702  104416                                                                          TRAP    C$PNTS
3767  022704  062706  000012                                                                  ADD     #12,SP
3768
3769  022710  000137  022124                  JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY
3770
3771
3772
3773
3774
3775  022714  012237  007654          RPTMSC: MOV     (R2)+,EVTTMP
3776  022720                                  PRINTS  #EVTF3,EVTTMP   ;PRINT CHANGE TYPE
3777  022720  013746  007654                                                                  MOV     EVTTMP,-(SP)
3778  022724  012746  015744                                                                  MOV     #EVTF3,-(SP)
3779  022730  012746  000002                                                                  MOV     #2,-(SP)
3780  022734  010600                                                                          MOV     SP,R0
3781  022736  104416                                                                          TRAP    C$PNTS
3782  022740  062706  000006                                                                  ADD     #6,SP
3783  022744  012203                  MOV     (R2)+,R3        ;PUT OLD MODEM STATUS IN R3 FOR PRINTING
3784  022746  004737  023014          JSR     PC,RPTMSB       ;GO PRINT OLD MODEM STATUS
3785  022752                          PRINTS  #EVMOCG         ;GO PRINT "CHANGED TO:"
3786  022752  012746  016357                                                                  MOV     #EVMOCG,-(SP)
3787  022756  012746  000001                                                                  MOV     #1,-(SP)
```

```
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTS
                                                                                ADD     #4,SP
                        MOV     (R2)+,R3        ;PUT NEW MODEM STATUS IN R3 FOR PRINTING
                        JSR     PC,RPTMSB       ;GO PRINT NEW MODEM STATUS
RPTMSE:                 JMP     RPT             ;THEN GO GET NEXT EVENT

ENDEVT:                 MOV     (SP)+,R4        ;RESTORE R4,R3,R2
                        MOV     (SP)+,R3
                        MOV     (SP)+,R2
                        RTS     PC              ;RETURN TO CALLING ROUTINE

;REPORT MODEM STATUS SUBROUTINE
;       PART OF STATISICAL REPORTING (DUMPING EVENT LOG)

RPTMSB:                 PRINTS  #EVMOHD                 ;PRINT MODEM STATUS HEADER
                                                                                MOV     #EVMOHD,-(SP)
                                                                                MOV     #1,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTS
                                                                                ADD     #4,SP
                        MOV     #MOBITS,R4      ;MAKE R4 A POINTER TO MODEM SIG. BIT DEF. TABLE
                        MOV     #MOMSGS,R5      ;MAKE R5 A POINTER TO MODEM MSG. POSITION TABLE
6$:                     TST     (R4)            ;SEE IF BIT AVAIABLE FROM DEVICE
                        BNE     7$              ;BR IF THAT MODEM SIG. AVAIABLE
                        MOVB    #'X,@(R5)+      ;ELSE PUT "X" IN REPORT IF SIGNAL NOT AVAILABLE
                        TST     (R4)+           ;BUMP R4 TO POINT TO NEXT BIT DEFINITION
                        BR      9$              ;GO SEE IF CHECKED ALL MODEM SIGNALS
7$:                     BIT     (R4)+,R3        ;IF THERE, SEE IF THAT BIT IN DEVICE'S ENTRY=1
                        BEQ     8$              ;BR IF BIT (SIGNAL) VALUE =0
                        MOVB    #'1,@(R5)+      ;IF=1, PUT "1" IN REPORT MESSAGE
                        BR      9$              ;GO SEE IF ALL MODEM SIGNALS CHECKED
8$:                     MOVB    #'0,@(R5)+      ;IF BIT(SIGNAL)=0, PUT "0" IN REPORT MESSAGE
9$:                     CMP     R4,#MOBITE      ;SEE IF ALL BITS(SIGNALS) CHECKED
                        BLT     6$              ;LOOP UNTIL ALL SIGNALS(BITS) CHECKED
                        PRINTS  #EVMOST         ;THEN PRINT MODEM SIGNAL VALUE MESSAGE
                                                                                MOV     #EVMOST,-(SP)
                                                                                MOV     #1,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTS
                                                                                ADD     #4,SP
                        RTS     PC              ;RETURN TO EVENT DECODING
```

```
                        .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
                        ;
                        ;--

023126 013702 006510    DUMPSR: MOV     STADD,R2        ;SET R2 UP TO STARTING ADDR.
023132 005003           DUM4:   CLR     R3              ;CLEAR R3
023134                          PRINTF  #BASM1,R2       ;PRINT ADDRESS
023134 010246                                                   MOV     R2,-(SP)
023136 012746 015501                                            MOV     #BASM1,-(SP)
023142 012746 000002                                            MOV     #2,-(SP)
023146 010600                                                   MOV     SP,R0
023150 104417                                                   TRAP    C$PNTF
023152 062706 000006                                            ADD     #6,SP
023156 005737 006514    DUM3:   TST     BYTBIT          ;IS THIS BYTE OR WORD
023162 001416                   BEQ     DUM1            ;BR IF WORD
023164 112237 006534            MOVB    (R2)+,TEMP      ;MOV BYTE TO TEMP
023170                          PRINTF  #BASM3,<B,TEMP> ;PRINT BYTE
023170 005046                                                   CLR     -(SP)
023172 153716 006534                                            BISB    TEMP,(SP)
023176 012746 015463                                            MOV     #BASM3,-(SP)
023202 012746 000002                                            MOV     #2,-(SP)
023206 010600                                                   MOV     SP,R0
023210 104417                                                   TRAP    C$PNTF
023212 062706 000006                                            ADD     #6,SP
023216 000411                   BR      DUM2
023220          DUM1:   PRINTF  #BASM2,(R2)+    ;PRINT WORD
023220 012246                                                   MOV     (R2)+,-(SP)
023222 012746 015472                                            MOV     #BASM2,-(SP)
023226 012746 000002                                            MOV     #2,-(SP)
023232 010600                                                   MOV     SP,R0
023234 104417                                                   TRAP    C$PNTF
023236 062706 000006                                            ADD     #6,SP
023242 020237 006512    DUM2:   CMP     R2,ENADD        ;COMPARE FOR LAST ADD
```

```
 3890  023246  003005                  BGT     DUMEX           ;IF DONE EXIT
 3891  023250  005203                  INC     R3              ;ELSE BUMP R3
 3892  023252  022703  000010          CMP     #8.,R3          ;HAVE WE PRINTED 8 ACCROSS
 3893  023256  001725                  BEQ     DUM4            ;IF SO GO BACK TO 4
 3894  023260  000736                  BR      DUM3            ;ELSE GO BACK AND PRINT ANOTHER
 3895                                                          ;BYTE OR WORD
 3896  023262  000207          DUMEX:  RTS     PC              ;RETURN TO CALLER
 3897
```

```
                        .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
                        ;--

023264  063737  006520  006530  ADDCC:  ADD     CURCC,TOTCC     ;ADD CURRENT TO TOTAL
023272  022737  001000  006530           CMP     #BUFLIM,TOTCC   ; COMPARE TO 'BUFLIM'
023300  103027                         BHIS    ADDC1           ;IF NOT MORE THEN 'BUFLIM' EXIT

                        ; PRINT MESSAGE AND TRUNCATE COUNT

023302                  PRINTF  #MSGTRU
023302  012746  014416                                                  MOV     #MSGTRU,-(SP)
023306  012746  000001                                                  MOV     #1,-(SP)
023312  010600                                                          MOV     SP,R0
023314  104417                                                          TRAP    C$PNTF
023316  062706  000004                                                  ADD     #4,SP
023322  163737  006520  006530           SUB     CURCC,TOTCC     ;SUB CURRENT FROM TOTAL
023330  012737  001000  006534           MOV     #BUFLIM,TEMP    ;MOV 'BUFLIM' TO TEMP
023336  163737  006530  006534           SUB     TOTCC,TEMP      ;SUB TOTAL FROM 'BUFLIM'
023344  013737  006534  006520           MOV     TEMP,CURCC      ;AND ESTABLISH NEW CURRENT
023352  063737  006520  006530           ADD     CURCC,TOTCC     ;ADD "ADJUSTED CURRENT" TO TOTAL CHAR. CNT.
023360  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.
                        ;--
                        ;
023362                  BLDBUF:
023362  010246                  MOV     R2,-(SP)        ;SAVE R2 AND R3 ON THE STACK
023364  010346                  MOV     R3,-(SP)
023366  013702  006524          MOV     CPTR,R2
                        ;
023372  013722  006526  BLDB1:  MOV     CURADD,(R2)+    ;PUT CURRENT ADD ON POINTER TAB
023376  013722  006520          MOV     CURCC,(R2)+     ;PUT CURRENT CC ON POINTER TAB
023402  010237  006524          MOV     R2,CPTR         ;PUT UPDATED R2 BACK TO CURRENT POINT
023406  013702  006516          MOV     MSGTYP,R2       ;GET MESSAGE TYPE TO USE AS INDEX
023412  006302                  ASL     R2              ;DOUBLE FOR WORD INDEX
023414  013737  006526  006534  MOV     CURADD,TEMP     ;MOVE CURRENT ADD TO TEMP
023422  063737  006520  006534  ADD     CURCC,TEMP      ;ADD CHAR COUNT TO IT TO GET END
023430  013703  006526          MOV     CURADD,R3       ;SET R3 TO CURRENT START ADD
023434  016237  002150  006540  BLDB2:  MOV     DMSGCT(R2),TEMP2        ;GET BYTE COUNT
023442  016204  002172          MOV     DMSGAD(R2),R4   ;PUT STARTING FROM ADD IN R4
023446  060437  006540          ADD     R4,TEMP2        ;ADD IT TO TEMP2 TO GET END OF FROM
023452  112423          BLDB3:  MOVB    (R4)+,(R3)+     ;MOV BYTE FROM PATTERN TO BUFFER
023454  020337  006534          CMP     R3,TEMP         ;ALL DONE?
023460  001404                  BEQ     BLDBEX          ;IF SO EXIT
023462  020437  006540          CMP     R4,TEMP2        ;IS PATTERN COUNT EXPIRED
023466  001762                  BEQ     BLDB2           ;IF SO GO START AGAIN
023470  000770                  BR      BLDB3           ;IF NOT GET ANOTHER BYTE
023472  063737  006520  006526  BLDBEX: ADD     CURCC,CURADD    ;BUMP CURADD
023500  012603                  MOV     (SP)+,R3        ;RESTORE R3 AND R2
023502  012602                  MOV     (SP)+,R2
023504  000207                  RTS     PC              ;RETURN TO CALLER
```

```
3991                    .SBTTL  CREATE FACSIMILE OF TX BUFFER AND MESSAGE LIST
3992
3993                    ;++
3994                    ;       THIS ROUTINE ADDED FOR REV B BY EC
3995                    ;FUNCTIONAL DESCRIPTION:
3996                    ;       FACSIMILE: THIS ROUTINE IS USED TO CREATE A FACSIMILE OF THE
3997                    ;                  OF THE TRANSMIT LIST AND TRANSMIT BUFFER IN THE
3998                    ;                  EXPECTED LIST AND EXPECTED BUFFER. THE ROUTINE IS
3999                    ;                  NORMALLY CALLED WHEN USER COMMAND ''SET E [XPECT]=
4000                    ;                  T [RANSMIT] IS ENTERED.
4001                    ;
4002                    ;       CALLING SEQUENCE: JSR   PC,FACSIMILE
4003                    ;
4004                    ;--
4005                    ;       DEFINITIONS  CMPBUF = EXPECTED DATA BUFFER  HOLDS MAX 512 BYTES
4006                    ;                    TXBUF  = TRANSMIT DATA BUFFER  HOLDS MAX 512 BYTES
4007                    ;                    TTOTCC = NUMBER OF BYTES IN TXBUF
4008                    ;                    PTRTAB = TOP OF MESSAGE LIST POINTER TABLE
4009                    ;                    CTOTCC = NUMBER OF BYTES IN EXPECT MESSAGE
4010                    ;                    CMPTOT = NUMBER OF EXPECTED MESSAGES
4011                    ;                    CMPPTR = EXPECTED MESSAGE LIST POINTER
4012                    ;                    TXPTR  = TRANSMIT MESSAGE LIST POINTER
4013                    ;                    TXMTOT = NUMBER OF TRANSMIT MESSAGES
4014                    ;                    CCURAD = STORAGE ADDRESS OF MESSAGE IN CMPBUF
4015                    ;                    MSGLIN = MAXIMUM NUMBER OF MESSAGES THAT CAN BE STORED
4016                    ;
4017                    ;       BEGIN FACSIMILE ROUTINE
4018                    ;       (*COPY TXBUF ==> CMPBUF*)
4019                    ;       ..SAVE R1
4020                    ;       ..INIT R1
4021                    ;       ..REPEAT
4022                    ;       ....[CMPBUF]R1=[TXBUF]R1
4023                    ;       ....R1=R1+1
4024                    ;       ..UNTIL R1 = BUFLIM
4025                    ;
4026                    ;       (*NOW CALCULATE EXPECT LIST MESSAGE POINTER*)
4027                    ;       ..CMPPTR = PTRTAB + (2 * MSGLIM)
4028                    ;
4029                    ;       (*NOW PRIME THE WHILE - DO LOOP*)
4030                    ;       ..TXPTR = PTRTAB
4031                    ;       ..CCURAD = CMPBUF
4032                    ;       ..TXPTR = TXPTR + 2
4033                    ;       ..CTOTCC = [TXPTR]
4034                    ;       ..CMPTOT = 0
4035                    ;       ..WHILE TXMTOT <> CMPTOT DO
4036                    ;       ....[CMPPTR] = CCURAD
4037                    ;       ....CMPPTR = CMPPRT + 2
4038                    ;       ....[CMPPTR] = CTOTCC
4039                    ;       ....TXPTR = TXPTR + 4
4040                    ;       ....CCURAD = CCURAD + CTOTCC
4041                    ;       ....CTOTCC = [TXPTR]
4042                    ;       ....CMPPTR = CMPPTR + 2
4043                    ;       ....CMPTOT = CMPTOT + 1
4044                    ;       ..END WHILE DO
4045                    ;       ..CTOTCC = TTOTCC
4046                    ;       END FACSIMILE ROUTINE
```

```
                                        FACSIMILE:

023506  010146                                  MOV     R1,-(SP)                ;SAVE R1
023510  005001                                  CLR     R1                      ;INIT R1
023512  116161  003150  005150          10$:    MOVB    TXBUF(R1),CMPBUF(R1)    ;COPY TX BUFFER TO EXPECTED BUFFER
023520  005201                                  INC     R1                      ;BUMP INDEX
023522  020127  001000                          CMP     R1,#BUFLIM              ;ALL DATA COPIED ?
023526  001371                                  BNE     10$                     ;NO,BRANCH

023530  012701  000017                  20$:    MOV     #MSGLIM,R1              ;MESSAGE LIMIT
023534  006301                                  ASL     R1                      ;MULTIPLY BY 2
023536  006301                                  ASL     R1                      ;MULTIPLY BY 2
023540  012737  006150  006444                  MOV     #PTRTAB,CMPPTR          ;TOP OF POINTER TABLE
023546  060137  006444                          ADD     R1,CMPPTR               ;START OF EXPECTED POINTER TABLE
023552  005001                                  CLR     R1                      ;INIT R1

                                                ;SET UP WHILE - DO LOOP
023554  012737  006150  006442                  MOV     #PTRTAB,TXPTR           ;TX POINTER NOW AT TOP OF TABLE
023562  012737  005150  006452                  MOV     #CMPBUF,CCURAD          ;TRANSFER ADDRESS OF 1ST MESSAGE
023570  062737  000002  006442                  ADD     #2,TXPTR                ;BUMP POINTER
023576  017737  162640  006450                  MOV     @TXPTR,CTOTCC           ;BYTE COUNTER 1ST MESSAGE
023604  005037  006446                          CLR     CMPTOT                  ;INIT EXPECTED MESSAGE COUNT

                                                ;WHILE TX MESSAGE TOTAL <> EXPECTED MESSAGE TOTAL DO
023610  023737  006462  006446  30$:    CMP     TXMTOT,CMPTOT                   ;ALL MESSAGES COPIED ?
023616  001430                          BEQ     40$                             ;YES,BRANCH
023620  013777  006452  162616          MOV     CCURAD,@CMPPTR                  ;TRANSFER ADDRESS OF MESSAGE
023626  062737  000002  006444          ADD     #2,CMPPTR                       ;BUMP POINTER
023634  013777  006450  162602          MOV     CTOTCC,@CMPPTR                  ;BYTE COUNT OF MESSAGE
023642  062737  000004  006442          ADD     #4,TXPTR                        ;BUMP TX MESSAGE POINTER
023650  063737  006450  006452          ADD     CTOTCC,CCURAD                   ;CALC. TRANSFER ADDRESS
023656  017737  162560  006450          MOV     @TXPTR,CTOTCC                   ;BYTE COUNT NEXT MESSAGE
023664  062737  000002  006444          ADD     #2,CMPPTR                       ;BUMP POINTER
023672  005237  006446                  INC     CMPTOT                          ;INCREMENT MESSAGE COUNT
023676  000744                          BR      30$                             ;DO IT AGAIN
                                        ;END WHILE - DO
023700  013737  006464  006450  40$:    MOV     TTOTCC,CTOTCC                   ;COPY TOTAL CHARACTER COUNT

                                        ;END ROUTINE
023706  012601                          MOV     (SP)+,R1                        ;RESTORE R1
023710  000207                          RTS     PC                              ;RETURN
```

```
                                .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
                                ;--

023712  013702  007706        SHWOP:  MOV     DEV1,R2         ;GET THE MODE TYPE IN R2
023716  006302                        ASL     R2              ;MAKE IT A WORD TABLE OFFSET
023720  016237  003102  006534        MOV     MODES(R2),TEMP  ;GET ADDRESS OF MODE-IN-ASCII
023726  013702  007710                MOV     DEV2,R2         ;GET MAINTENANCE LOOP TYPE
023732  006302                        ASL     R2
023734  012737  013715  006542        MOV     #LP00,TEMP3     ;LOAD TEMP3 TO POINT TO ''/LOOP=''
023742  005702                        TST     R2              ;SEE IF /LOOP=XXXXX OR NONE
023744  001003                        BNE     10$             ; BR IF /LOOP= OF SOME KIND
023746  012737  013714  006542        MOV     #LP0,TEMP3      ;IF NO LOOP THEN DON'T PRINT ''/LOOP=''
023754  016237  003120  006536  10$:  MOV     LOOPS(R2),TEMP1 ;GET ADDRESS OF LOOP-IN-ASCII
023762  013737  007712  006540        MOV     DEV3,TEMP2      ;GET NUMBER OF PASSES
023770                                PRINTS  #SHF0,TEMP,TEMP3,TEMP1,TEMP2
023770  013746  006540                                                        MOV     TEMP2,-(SP)
023774  013746  006536                                                        MOV     TEMP1,-(SP)
024000  013746  006542                                                        MOV     TEMP3,-(SP)
024004  013746  006534                                                        MOV     TEMP,-(SP)
024010  012746  014501                                                        MOV     #SHF0,-(SP)
024014  012746  000005                                                        MOV     #5,-(SP)
024020  010600                                                                MOV     SP,R0
024022  104416                                                                TRAP    C$PNTS
024024  062706  000014                                                        ADD     #14,SP

024030  005002                        CLR     R2              ;NOW SET UP FOR QUALIFIERS IN ASCII
024032  012737  013774  006534        MOV     #PST,TEMP
024040  032737  000001  007714        BIT     #STATB,DEV4     ;SEE IF /STATUS OR /NOSTATUS
024046  001003                        BNE     1$              ;BR IF /STATUS
024050  012737  013772  006534        MOV     #PNST,TEMP
024056  012737  014005  006536  1$:   MOV     #PCK,TEMP1
024064  032737  000002  007714        BIT     #DATCKB,DEV4    ;SEE IF /CHECK OR /NOCHECK
024072  001003                        BNE     2$              ;BR IF /CHECK
024074  012737  014003  006536        MOV     #PNCK,TEMP1
024102  012737  014015  006540  2$:   MOV     #PEC,TEMP2
024110  032737  000004  007714        BIT     #ECHOB,DEV4     ;SEE IF /ECHO OR /NOECHO
024116  001003                        BNE     4$              ;BR IF /ECHO
024120  012737  014013  006540        MOV     #PNEC,TEMP2
```

```
4147
4148  024126  012737  014034  006544  4$:     MOV     #PPR,TEMP4
4149  024134  032737  000040  007714          BIT     #PROTOB,DEV4    ;SEE OF /PROTOCOL OR /NOPROTOCOL
4150  024142  001003                          BNE     3$              ;BR IF /PROTOCOL
4151  024144  012737  014032  006544          MOV     #PNPR,TEMP4
4152  024152  012737  014024  006546  3$:     MOV     #PMS,TEMP5
4153  024160  032737  000010  007714          BIT     #MOCHK,DEV4     ;SEE IF /MODEM OR /NOMODEM
4154  024166  001003                          BNE     5$              ;BR IF MODEM
4155  024170  012737  014022  006546          MOV     #PNMS,TEMP5
4156
4157
4158  024176                          5$:     PRINTS  #SHF1,TEMP,TEMP1,TEMP2,TEMP5,TEMP4      ;,TEMP3
4159  024176  013746  006544                                                                  MOV     TEMP4,-(SP)
4160  024202  013746  006546                                                                  MOV     TEMP5,-(SP)
4161  024206  013746  006540                                                                  MOV     TEMP2,-(SP)
4162  024212  013746  006536                                                                  MOV     TEMP1,-(SP)
4163  024216  013746  006534                                                                  MOV     TEMP,-(SP)
4164  024222  012746  014537                                                                  MOV     #SHF1,-(SP)
4165  024226  012746  000006                                                                  MOV     #6,-(SP)
4166  024232  010600                                                                          MOV     SP,R0
4167  024234  104416                                                                          TRAP    C$PNTS
4168  024236  062706  000016                                                                  ADD     #16,SP
4169  024242  000207                          RTS     PC              ;RETURN
4170
4171
```

```
                        .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
                        ;--

024244                  P$TRV:
024244  013704  003132          MOV     P$BUFA,R4
024250  013703  003134          MOV     P$TREE,R3
024254  105714          P$TR5:  TSTB    (R4)            ;SEE IF ANY CHARS LEFT IN INPUT STRING
024256  001441                  BEQ     P$EXIT          ;BR IF NO
024260  121327  000013          CMPB    (R3),#11.       ;SEE IF SPECIAL CLI CHAR CODE OR ASCII
024264  003023                  BGT     20$             ;BR IF REGULAR ASCII CHAR.
024266  111305                  MOVB    (R3),R5         ;GET SPECIAL CHAR CODE INTO R5
024270  006305                  ASL     R5
024272  016505  024306          MOV     10$(R5),R5      ;BUILD TRAVERSE ROUTINE ADDRESS
024276  062705  024306          ADD     #10$,R5
024302  004715                  JSR     PC,(R5)         ;JSR TO SPECIAL CLI TRAVERSE ROUTINE
024304  000763                  BR      P$TR5           ;GO SEE IF MORE OF STRING LEFT

024306  000114          10$:    .WORD   TRVERR-10$      ;TRAVERSE TABLE FOR "CLI FUNTIONS"
024310  000134                  .WORD   TRVEXI-10$      ;1
024312  000152                  .WORD   TRVBR-10$       ;2
024314  000162                  .WORD   TRVBIF-10$      ;3
024316  000204                  .WORD   TRVSPA-10$      ;4
024320  000270                  .WORD   TRVNUM-10$      ;5
024322  000604                  .WORD   TRVALP-10$      ;6
024324  000650                  .WORD   TRVALN-10$      ;7
024326  000270                  .WORD   TRVOCT-10$      ;8
024330  000256                  .WORD   TRVDEC-10$      ;9
024332  000736                  .WORD   TRVSTR-10$      ;10

                        ;NOT A SPECIAL CODE

024334  121314          20$:    CMPB    (R3),(R4)       ;SEE IF FIRST CHAR OF STRING IS A MATCH
024336  001403                  BEQ     22$             ;BR IF A MATCH
024340  004737  024404          JSR     PC,TRVBRC       ;IF NOT A MATCH, GO TAKE MISS BRANCH
024344  000743                  BR      P$TR5           ; THEN GO BACK PT'G TO MISS NODE
024346  004737  024364  22$:    JSR     PC,TRVACT       ;IF A MATCH, GO DO ACTION DEFINED BY
024352  062703  000004          ADD     #4,R3           ; ACTION CODE IN CLI NODE, THEN
                                                        ; ADJUST PTR TO NEXT CLI NODE
024356  005204                  INC     R4              ;ADJUST BUF PTR TO NEXT CHAR IF MATCH
024360  000735                  BR      P$TR5
```

```
4228
4229  024362  000207                  P$EXIT: RTS     PC                      ;RETURN FROM PARSER
4230
4231                                  ;-----------------------------------------------------------------------
4232
4233                                  ;GOTO USER ACTION ROUTINE
4234  024364  116302  000001          TRVACT: MOVB    1(R3),R2                ;GET ACTION CODE FROM CLI NODE
4235  024370  042702  177400                  BIC     #177400,R2              ;CLEAR ANY SIGN EXTENSION
4236  024374  013705  003136                  MOV     P$ACT,R5                ;GET ADDRESS OF CLI ACTION ROUTINE
4237  024400  004715                          JSR     PC,(R5)                 ;GO DO ACTION DEFINED BY CODE
4238  024402  000207                          RTS     PC                      ;RETURN TO CALLING CODE
4239
4240                                  ;TAKE BRANCH IN TREE
4241  024404  016305  000002          TRVBRC: MOV     2(R3),R5                ;GET BRANCH DISPLACEMENT FROM TREE
4242  024410  060503                          ADD     R5,R3                   ; AND POINT R3 TO THE 'MISS'' NODE
4243  024412  000207                          RTS     PC                      ; RETURN TO P$TRV
4244
4245                                  ;NO BRANCH TAKEN
4246  024414  062703  000004          TRVNOB: ADD     #4,R3                   ;THINGS OK, UPDATE R3 TO POINT TO NEXT
4247  024420  000207                          RTS     PC                      ; NODE AND RETURN TO P$TRV
4248
4249                                  ;-----------------------------------------------------------------------
4250  024422  004737  024364          TRVERR: JSR     PC,TRVACT               ;TAKE ERROR ACTION
4251  024426  112737  177777  003147          MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
4252  024434  005726                          TST     (SP)+                   ;GET RID OF ''JSR PUSH TO TRVERR''
4253  024436  000137  024362                  JMP     P$EXIT                  ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
4254
4255  024442  004737  024364          TRVEXI: JSR     PC,TRVACT               ;TAKE EXIT ACTION
4256  024446  105037  003147                  CLRB    P$GD3D                  ;SET GOOD/BAD FLAG TO ''SUCCESS (0)''
4257  024452  005726                          TST     (SP)+                   ;GET RID OF ''JSR PUSH TO TRVEXI''
4258  024454  000137  024362                  JMP     P$EXIT                  ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
4259
4260  024460  004737  024364          TRVBR:  JSR     PC,TRVACT               ;GO TAKE BRANCH ACTION
4261  024464  000137  024404                  JMP     TRVBRC
4262
4263  024470  004737  024364          TRVBIF: JSR     PC,TRVACT
4264  024474  105737  003147                  TSTB    P$GDBD                  ;SEE IF P$GDBD SET OR CLEARED BY ACTION
4265  024500  001402                          BEQ     1$                      ;IF CLEAR FALL THRU TO NEXT NODE
4266  024502  000137  024404                  JMP     TRVBRC                  ;ELSE TAKE THE 'MISS'' BRANCH
4267  024506  000137  024414          1$:     JMP     TRVNOB                  ;JUST UPDATE TO NEXT NODE IF THINGS OK
4268
4269  024512  005005                  TRVSPA: CLR     R5                      ;CLEAR ''SPACE OR TAB FOUND'' FLAG
4270  024514  121427  000011          1$:     CMPB    (R4),#11                ;SEE IF CHAR. IN CMD LINE= TAB
4271  024520  001003                          BNE     2$                      ;BR IF NO, NOT A TAB
4272  024522  005204                          INC     R4                      ;INC INPUT STRING POINTER
4273  024524  005205                          INC     R5                      ;INDICATE A TAB FOUND
4274  024526  000772                          BR      1$                      ;GO CHECK NEXT CHAR
4275
4276  024530  121427  000040          2$:     CMPB    (R4),#40                ;SEE IF CHAR. IN CMD LINE= SPACE
4277  024534  001003                          BNE     10$                     ;BR IF NO, NON-SPACE OR NON-TAB CHAR.
4278  024536  005204                          INC     R4                      ;INC INPUT STRING POINTER
4279  024540  005205                          INC     R5                      ;INDICATE A SPACE FOUND
4280  024542  000764                          BR      1$                      ;GO CHECK NEXT CHAR
4281  024544  005705                  10$:    TST     R5                      ;SEE IF ANY SPACES OR TABS FOUND
4282  024546  001404                          BEQ     15$                     ;BR IF NO, TAKE NO ACTION
4283  024550  004737  024364                  JSR     PC,TRVACT               ;GO TAKE ACTION IF ANY FOUND
```

```
4284  024554  000137  024414                  JMP     TRVNOB                  ;JUST GO UPDATE R3 TO NEXT NODE IF OK
4285  024560  000137  024404          15$:    JMP     TRVBRC                  ;TAKE BRANCH (MISS) IF NONE FOUND
4286
4287
4288  024564  012737  000012  003144  TRVDEC: MOV     #10.,P$RADX             ;USE DECIMAL AS RADIX AND ASSUME +
4289  024572  000137  024604                  JMP     TRVNMA
4290  024576                          TRVOCT: ;(SAME AS TRVNUM SINCE DEFAULT RADIX IS OCTAL)
4291  024576  012737  000010  003144  TRVNUM: MOV     #8.,P$RADX              ;USE OCTAL AS RADIX AND ASSUME +
4292  024604  005005                  TRVNMA: CLR     R5                      ;CLEAR DIGIT COUNTER
4293  024606  121427  000053                  CMPB    (R4),#'+                ;SEE IF THERE'S A + SIGN THERE
4294  024612  001001                          BNE     10$                     ; BR IF NO
4295  024614  000406                          BR      11$                     ; ELSE P$RADX ALREADY SAYS +, JUST BR
4296  024616  121427  000055          10$:    CMPB    (R4),#'-                ;SEE IF THERE'S A - SIGN THERE
4297  024622  001004                          BNE     1$                      ; BR IF NO
4298  024624  112737  177777  003145          MOVB    #-1,P$RADX+1            ;SET "MINUS FLAG" (HI BYTE OF P$RADX)
4299  024632  005204                  11$:    INC     R4                      ;BUMP R4 TO POINT TO FIRST CHAR
4300
4301  024634  121427  000060          1$:     CMPB    (R4),#60                ;SEE IF CHAR. LESS THAN A "0"
4302  024640  002434                          BLT     2$                      ;BR IF YES (NOT NUMERIC)
4303  024642  121427  000067                  CMPB    (R4),#67                ;SEE IF CHAR. GREATER THAN A "7"
4304  024646  003426                          BLE     13$                     ; BR IF YES
4305  024650  123727  003144  000012          CMPB    P$RADX,#10.             ;SEE IF IN DECIMAL MODE
4306  024656  001417                          BEQ     12$                     ; BR IF YES (CAN USE HIGHER LIMIT)
4307  024660  121427  000071                  CMPB    (R4),#71                ;SEE IF DIGIT WAS A 8 OR 9
4308  024664  003022                          BGT     2$                      ;BR IF NON-NUMERIC
4309  024666                                  PRINTF  #CLIBRX                 ;ELSE WAS A 8 OR 9 WHEN IN OCTAL RADIX
4310  024666  012746  011702                                                          MOV     #CLIBRX,-(SP)
4311  024672  012746  000001                                                          MOV     #1,-(SP)
4312  024676  010600                                                                  MOV     SP,R0
4313  024700  104417                                                                  TRAP    C$PNTF
4314  024702  062706  000004                                                          ADD     #4,SP
4315  024706  112737  177777  003147          MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
4316  024714  000474                          BR      5$                      ; PRINT ERROR AND TAKE MISS
4317
4318  024716  121427  000071          12$:    CMPB    (R4),#71                ;SEE IF CHAR. GREATER THAN A "9"
4319  024722  003003                          BGT     2$                      ;BR IF YES (NOT NUMERIC)
4320  024724  005204                  13$:    INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR.
4321  024726  005205                          INC     R5                      ;INDICATE A NUMERIC FOUND
4322  024730  000741                          BR      1$                      ;GO LOOK AT NEXT CHAR.
4323
4324  024732  005705                  2$:     TST     R5                      ;SEE IF FOUND ANY NUMERICS
4325  024734  001464                          BEQ     5$                      ;BR IF NO, TAKE "MISS" BRANCH
4326  024736  010401                          MOV     R4,R1                   ;GET POINTER TO START OF NUMERIC STRING
4327  024740  160501                          SUB     R5,R1
4328  024742  005037  003142                  CLR     P$NUM                   ;CLEAR LOC. WHERE VALUE WILL BE STORED
4329  024746  112102                  3$:     MOVB    (R1)+,R2                ;GET ASCII CHAR AND CONVERT IT TO A #
4330  024750  162702  000060                  SUB     #60,R2
4331  024754  006337  003142                  ASL     P$NUM                   ;SHIFT CURRENT VALUE TO MAKE ROOM
4332  024760  103437                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
4333  024762  013737  003142  003140          MOV     P$NUM,P$CNT             ;SAVE FOR LATER IN CASE DECIMAL RADIX
4334  024770  006337  003142                  ASL     P$NUM
4335  024774  103431                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
4336  024776  006337  003142                  ASL     P$NUM
4337  025002  103426                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
4338  025004  123727  003144  000012          CMPB    P$RADX,#10.             ;SEE IF DECIMAL RADIX
4339  025012  001004                          BNE     4$                      ;BR IF NOT EQUAL
```

```
4340  025014  063737  003140  003142          ADD     P$CNT,P$NUM
4341  025022  103416                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
4342  025024  060237  003142          4$:     ADD     R2,P$NUM
4343  025030  103413                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
4344  025032  005305                          DEC     R5
4345  025034  001344                          BNE     3$
4346  025036  105737  003145                  TSTB    P$RADX+1                ;SEE IF NUM WAS PRECEDED BY A - SIGN
4347  025042  001402                          BEQ     15$                     ; BR IF NO
4348  025044  005437  003142                  NEG     P$NUM                   ; ELSE NEGATE THE NUMBER BEFORE LEAVING
4349  025050  004737  024364          15$:    JSR     PC,TRVACT               ;SINCE NUMERIC FOUND, GO TAKE ACTION
4350  025054  000137  024414                  JMP     TRVNOB                  ;GO POINT R3 TO NEXT NODE
4351
4352  025060                          7$:     PRINTF  #CLINBG                 ;PRINT NUMBER TOO BIG ERROR
4353  025060  012746  011660                                                          MOV     #CLINBG,-(SP)
4354  025064  012746  000001                                                          MOV     #1,-(SP)
4355  025070  010600                                                                  MOV     SP,R0
4356  025072  104417                                                                  TRAP    C$PNTF
4357  025074  062706  000004                                                          ADD     #4,SP
4358  025100  112737  177777  003147          MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
4359  025106  000137  024404          5$:     JMP     TRVBRC                  ;TAKE "MISS" BRANCH
4360
4361
4362  025112  005005                  TRVALP: CLR     R5                      ;CLEAR ALPHA FOUND FLAG
4363  025114  121427  000101          1$:     CMPB    (R4),#101               ;SEE IF CHAR. LESS THAN A "A"
4364  025120  002406                          BLT     2$                      ;BR IF YES (NOT ALPHA)
4365  025122  121427  000132                  CMPB    (R4),#132               ;SEE IF CHAR. GREATER THAN A "Z"
4366  025126  003003                          BGT     2$                      ;BR IF YES (NOT ALPHA)
4367  025130  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR
4368  025132  005205                          INC     R5                      ;INDICATE AN ALPHA WAS FOUND
4369  025134  000767                          BR      1$                      ;GO LOOK AT NEXT CHAR.
4370  025136  005705                  2$:     TST     R5                      ;SEE IF ANY ALPHA'S WERE FOUND
4371  025140  001404                          BEQ     3$                      ;BR IF NO
4372  025142  004737  024364                  JSR     PC,TRVACT               ;IF ANY FOUND TAKE ACTION
4373  025146  000137  024414                  JMP     TRVNOB                  ;THEN UPDATE R3 TO NEXT NODE -NO BRANCH
4374  025152  000137  024404          3$:     JMP     TRVBRC                  ;NONE FOUND, TAKE MISS BRANCH
4375
4376  025156  005005                  TRVALN: CLR     R5                      ;CLEAR ALPHANUM FOUND FLAG
4377  025160  121427  000060          10$:    CMPB    (R4),#60                ;SEE IF CHAR. LESS THAN A "0"
4378  025164  002417                          BLT     2$                      ;BR IF YES (NOT NUMERIC OR ALPHA)
4379  025166  121427  000072                  CMPB    (R4),#72                ;SEE IF CHAR. GREATER THAN A "9"
4380  025172  003003                          BGT     1$                      ;BR IF YES (NOT NUMERIC)
4381  025174  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR.
4382  025176  005205                          INC     R5                      ;INDICATE A NUMERIC FOUND
4383  025200  000767                          BR      10$                     ;GO LOOK AT NEXT CHAR.
4384  025202  121427  000101          1$:     CMPB    (R4),#101               ;SEE IF CHAR. LESS THAN A "A"
4385  025206  002406                          BLT     2$                      ;BR IF YES (NOT ALPHA)
4386  025210  121427  000132                  CMPB    (R4),#132               ;SEE IF CHAR. GREATER THAN A "Z"
4387  025214  003003                          BGT     2$                      ;BR IF YES (NOT ALPHA)
4388  025216  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR
4389  025220  005205                          INC     R5                      ;INDICATE AN ALPHA FOUND
4390  025222  000756                          BR      10$                     ;GO LOOK AT NEXT CHAR.
4391  025224  005705                  2$:     TST     R5                      ;SEE IF ANY ALPHANUM'S WERE FOUND
4392  025226  001404                          BEQ     3$                      ;BR IF NO
4393  025230  004737  024364                  JSR     PC,TRVACT               ;IF ANY FOUND TAKE ACTION
4394  025234  000137  024414                  JMP     TRVNOB                  ;THEN UPDATE R3 TO NEXT NODE -NO BRANCH
4395  025240  000137  024404          3$:     JMP     TRVBRC                  ;NONE FOUND, TAKE MISS BRANCH
```

```
4396
4397
4398
4399  025244  010401                TRVSTR: MOV     R4,R1           ;POINT R1 TO CMD STRING
4400  025246  010305                        MOV     R3,R5
4401  025250  062705  000006                ADD     #6,R5           ;POUNT R5 TO MATCH STRING FROM CLI NODE
4402  025254  005037  003140                CLR     P$CNT           ;CLEAR CHAR MATCH COUNT
4403  025260  105715                2$:     TSTB    (R5)            ;SEE IF END OF MATCH STRING YET
4404  025262  001411                        BEQ     10$             ;BR IF YES
4405  025264  105711                        TSTB    (R1)            ;SEE IF END OF CMD LINE YET
4406  025266  001407                        BEQ     10$             ;BR IF YES
4407  025270  121115                        CMPB    (R1),(R5)       ;SEE IF CHARACTERS MATCH
4408  025272  001005                        BNE     10$             ;BR IF NO
4409  025274  005237  003140                INC     P$CNT           ;MATCH -INCREMENT MATCH COUNT
4410  025300  005201                        INC     R1              ;UPDATE STRING POINTERS
4411  025302  005205                        INC     R5
4412  025304  000765                        BR      2$              ;BR TO CONTINUE CHECKING CHARS.
4413
4414  025306  005737  003140        10$:    TST     P$CNT           ;WHEN DONE SEE IF ANY MATCHES FOUND
4415  025312  001406                        BEQ     15$             ;BR IF NO, GO TAKE THE MISS BRANCH
4416  025314  010104                        MOV     R1,R4           ;POINT CMD POINTER TO END OF STRING &
4417  025316  004737  024364                JSR     PC,TRVACT       ;IF A MATCH FOUND, GO DO MATCH ACTION
4418  025322  066303  000004                ADD     4(R3),R3        ;UPDATE R3 TO NEXT NODE (NO BRANCH)
4419  025326  000207                        RTS     PC              ; (NO RETURN THRU TRVNOB SINCE DIFFERNT
4420                                                                ;  DISPLACEMENT DUE TO MATCH STRING)
4421  025330  000137  024404        15$:    JMP     TRVBRC          ; GO TAKE BRANCH
4422
4423                                                                ; (PARSED OK), -1 IF ILL CMD.....
4424                                ;-----------------------------------------------------------------
4425
```

```
 4426                                   .SBTTL  REPORT CODING SECTION
 4427
 4428
 4429                                   ;++
 4430                                   ; THE REPORT CODING SECTION CONTAINS THE
 4431                                   ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
 4432                                   ;--
 4433
 4434  025334                                   BGNRPT
 4435  025334                                                                  L$RPT::
 4436
 4437
 4438  025334  004737  020610                   JSR     PC,REPORT             ;CALL SUBROUTINE TO DUMP EVENT LOG
 4439                                                                          ; AND BASE TABLE
 4440
 4441
 4442
 4443
 4444  025340                                   ENDRPT
 4445  025340                                                                  L10010:
 4446  025340  104425                                                                 TRAP    C$RPT
```

```
 4447                                   .SBTTL  PROTECTION TABLE
 4448
 4449                                   ;++
 4450                                   ; THIS TABLE IS USED BY THE RUNTIME SERVICES
 4451                                   ; TO PROTECT THE LOAD MEDIA.
 4452                                   ;--
 4453
 4454  025342                                   BGNPROT
 4455  025342                                                                  L$PROT::
 4456
 4457  025342  177777                          -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
 4458  025344  177777                          -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
 4459  025346  177777                          -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER
 4460
 4461  025350                                   ENDPROT
 4462
```

```
                        .SBTTL  INITIALIZE SECTION

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

025350                          BGNINIT
025350                                                          L$INIT::

025350 005037 003010            CLR     KEYWD1                  ;INIT COMMAND STORAGE VARIABLE
025354 005737 006562            TST     DCLFLG                  ;CLEANUP AND EXIT? REV B EC
025360 001403                   BEQ     INIT1                   ;NO,BRANCH REV B EC
025362 005037 006562            CLR     DCLFLG                  ;CLEAR FLAG REV B EC
025366                          DOCLN                           ;GO CLEANUP AND EXIT REV B EC
025366 104444                                                           TRAP    C$DCLN

025370 012737 177777 006564 INIT1:  MOV     #-1,RESFLG          ;SET RESTART FLAG
025376                          READEF  #EF.START               ;IF HERE CAUSE OF START,DO SOME INIT
025376 012700 000040                                                    MOV     #EF.START,R0
025402 104447                                                           TRAP    C$REFG
025404                          BCOMPLETE       START
025404 103417                                                           BCS     START
025406                          READEF  #EF.RESTART             ;IF HERE CAUSE OF RESTART, DO SOME INIT
025406 012700 000037                                                    MOV     #EF.RESTART,R0
025412 104447                                                           TRAP    C$REFG
025414                          BCOMPLETE       RESTRT
025414 103513                                                           BCS     RESTRT
025416                          READEF  #EF.CONTINUE            ;SEE IF WE'RE HERE CAUSE OF A CONTINUE
025416 012700 000036                                                    MOV     #EF.CONTINUE,R0
025422 104447                                                           TRAP    C$REFG
025 24                          BNCOMPLETE      S1              ;BR IF NOT HERE CAUSE OF  CONITNUE
025424 103002                                                           BCC     S1
025426 000137 026116            JMP     ENDIT                   ;JMP IF HERE CAUSE OF A CONTINUE
025432                      S1: READEF  #EF.NEW                 ;SEE IF THIS IS A 'NEW PASS'
025432 012700 000035                                                    MOV     #EF.NEW,R0
025436 104447                                                           TRAP    C$REFG
025440                          BCOMPLETE       NEW             ;IF YES, BR AROUND LOGUNIT # SETUP
025440 103521                                                           BCS     NEW
025442 000523                   BR      GETPRM

025444 005037 006564        START:  CLR     RESFLG              ;CLEAR RESTART FLAG SINCE HERE ON START
025450 005037 006624            CLR     CLKVEC                  ;CLEAR CLK VECTOR PTR. AS A FLAG IN
                                                                ; NO CLOCK IS FOUND.
025454 012702 006620            MOV     #CLKCSR,R2              ;SETUP R2 AS A PTR. TO CLOCK INFO BLOCK
025460                          CLOCK   L,R1                    ;LOOK FOR A LINE CLOCK
025460 012700 000114                                                    MOV     #'L,R0
025464 104462                                                           TRAP    C$CLCK
025466 010001                                                           MOV     R0,R1
025470                          BNCOMPLETE      S2              ; IF NONE THERE GO LOOK FOR A P-CLOCK
025470 103006                                                           BCC     S2
025472 004737 017730            JSR     PC,CLKSET               ; GO SET UP CLOCK INFO TABLE & CLK VEC.
025476 012737 000100 006630     MOV     #LCLKEN,CLKEN           ;SETUP THE ENABLE LINE CLOCK DATA
025504 000457                   BR      RESTRT

025506                      S2: CLOCK   P,R1                    ;LOOK FOR A P-CLOCK SINCE NO LINE CLOCK
```

```
4519  025506  012700  000120                                                    MOV     #'P,R0
4520  025512  104462                                                            TRAP    C$CLCK
4521  025514  010001                                                            MOV     R0,R1
4522  025516                        BNCOMPLETE      S3              ; IF NONE THERE GO SEE IF THIS IS LSI
4523  025516  103017                                                            BCC     S3
4524  025520  004737  017730        JSR     PC,CLKSET               ; ELSE GO SET UP CLOCK INFO & VECTOR
4525  025524  062737  000002  006620  ADD     #2,CLKCSR               ;POINT CLKCSR TO P-CLK COUNT SET REG.
4526  025532  012777  001600  161060  MOV     #PCLKCT,@CLKCSR         ;LOAD CLK SET REG. WITH COUNT VALUE
4527  025540  162737  000002  006620  SUB     #2,CLKCSR               ;POINT CLKCSR BAC TO P-CLK CSR
4528  025546  012737  000111  006630  MOV     #PCLKEN,CLKEN           ;SETUP THE ENABLE THE P-CLK DATA
4529  025554  000433                  BR      RESTRT
4530
4531  025556                  S3:     READBUS                         ;READ BUS TYPE TO SEE IF ON AN LSI
4532  025556  104407                                                            TRAP    C$RDBU
4533  025560                        BNCOMPLETE      S4              ;BR IF NOT, NO CHANCE OF A CLOCK
4534  025560  103021                                                            BCC     S4
4535  025562  012737  000100  006624  MOV     #100,CLKVEC             ;LOAD 100 AS CLK VECTOR
4536  025570  005037  006622          CLR     CLKBR                   ;LOAD 0 AS CLK INT. LEVEL
4537  025574  012737  006630  006620  MOV     #CLKEN,CLKCSR           ;KLUDGE UP THE CSR & ENABLE DATA LOCS
4538  025602                        GMANID  L5060,CLKHZ,D,377,50.,60.,YES
4539  025602  104443                                                            TRAP    C$GMAN
4540  025604  000406                                                            BR      10000$
4541  025606  006626                                                            .WORD   CLKHZ
4542  025610  000052                                                            .WORD   T$CODE
4543  025612  014063                                                            .WORD   L5060
4544  025614  000377                                                            .WORD   377
4545  025616  000062                                                            .WORD   T$LOLIM
4546  025620  000074                                                            .WORD   T$HILIM
4547  025622                                                          10000$:
4548  025622  000410                  BR      RESTRT
4549
4550  025624                  S4:     PRINTF  #BDCLK                  ;INFORM OPR. NO CLOCK, & EXIT INIT
4551  025624  012746  014202                                                    MOV     #BDCLK,-(SP)
4552  025630  012746  000001                                                    MOV     #1,-(SP)
4553  025634  010600                                                            MOV     SP,R0
4554  025636  104417                                                            TRAP    C$PNTF
4555  025640  062706  000004                                                    ADD     #4,SP
4556
4557  025644  005037  006632  RESTRT: CLR     TIMMIN                  ;CLEAR TIME SINCE START LOCATIONS
4558  025650  005037  006634          CLR     TIMSEC
4559  025654  013737  006626  006636  MOV     CLKHZ,TIMTCK            ;LOAD TICKS/SEC
4560  025662  012702  006650          MOV     #EVTLOG,R2              ;INIT EVENT TABLE TO ALL 1'S AFTER EACH
4561  025666  010237  006646          MOV     R2,EVTPTR               ; START OR RES AND INIT TABLE POINTER
4562  025672  012722  177777  1$:     MOV     #-1,(R2)+
4563  025676  020227  007552          CMP     R2,#EVTEND              ;SEE IF REACHED END OF TABLE
4564  025702  001373                  BNE     1$                      ;LOOP UNTIL DONE
4565
4566  025704  012737  177777  006556  NEW:    MOV     #-1,LOGUNT              ;INITIALIZE LOGICAL UNIT #
4567
4568  025712  005237  006556  GETPRM: INC     LOGUNT                  ;POINT TO NEXT LOGICAL UNIT
4569  025716  023737  006556  002012  CMP     LOGUNT,L$UNIT           ;SEE IF PAST MAX. LOG. UNIT #
4570  025724  002367                  BGE     NEW                     ;BR IF YES, AND START OVER
4571
4572  025726                        GPHARD  LOGUNT,R1               ;GET THE P-TABLE FOR THIS LOG. UNIT
4573  025726  013700  006556                                                    MOV     LOGUNT,R0
4574  025732  104442                                                            TRAP    C$GPHRD
```

```
025734  010001                                                                                  MOV     R0,R1
025736                                  BNCOMPLETE      GETPRM          ;IF NO P-TABLE AVAIL., GO GET NEXT ONE
025736  103365                                                                                  BCC     GETPRM

025740  011137  006572                  MOV     (R1),FHDPLX             ;PUT FULL OR HALF DUPLEX ANSWER IN LOC.


                                        ;DEVICE DEPENDENT PART OF GETTING INFO FROM P-TABLE

025744  016137  000002  011452          MOV     2(R1),RXCSR             ;STORE AWAY CSR ADDRESSES


025752  016137  000002  011454          MOV     2(R1),PCSAR
025760  062737  000002  011454          ADD     #2,PCSAR
025766  016137  000002  011456          MOV     2(R1),RDSR
025774  062737  000002  011456          ADD     #2,RDSR
026002  016137  000002  011460          MOV     2(R1),TXCSR
026010  062737  000004  011460          ADD     #4,TXCSR
026016  016137  000002  011462          MOV     2(R1),TDSR
026024  062737  000006  011462          ADD     #6,TDSR

026032  016137  000004  011464          MOV     4(R1),INVEC             ;STORE AWAY INPUT INTERRUPT VECTOR
026040  016137  000004  011466          MOV     4(R1),OUTVEC
026046  062737  000004  011466          ADD     #4,OUTVEC               ;BUILD OUTPUT INTERRUPT VECTOR
026054  016137  000006  011470          MOV     6(R1),INTPRI            ;STORE AWAY INTERRUPT PRIORITY
026062  016137  000014  011524          MOV     14(R1),RNODE            ;STORE AWAY THE REMOTE NODE TYPE
026070  016137  000010  037126          MOV     10(R1),MPPTP            ;MULTI-POINT = 1
026076  001004                          BNE     10$                     ;IF MTP THEN GET TRIB ADDRESS FROM P-TABLE
026100  112737  000001  037061          MOVB    #1,TRIBN                ;PTP TRIB ADDRESS ALWAYS 1
026106  000403                          BR      ENDIT                   ;BRANCH
026110  116137  000012  037061  10$:    MOVB    12(R1),TRIBN            ;STORE AWAY TRIB NUMBER
026116                          ENDIT:
026116                                  SETVEC  CLKVEC,#CLKINT,#340     ;SETUP CLOCK VECTOR
026116  012746  000340                                                                          MOV     #340,-(SP)
026122  012746  017754                                                                          MOV     #CLKINT,-(SP)
026126  013746  006624                                                                          MOV     CLKVEC,-(SP)
026132  012746  000003                                                                          MOV     #3,-(SP)
026136  104437                                                                                  TRAP    C$SVEC
026140  062706  000010                                                                          ADD     #10,SP


                                        ;DEVICE DEPENDENT VECTOR SETUP

026144                                  SETVEC  INVEC,#DVRXI,#PRI04     ;SETUP INPUT INTERRUPT VECTOR
026144  012746  000200                                                                          MOV     #PRI04,-(SP)
026150  012746  035554                                                                          MOV     #DVRXI,-(SP)
026154  013746  011464                                                                          MOV     INVEC,-(SP)
026160  012746  000003                                                                          MOV     #3,-(SP)
026164  104437                                                                                  TRAP    C$SVEC
026166  062706  000010                                                                          ADD     #10,SP
026172                                  SETVEC  OUTVEC,#DVTXI,#PRI04    ;SETUP OUTPUT INTERRUPT VECTOR
026172  012746  000200                                                                          MOV     #PRI04,-(SP)
026176  012746  036254                                                                          MOV     #DVTXI,-(SP)
026202  013746  011466                                                                          MOV     OUTVEC,-(SP)
026206  012746  000003                                                                          MOV     #3,-(SP)
026212  104437                                                                                  TRAP    C$SVEC
026214  062706  000010                                                                          ADD     #10,SP
```

```
4631
4632  026220                          SETPRI  #PRI00              ;SET THE "RUN" PRIORITY TO 0
4633  026220  012700  000000                                              MOV     #PRI00,R0
4634  026224  104441                                                      TRAP    C$SPRI
4635  026226                          EXIT    INIT
4636  026226  104432                                                      TRAP    C$EXIT
4637  026230  000002                                                      .WORD   L10012-.
4638
4639
4640                                  .EVEN
4641
4642  026232                          ENDINIT
4643  026232                                                  L10012:
4644  026232  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.
                                ;--

026234                                  BGNAUTO
026234                                                          L$AUTO::

026234                                  ENDAUTO
026234                                                          L10013:
026234  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
                                CMP     #EXIT,KEYWD1    ;'EXIT' COMMAND ?
                                BEQ     EXITCLN         ;YES,BRANCH

                        ;;LOG ^C ABORT IN EVENT LOG
                                MOV     #ABO,TEMP       ;EVENT TYPE
                                MOV     @PVAR,TEMP4     ;START TIME OUTS
                                MOV     PSCNT,TEMP2     ;PASSES
                                MOV     ERRCNT,TEMP3    ;ERRORS
                                JSR     PC,LOGS5        ;GO LOG IT

                        EXITCLN:BRESET                  ;RESET
                                                                        TRAP    C$RESET
                                EXIT    CLN
                                                                        TRAP    C$EXIT
                                                                        .WORD   L10014-.


                                .EVEN

                                ENDCLN
                                                                L10014:
                                                                        TRAP    C$CLEAN
```

```
 4697                                   .SBTTL  DROP UNIT SECTION
 4698
 4699                                   ;++
 4700                                   ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
 4701                                   ; TO NO LONGER BE TESTED.
 4702                                   ;--
 4703
 4704  026324                                   BGNDU
 4705  026324                                                                  L$DU::
 4706
 4707
 4708  026324                                   EXIT    DU
 4709  026324  000167                                                                 .WORD   J$JMP
 4710  026326  000000                                                                 .WORD   L10015-2-.
 4711
 4712
 4713                                   .EVEN
 4714
 4715  026330                                   ENDDU
 4716  026330                                                                  L10015:
 4717  026330  104453                                                                 TRAP    C$DU
```

```
                        .SBTTL  ADD UNIT SECTION

                        ;++
                        ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
                        ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
                        ;  TO THE TEST CYCLE.
                        ;--

026332                          BGNAU
026332                                                          L$AU::


026332                          EXIT    AU
026332  000167                                                          .WORD   J$JMP
026334  000000                                                          .WORD   L10016-2-.


                        .EVEN

                        ENDAU
026336                                                          L10016:
026336                                                                  TRAP    C$AU
026336  104452
```

```
                                .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.
                                ;--


026340                                  BGNTST
026340                                                                  T1::


                                .SBTTL          PROGRAM SETUP SECTION

026340  013777  006630  160252           MOV     CLKEN,@CLKCSR   ;ENABLE THE CLOCK

026346                          GTXRXB:
026346  005001                  GTRA2:  CLR     R1
026350  012737  000001  006640           MOV     #1,TIMER1       ;SET TIMER TO COUNT 1 TICK
026356  005737  006640           1$:     TST     TIMER1          ;CHECK FOR IT TO BE COUNTED OFF
026362  001412                          BEQ     GTRA3           ;BRANCH IF CLOCK EXISTS (COUNTED A TICK)
026364  005301                          DEC     R1
026366  001373                          BNE     1$              ;KEEP CHECKING UNTIL R1 DOES FULL COUNTDWN
026370                          .PRINTF #NOCLK                  ;PRINT BAD CLK MSG AND WARN OF HANG IF TIMEOUT
026370  012746  014226                                                  MOV     #NOCLK,-(SP)
026374  012746  000001                                                  MOV     #1,-(SP)
026400  010600                                                          MOV     SP,R0
026402  104417                                                          TRAP    C$PNTF
026404  062706  000004                                                  ADD     #4,SP

026410  005737  006564           GTRA3:  TST     RESFLG          ;SEE IF HERE AFTER A RESTART.
026414  001112                          BNE     GTRA5           ;BR IF HERE CAUSE OF A RESTART

                                ; CLEAR COUNTS AND SET UP DEFAULTS

026416  005037  006530           GTRA4:  CLR     TOTCC           ;CLEAR TOTAL CHAR. COUNT TEMP. LOC.
026422  005037  006464                   CLR     TTOTCC          ; CLEAR TOTAL CHAR. COUNT FOR TX BUFF
026426  005037  006450                   CLR     CTOTCC          ; CLEAR TOTAL CHAR. COUNT FOR CMP BUFF
026432  012701  006150                   MOV     #PTRTAB,R1      ;INIT TRANSMIT MESSAGE POINTER
026436  010137  006442                   MOV     R1,TXPTR
026442  005037  006440                   CLR     RXPTR           ; ZERO RX POINTER

026446  012737  006244  006444           MOV     #PTR13,CMPPTR   ;INIT COMPARE MESSAGE POINTER

026454  012737  000005  006516           MOV     #5,MSGTYP       ;SET UP DEFAULT MSG TYPE (QUICK FOX - ITEP MSG)
026462  013737  002162  006520           MOV     MSG5C,CURCC     ;SET UP DEFAULT CHAR COUNT
026470  012737  003150  006466           MOV     #TXBUF,TCURAD   ;SET UP CURRENT ADD TO START OF TX BUFFER
026476  012737  005150  006452           MOV     #CMPBUF,CCURAD  ;SET UP CURRENT ADD TO START OF CMP BUFFER

026504  013737  006466  006526           MOV     TCURAD,CURADD   ;SETUP CURRENT ADDR TO START OF TXBUF
026512  013737  006442  006524           MOV     TXPTR,CPTR      ;SETUP CURRENT POINTER TABLE POINTER FOR TXBUF
026520  004737  023362                   JSR     PC,BLDBUF       ; GO BUILD POINTER TABLE AND BUFFER
026524  012737  000001  006462           MOV     #1,TXMTOT       ;BUMP TOTAL MESSAGE COUNT
```

```
4798
4799  026532  013737  006444  006524          MOV     CMPPTR,CPTR     ;SET UP START OF COMPARE POINTER TABLE
4800  026540  013737  006452  006526          MOV     CCURAD,CURADD   ;SET UP CURRENT ADDR. TO START OF CMPBUF
4801  026546  012737  000005  006516          MOV     #5,MSGTYP
4802  026554  013737  002162  006520          MOV     MSG5C,CURCC
4803  026562  004737  023362                  JSR     PC,BLDBUF       ;PUT DEFAULT MESSAGE INTO CMPBUF
4804  026566  012737  000001  006446          MOV     #1,CMPTOT       ;BUMP THE COMP MESG COUNT
4805  026574  012737  000003  006566          MOV     #ACT,MODTYP     ;SET DEFAULT MODE= ACTIVE
4806  026602  005037  006570                  CLR     MLTYP           ;SET DEFAULT MAINTENANCE LOOP MODE =NONE
4807  026606  012737  000001  006576          MOV     #1,RPASS        ;SET UP DEFAULT 'RUN PASS' COUNT TO 1
4808  026614  012737  000002  006574          MOV     #2,PARAM        ;SET UP PROG. PARAMETERS - DATACHECKING ENABLED
4809                                                                  ;                          OPERATOR STATUS MSGS. PRINT OFF
4810  026622                                  PRINTF  #HLP0
4811  026622  012746  012215                                                  MOV     #HLP0,-(SP)
4812  026626  012746  000001                                                  MOV     #1,-(SP)
4813  026632  010600                                                          MOV     SP,R0
4814  026634  104417                                                          TRAP    C$PNTF
4815  026636  062706  000004                                                  ADD     #4,SP
4816  026642                          GTRA5:  SETVEC  INVEC,#DVRXI,#PRI04     ;DEFAULT NON-PROTOCOL RX INTER. VECTOR
4817  026642  012746  000200                                                  MOV     #PRI04,-(SP)
4818  026646  012746  035554                                                  MOV     #DVRXI,-(SP)
4819  026652  013746  011464                                                  MOV     INVEC,-(SP)
4820  026656  012746  000003                                                  MOV     #3,-(SP)
4821  026662  104437                                                          TRAP    C$SVEC
4822  026664  062706  000010                                                  ADD     #10,SP
4823  026670  042737  000300  006574          BIC     #PRORUN!ABORT,PARAM     ;INIT PROTOCOL VARIABLES
4824  026676  013737  006566  007706          MOV     MODTYP,DEV1
4825  026704  013737  006570  007710          MOV     MLTYP,DEV2
4826  026712  013737  006576  007712          MOV     RPASS,DEV3
4827  026720  013737  006574  007714          MOV     PARAM,DEV4
4828  026726  004737  023712                  JSR     PC,SHWOP        ;PRINT TO OPERATOR THE CURRENT MODE.....
4829
4830  026732                                  MANUAL                  ;SEE IF MANUAL INTERVENTION ALLOWED
4831  026732  104450                                                          TRAP    C$MANI
4832  026734                                  BCOMPLETE       GETCL   ;  BR IF YES (UAM=0 AND NOT CHAINED)
4833  026734  103412                                                          BCS     GETCL
4834  026736  005737  006576                  TST     RPASS           ;SEE IF THIS IS FIRST 'DCLT PASS"
4835  026742  001002                          BNE     1$              ; BR IF NOT COMPLETED 1 PASS
4836  026744                                  EXIT    TST             ; IF DONE 1 PASS IN UNATTENDED MODE - EXIT
4837  026744  104432                                                          TRAP    C$EXIT
4838  026746  017276                                                          .WORD   L10017-.
4839  026750  012737  000001  006570  1$:     MOV     #TTL,MLTYP      ;SET UP DEFAULT FOR UNATTENDED MODE
4840  026756  000137  031734                  JMP     GTR9            ;  'R M=ACT/LO=I/PAS=1/NOST/CH" AND RUN
4841
4842                                  .SBTTL          COMMAND LINE FETCH & INTERPRETATION SECTION
4843
4844  026762  105037  003147          GETCL:  CLRB    P$GDBD          ;CLEAR CMD LINE PARSING ERROR FLAGS
4845  026766  105037  003146                  CLRB    P$NNUF
4846  026772                                  GMANID  CLI$PM,CMDBUF,A,-1,1,72.,NO     ;GET A COMMAND LINE FROM OPR.
4847  026772  104443                                                          TRAP    C$GMAN
4848  026774  000406                                                          BR      10000$
4849  026776  002666                                                          .WORD   CMDBUF
4850  027000  000142                                                          .WORD   T$CODE
4851  027002  011572                                                          .WORD   CLI$PM
4852  027004  177777                                                          .WORD   -1
4853  027006  000001                                                          .WORD   T$LOLIM
```

```
4854  027010  000110                                                          .WORD   T$HILIM
4855  027012                                                          10000$:
4856  027012  012737  002666  003132        MOV     #CMDBUF,P$BUFA
4857  027020  012737  007716  003134        MOV     #CLITRE,P$TREE
4858  027026  012737  027736  003136        MOV     #CLIACT,P$ACT
4859  027034  005037  003012                CLR     QUALFG                  ;CLEAR QUALIFIER FLAG LOCATION
4860  027040  004737  024244                JSR     PC,P$TRV                ;GO PARSE COMMAND LINE
4861  027044  105737  003147                TSTB    P$GDBD                  ;SEE IF PARSED OK OR AN ERROR
4862  027050  001412                        BEQ     1$
4863  027052                                PRINTF  #CLIERM
4864  027052  012746  011605                                                  MOV     #CLIERM,-(SP)
4865  027056  012746  000001                                                  MOV     #1,-(SP)
4866  027062  010600                                                          MOV     SP,R0
4867  027064  104417                                                          TRAP    C$PNTF
4868  027066  062706  000004                                                  ADD     #4,SP
4869  027072  000137  026762                JMP     GETCL
4870  027076  105737  003146        1$:     TSTB    P$NNUF                  ;SEE IF INCOMPLETE COMMAND TYPED
4871  027102  001412                        BEQ     10$
4872  027104                                PRINTF  #CLINUF
4873  027104  012746  011635                                                  MOV     #CLINUF,-(SP)
4874  027110  012746  000001                                                  MOV     #1,-(SP)
4875  027114  010600                                                          MOV     SP,R0
4876  027116  104417                                                          TRAP    C$PNTF
4877  027120  062706  000004                                                  ADD     #4,SP
4878  027124  000137  026762                JMP     GETCL
4879
4880  027130  023727  003010  000060  10$:  CMP     KEYWD1,#SETET           ;WAS"SET EXPECT=TRANSMIT" TYPED ? REVB EC
4881  027136  001711                        BEQ     GETCL                   ;YES,BRANCH REV B EC
4882  027140  023727  003010  000005        CMP     KEYWD1,#HLP             ;SEE IF HELP WAS TYPED
4883  027146  001705                        BEQ     GETCL                   ;GO GET CMD AGAIN IF YES
4884  027150  023727  003010  000055        CMP     KEYWD1,#PRNT            ;SEE IF PRINT WAS TYPED
4885  027156  001701                        BEQ     GETCL                   ; GO GET CMD AGAIN IF YES
4886  027160  023727  003010  000004        CMP     KEYWD1,#RUN             ;SEE IF RUN WAS TYPED
4887  027166  001002                        BNE     11$                     ; BR IF NO
4888  027170  000137  031734                JMP     GTR9                    ; START EXEC. IF YES
4889  027174  023727  003010  000052  11$:  CMP     KEYWD1,#DMPS            ;SEE IF DUMP WAS TYPED
4890  027202  001004                        BNE     12$                     ; BR IF NO
4891  027204  004737  023126                JSR     PC,DUMPSR               ; ELSE, DUMP PART OF MEMORY
4892  027210  000137  026762                JMP     GETCL                   ; THEN RETURN TO GET ANOTHER CMD.
4893  027214  023727  003010  000057  12$:  CMP     KEYWD1,#EXIT            ;EXIT ? REV B EC
4894  027222  001005                        BNE     13$                     ;NO,BRANCH REV B EC
4895  027224  C12737  000001  006562        MOV     #1,DCLFLG               ;SET CLEANUP FLAG REV B EC
4896  027232                                EXIT    TST                     ;GO BACK TO INIT REV B EC
4897  027232  104432                                                          TRAP    C$EXIT
4898  027234  017010                                                          .WORD   L10017-.
4899
4900  027236  023727  003010  000001  13$:  CMP     KEYWD1,#CLEAR           ;SEE IF CLEAR WAS TYPED
4901  027244  001646                        BEQ     GETCL                   ; IF YES, BACK TO GET ANOTHER CMD.
4902  027246  023727  003010  000002        CMP     KEYWD1,#SHOW            ;SEE IF SHOW WAS TYPED
4903  027254  001642                        BEQ     GETCL                   ; IF YES, BACK TO GET ANOTHER CMD.
4904  027256  023727  003010  000010  4$:   CMP     KEYWD1,#SETEXP          ;SEE IF SET EXPECTED
4905  027264  001512                        BEQ     2$                      ; BR IF YES (A SETEXP WAS TYPED)
4906  027266  013737  006464  006530  5$:   MOV     TTOTCC,TOTCC
4907  027274  023727  006530  001000        CMP     TOTCC,#BUFLIM           ;SEE IF BUFFER ALREADY FULL
4908  027302  002414                        BLT     15$                     ; BR IF NOT FULL (BUFLIM # OF CHARS.)
4909  027304                                PRINTF  #MSGTRN,#BUFEX          ; ELSE TELL OPR. AND DON'T BUILD MSG.
```

```
                                                                                MOV     #BUFEX,-(SP)
                                                                                MOV     #MSGTRN,-(SP)
                                                                                MOV     #2,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTF
                                                                                ADD     #6,SP
                        JMP     GETCL                   ; THEN GO GET A NEW COMMAND
                15$:    TST     TTOTCC                  ;IF FIRST "SET" THEN GET RID OF DEFAULT
                        BNE     6$
                        CLR     TXMTOT
006442          6$:     MOV     #PTRTAB,TXPTR           ;GET POSITION OF END OF TX LIST
                        MOV     TXMTOT,R1
                        CMP     R1,#MSGLIM              ;SEE IF MSG COUNT EXCEEDED.
                        BLT     17$                     ; BR IF NO
                        PRINTF  #MSGTRN,#TABEX          ; ELSE TELL OPR. AND DON'T BUILD MSG.
                                                                                MOV     #TABEX,-(SP)
                                                                                MOV     #MSGTRN,-(SP)
                                                                                MOV     #2,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTF
                                                                                ADD     #6,SP
                        JMP     GETCL                   ; THEN GO GET A NEW COMMAND.
                17$:    ASL     R1                      ;# OF MSGS *4 = NEXT FREE PTR BLOCK
                        ASL     R1
                        ADD     R1,TXPTR
006524                  MOV     TXPTR,CPTR              ;SETUP CHAR. COUNT, CURRENT ADDR, & PTR
006526                  MOV     TCURAD,CURADD
                        JSR     PC,ADDCC                ;ADD IN CHAR. COUNT AND CHECK TOTAL
                        JSR     PC,BLDBUF               ;GO BUILD MESSAGE IN BUFFER AND PTRS.
006442                  MOV     CPTR,TXPTR
006464                  MOV     TOTCC,TTOTCC            ;UPDATE CHAR. COUNT, CURR ADDR, & PTR
006466                  MOV     CURADD,TCURAD
                        INC     TXMTOT
                        DEC     QUALVL                  ;DEC THE COPY COUNT
                        BNE     5$
                        JMP     GETCL

006530          2$:     MOV     CTOTCC,TOTCC            ;SETUP CHAR. COUNT, CURR. ADDR. & PTR
001000                  CMP     TOTCC,#BUFLIM           ;SEE IF BUFFER ALREADY FULL
                        BLT     16$                     ; BR IF NOT FULL (BUFLIM # OF CHARS.)
                        PRINTF  #MSGTRN,#BUFEX          ; ELSE TELL OPR. AND DON'T BUILD MSG.
                                                                                MOV     #BUFEX,-(SP)
                                                                                MOV     #MSGTRN,-(SP)
                                                                                MOV     #2,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTF
                                                                                ADD     #6,SP
                        JMP     GETCL                   ; THEN GO GET A NEW COMMAND
                16$:    TST     CTOTCC                  ;IF FIRST "SET" THEN GET RID OF DEFAULT
                        BNE     7$
                        CLR     CMPTOT
                7$:
006444                  MOV     #PTR13,CMPPTR           ;INIT COMPARE MESSAGE POINTER
                        MOV     CMPTOT,R1
                        CMP     R1,#MSGLIM              ;SEE IF MSG COUNT EXCEEDED.
                        BLT     18$                     ; BR IF NO
```

```
4966  027612
4967  027612  012746  014307                    PRINTF  #MSGTRN,#TABEX          ; ELSE TELL OPR. AND DON'T BUILD MSG.
4968  027616  012746  014365                                                            MOV     #TABEX,-(SP)
4969  027622  012746  000002                                                            MOV     #MSGTRN,-(SP)
4970  027626  010600                                                                    MOV     #2,-(SP)
4971  027630  104417                                                                    MOV     SP,R0
4972  027632  062706  000006                                                            TRAP    C$PNTF
4973  027636  000137  026762                                                            ADD     #6,SP
                                                  JMP     GETCL                   ; THEN GO GET A NEW COMMAND.
4974  027642  006301                    18$:     ASL     R1                      ;# OF MSGS *4 = NEXT FREE PTR BLOCK
4975  027644  006301                             ASL     R1
4976  027646  060137  006444                     ADD     R1,CMPPTR
4977  027652  013737  006444  006524             MOV     CMPPTR,CPTR
4978  027660  013737  006452  006526             MOV     CCURAD,CURADD
4979  027666  004737  023264                     JSR     PC,ADDCC                ;ADD IN XHAR. COUNT AND CHECK TOTAL
4980  027672  004737  023362                     JSR     PC,BLDBUF
4981  027676  013737  006524  006444             MOV     CPTR,CMPPTR
4982  027704  005237  006446                     INC     CMPTOT
4983  027710  013737  006526  006452             MOV     CURADD,CCURAD           ;UPDATE CHAR. COUNT, CURR ADDRR. & PTR
4984  027716  013737  006530  006450             MOV     TOTCC,CTOTCC
4985  027724  005337  003014                     DEC     QUALVL                  ;IF COPY WAS GIVEN, PUT MSG IN BUFF
4986  027730  001270                             BNE     2$                      ; AGAIN
4987  027732  000137  026762                     JMP     GETCL                   ;GO BACK UNTIL GET A "RUN"
4988
4989
4990
4991
4992
```

```
                                        .SBTTL          ACTION TABLE AND ROUTINES
                                        ;       USER MUST CLEAR/SET P$GDBD IF USE "CLIBIF" IN CONNECTION WITH ACTION
                                        ;       R2 WILL HOLD ACTION CODE FROM PARSING (CLI) NODE
                                        ;
027736                                  CLIACT:
027736  006302                                  ASL     R2              ;MULTIPLY ACTION CODE BY 2
027740  016202  027754                          MOV     10$(R2),R2      ;OFFSET VALUE
027744  062702  027754                          ADD     #10$,R2         ;ADD BASE VALUE
027750  004712                                  JSR     PC,(R2)         ;GO DO ACTION
027752  000207                                  RTS     PC              ;RETURN TO TRVACT:
                                                                        ;BRIEF DESCRIPTION OF ACTONS TAKEN
027754  000150                          10$:    .WORD   ACTNUL-10$      ;NULL
027756  000152                                  .WORD   ACTCLR-10$      ;CLEAR
027760  000162                                  .WORD   ACTSHO-10$      ;SHOW
027762  001550                                  .WORD   ACTCHK-10$      ;CHECK
027764  000262                                  .WORD   ACTRUN-10$      ;RUN
027766  000172                                  .WORD   ACTHLP-10$      ;HELP
027770  000306                                  .WORD   ACTCSE-10$      ;CLEAR OR SHOW EXPECT
027772  000400                                  .WORD   ACTCST-10$      ;CLEAR OR SHOW TRANSMIT
027774  000722                                  .WORD   ACTSTE-10$      ;SET EXPECT
027776  000732                                  .WORD   ACTSTT-10$      ;SET TRANSMIT
030000  000750                                  .WORD   ACTSZE-10$      ;SIZE
030002  000760                                  .WORD   ACTCOP-10$      ;COPY
030004  000770                                  .WORD   ACTNUM-10$      ;NUMERIC VALUE FOR SIZE OR COPY
030006  001062                                  .WORD   ACTQPM-10$      ;QUOTED MESSAGE FROM USER
030010  001556                                  .WORD   ACTSTS-10$      ;STATUS
030012  001102                                  .WORD   ACTEQQ-10$      ;END OF QUOTED MESSAGE FROM USER
030014  001162                                  .WORD   ACTMS0-10$      ;ONES AS DATA
030016  001170                                  .WORD   ACTMS1-10$      ;ZEROS AS DATA
030020  001200                                  .WORD   ACTMS2-10$      ;1ALT AS DATA
030022  001210                                  .WORD   ACTMS3-10$      ;0ACT AS DATA
030024  001220                                  .WORD   ACTMS4-10$      ;ITEP AS DATA
030026  001230                                  .WORD   ACTMS5-10$      ;CCITT AS DATA
030030  001246                                  .WORD   ACTMS6-10$      ;ALPHA AS DATA
030032  001334                                  .WORD   ACTATV-10$      ;ACTIVE MODE
030034  001344                                  .WORD   ACTPAS-10$      ;PASSIVE MODE
030036  001364                                  .WORD   ACTREC-10$      ;RECEIVE MODE
030040  001372                                  .WORD   ACTLIS-10$      ;LISTEN MODE
030042  001402                                  .WORD   ACTDLL-10$      ;DOWNLINE LOAD
030044  001412                                  .WORD   ACTTRA-10$      ;TRANSMIT MODE
030046  001422                                  .WORD   ACTTAL-10$      ;TALK MODE
030050  001450                                  .WORD   ACTNO-10$       ;NO   IE /NOCHECK
030052  001460                                  .WORD   ACTECH-10$      ;ECHO
030054  001564                                  .WORD   ACTCRC-10$      ;CRC
030056  001600                                  .WORD   ACTPRO-10$      ;PROTOCOL
030060  001632                                  .WORD   ACTRPS-10$      ;STATUS
030062  001642                                  .WORD   ACTMOP-10$      ;SATELLITE IN MAINTENANCE LOOP MODE
030064  001652                                  .WORD   ACTTLP-10$      ;INTERNALTTL
030066  001662                                  .WORD   ACTCLP-10$      ;CABLE LOOP
030070  001672                                  .WORD   ACTLLP-10$      ;LOCAL MODEM LOOP
030072  001702                                  .WORD   ACTRLP-10$      ;REMOTE MODEM LOOP
030074  000142                                  .WORD   ACTNUF-10$      ;MORE COMMAND NEEDED
030076  001140                                  .WORD   ACTBCR-10$      ;BAD CHARACTER IN OPERATOR MESSAGE
030100  000656                                  .WORD   ACTDMS-10$      ;DUMP MEMORY START ADDRESS
030102  000706                                  .WORD   ACTDME-10$      ;DUMP MEMORY END ADDRESS
```

```
 5049  030104  000700                .WORD   ACTDMQ-10$      ;DUMP WORD
 5050  030106  000246                .WORD   ACTPRT-10$      ;PRINT
 5051  030110  001572                .WORD   ACTMOS-10$      ;MODEM STATUS
 5052  030112  000236                .WORD   ACTEXT-10$      ;EXIT ROUTINE REV B EC
 5053  030114  001272                .WORD   ACTSEX-10$      ;SET EX=TR REV B EC
 5054
```

```
5055
5056  030116  112737  177777  003146  ACTNUF: MOVB    #-1,P$NNUF      ;SET FLAG TO SAY NEED MORE OF COMMAND
5057  030124  000207                  ACTNUL: RTS     PC              ;RETURN TO PARSER
5058
5059  030126  012737  000001  003010  ACTCLR: MOV     #CLEAR,KEYWD1   ;SET LOC TO SAY A CLEAR WAS TYPED
5060  030134  000207                          RTS     PC
5061
5062  030136  012737  000002  003010  ACTSHO: MOV     #SHOW,KEYWD1    ;SET LOC. TO SAY A SHOW WAS TYPED
5063  030144  000207                          RTS     PC
5064
5065  030146  012702  003016          ACTHLP: MOV     #HLPTAB,R2      ;SETUP R2 AS A POINTER TO HELP MSG TABLE
5066  030152                          1$:     PRINTF  #HLPF,(R2)+     ;PRINT HELP INFORMATION MESSAGES
5067  030152  012246                                                          MOV     (R2)+,-(SP)
5068  030154  012746  012273                                                  MOV     #HLPF,-(SP)
5069  030160  012746  000002                                                  MOV     #2,-(SP)
5070  030164  010600                                                          MOV     SP,R0
5071  030166  104417                                                          TRAP    C$PNTF
5072  030170  062706  000006                                                  ADD     #6,SP
5073  030174  020227  003036                  CMP     R2,#HLPEND      ;SEE IF ALL INFO PRINTED YET
5074  030200  001364                          BNE     1$              ;IF NO KEEP PRINTING
5075  030202  012737  000005  003010          MOV     #HLP,KEYWD1     ;SET LOC. TO SAY A HELP WAS TYPED
5076  030210  000207                          RTS     PC
5077
5078  030212  012737  000057  003010  ACTEXT: MOV     #EXIT,KEYWD1    ;EXIT COMMAND WAS INPUT REV B EC
5079  030220  000207                          RTS     PC              ;RETURN
5080
5081  030222  012737  000055  003010  ACTPRT: MOV     #PRNT,KEYWD1    ;SET LOC. TO SAY A HELP WAS TYPED
5082  030230  004737  020610                  JSR     PC,REPORT       ;CALL ROUTINE TO PRINT EVENT LOG AND BASE TABLE
5083  030234  000207                          RTS     PC
5084
5085  030236  012737  000004  003010  ACTRUN: MOV     #RUN,KEYWD1     ;SET RUN FLAG
5086  030244  112737  177777  003146          MOVB    #-1,P$NNUF      ;SET FLAG TO SAY NEED MORE OF COMMAND
5087  030252  012737  000001  006576          MOV     #1,RPASS        ;SET DEFAULT RUN 'PASS' TO 1
5088  030260  000207                          RTS     PC
5089
5090  030262  012737  006244  006444  ACTCSE: MOV     #PTR13,CMPPTR   ;INIT COMPARE MESSAGE POINTER
5091  030270  013701  006444                  MOV     CMPPTR,R1
5092
5093  030274  013702  006446                  MOV     CMPTOT,R2
5094  030300  105037  003146                  CLRB    P$NNUF          ;FLAG THAT HAVE VALID COMMAND AT THIS PT.
5095  030304  023727  003010  000002          CMP     KEYWD1,#SHOW    ;SEE IF A CLEAR OR SHOW WAS TYPED
5096  030312  001471                          BEQ     ACTSHW          ;BR IF A SHOW WAS TYPED
5097  030314  012737  000001  006446          MOV     #1,CMPTOT       ;CLEAR COMPARE MESSAGE COUNT, CHAR. COUNT
5098  030322  005037  006450                  CLR     CTOTCC          ; AND RESET POINTER
5099
5100  030326  012737  006244  006444          MOV     #PTR13,CMPPTR   ;INIT COMPARE MESSAGE POINTER
5101  030334  013737  006444  006524          MOV     CMPPTR,CPTR     ;SET UP TO FILL IN DEFAULT MESSAGE
5102  030342  012701  005150                  MOV     #CMPBUF,R1
5103  030346  010137  006452                  MOV     R1,CCURAD
5104  030352  000431                          BR      ACTCLB
5105
5106  030354  012701  006150          ACTCST: MOV     #PTRTAB,R1
5107  030360  013702  006462                  MOV     TXMTOT,R2
5108  030364  105037  003146                  CLRB    P$NNUF          ;FLAG THAT HAVE VALID COMMAND AT THIS PT.
5109  030370  023727  003010  000002          CMP     KEYWD1,#SHOW    ;SEE IF A CLEAR OR SHOW WAS TYPED
5110  030376  001437                          BEQ     ACTSHW          ;BR IF A SHOW WAS TYPED
```

```
5111  030400  012737  000001  006462          MOV     #1,TXMTOT       ;CLEAR TRANSMIT MESSAGE COUNT, CHAR. COUNT
5112  030406  005037  006464                  CLR     TTOTCC          ; AND RESET POINTER
5113  030412  012737  006150  006442          MOV     #PTRTAB,TXPTR
5114  030420  013737  006442  006524          MOV     TXPTR,CPTR
5115  030426  012701  003150                  MOV     #TXBUF,R1
5116  030432  010137  006466                  MOV     R1,TCURAD
5117
5118  030436  012702  001000          ACTCLB: MOV     #BUFLIM,R2
5119  030442  010137  006526                  MOV     R1,CURADD       ;SET UP TO PUT DEFAULT MSG IN LIST AFTER 033'S
5120  030446  012737  000005  006516          MOV     #5,MSGTYP
5121  030454  013737  002162  006520          MOV     MSG5C,CURCC
5122  030462  105021                  1$:     CLRB    (R1)+           ;FILL EXPT OR TRAN BUFFER WITH 0'S IF A CLEAR
5123  030464  005302                          DEC     R2              ;DO "BUFLIM" NUMBER OF BYTE LOCATIONS
5124  030466  001375                          BNE     1$
5125  030470  004737  023362                  JSR     PC,BLDBUF       ;"CLEAR" REALLY MEANS TO PUT DEFAULT MSG IN
5126  030474  000207                          RTS     PC              ;WHEN DONE, RETURN TO PARSER
5127
5128
5129  030476  012705  003072          ACTSHW: MOV     #SHTAB,R5
5130  030502  122571  000000                  5$:     CMPB    (R5)+,@(R1)     ;LOOK AT FIRST BYTE OF MSG TO DECIPHER TYPE
5131  030506  001404                          BEQ     6$
5132  030510  020527  003101                  CMP     R5,#SHTEND      ;SEE IF LOOKED AT ALL OF DEFAULTS YET
5133  030514  001372                          BNE     5$
5134  030516  005205                          INC     R5              ;MUST BE OPR. SPEC'D THEN
5135  030520  162705  003073                  6$:     SUB     #SHTAB+1,R5
5136  030524  006305                          ASL     R5
5137  030526  016137  000002  006534          MOV     2(R1),TEMP
5138  030534                                  PRINTF  #SHMSG,SHTYTB(R5),TEMP  ;PRINT MSG SIZE & TYPE
5139  030534  013746  006534                                                          MOV     TEMP,-(SP)
5140  030540  016546  003052                                                          MOV     SHTYTB(R5),-(SP)
5141  030544  012746  013507                                                          MOV     #SHMSG,-(SP)
5142  030550  012746  000003                                                          MOV     #3,-(SP)
5143  030554  010600                                                                  MOV     SP,R0
5144  030556  104417                                                                  TRAP    C$PNTF
5145  030560  062706  000010                                                          ADD     #10,SP
5146  030564  062701  000004                          ADD     #4,R1           ;BUMP R1 TO NEXT SET OF POINTERS
5147  030570  005302                          DEC     R2
5148  030572  001341                          BNE     ACTSHW
5149  030574  013737  006566  007706          MOV     MODTYP,DEV1
5150  030602  013737  006570  007710          MOV     MLTYP,DEV2
5151  030610  013737  006576  007712          MOV     RPASS,DEV3
5152  030616  013737  006574  007714          MOV     PARAM,DEV4
5153  030624  004737  023712                  JSR     PC,SHWOP        ;SHOW THE OPERATOR THE CURRENT MODE..... ALSO
5154  030630  000207                          RTS     PC
5155
5156  030632  013737  003142  006510  ACTDMS: MOV     P$NUM,STADD     ;SETUP STARTING ADDRESS FOR DUMP
5157  030640  005037  006514                  CLR     BYTBIT          ;SET DEFAULT OF WORD DUMP
5158  030644  012737  000052  003010          MOV     #DMPS,KEYWD1    ;FLAG THAT A DUMP WAS TYPED
5159  030652  000403                          BR      ACTDME
5160
5161  030654  012737  177777  006514  ACTDMQ: MOV     #-1,BYTBIT      ;SET DUMP FLAG TO "DUMP-WORD"
5162  030662  013737  003142  006512  ACTDME: MOV     P$NUM,ENADD     ;SETUP END ADDRESS FOR DUMP (=START IF NO "EEE"
5163  030670  105037  003146          ACTDMX: CLRB    P$NNUF          ;CLEAR NOT-ENOUGH FLAG, "DUMP N-N/B" IS VALID
5164  030674  000207                          RTS     PC
5165
```

```
 5166
 5167
 5168  030676  012737  000010  003010  ACTSTE: MOV     #SETEXP,KEYWD1
 5169  030704  000403                  BR      ACTSTX
 5170
 5171  030706  012737  000011  003010  ACTSTT: MOV     #SETTRN,KEYWD1
 5172  030714  012737  000001  003014  ACTSTX: MOV     #1,QUALVL       ;SET UP DEFAULT COPY TO 1 (/COPY=0)
 5173  030722  000207                  RTS     PC
 5174
 5175  030724  012737  000012  003012  ACTSZE: MOV     #SIZE,QUALFG
 5176  030732  000207                  RTS     PC
 5177
 5178  030734  012737  000013  003012  ACTCOP: MOV     #QCOPY,QUALFG
 5179  030742  000207                  RTS     PC
 5180
 5181  030744  023727  003012  000012  ACTNUM: CMP     QUALFG,#SIZE    ;SEE IF A SIZE OR COPY TYPED
 5182  030752  001023                  BNE     1$              ;BR IF IT WAS A COPY
 5183  030754  005737  003142          TST     P$NUM           ;CHECK TO BE SURE DIDN'T TRY SIZE=0
 5184  030760  001014                  BNE     3$              ; BR IF NO
 5185  030762                          PRINTF  #CLISE0
 5186  030762  012746  012074                                                  MOV     #CLISE0,-(SP)
 5187  030766  012746  000001                                                  MOV     #1,-(SP)
 5188  030772  010600                                                          MOV     SP,R0
 5189  030774  104417                                                          TRAP    C$PNTF
 5190  030776  062706  000004                                                  ADD     #4,SP
 5191  031002  112737  177777  003147          MOVB    #-1,P$GDBD      ;SEE ERROR-IN-CMD FLAG
 5192  031010  000411                  BR      2$
 5193  031012  013737  003142  006520  3$:     MOV     P$NUM,CURCC     ;IF A SIZE LOAD CURCC WITH BYTE COUNT
 5194  031020  000405                  BR      2$
 5195  031022  013737  003142  003014  1$:     MOV     P$NUM,QUALVL    ;IF A COPY, LOAD COPY COUNT
 5196  031030  005237  003014          INC     QUALVL          ;INCREMENT SO FIRST DEC MAKES IT REAL #
 5197  031034  000522                  2$:     BR      ACTMEX
 5198
 5199  031036  012737  000007  006516  ACTOPM: MOV     #7,MSGTYP
 5200  031044  010437  006534          MOV     R4,TEMP         ;KEEP TRACK OF START OF QUOTED TEXT
 5201  031050  005237  006534          INC     TEMP            ; SO CAN CALC OPCNT AT END OF QUOTES
 5202  031054  000207                  RTS     PC
 5203
 5204  031056  010402                  ACTEQO: MOV     R4,R2
 5205  031060  163702  006534          SUB     TEMP,R2
 5206  031064  010237  006520          MOV     R2,CURCC        ;CALC BYTE COUNT FOR QUOTED TEXT
 5207  031070  010237  002166          MOV     R2,OPCNT
 5208  031074  013701  006534          MOV     TEMP,R1
 5209  031100  012705  002520          MOV     #OPBUF,R5
 5210  031104  112125                  1$:     MOVB    (R1)+,(R5)+     ;COPY QUOTED TEXT TO OPBUF
 5211  031106  005302                  DEC     R2
 5212  031110  001375                  BNE     1$
 5213  031112  000473                  BR      ACTMEX
 5214
 5215  031114                          ACTBCR: PRINTF  #CLIBCR         ;BAD CHAR. IN OPR. QUOTED STRING
 5216  031114  012746  012027                                                  MOV     #CLIBCR,-(SP)
 5217  031120  012746  000001                                                  MOV     #1,-(SP)
 5218  031124  010600                                                          MOV     SP,R0
 5219  031126  104417                                                          TRAP    C$PNTF
 5220  031130  062706  000004                                                  ADD     #4,SP
 5221  031134  000207                  RTS     PC
```

```
5222
5223  031136  005037  006516          ACTMS0: CLR     MSGTYP
5224  031142  000435                          BR      ACTME1
5225  031144  012737  000001  006516  ACTMS1: MOV     #1,MSGTYP       ;SET MESSAGE TYPE = ALL ONES
5226  031152  000431                          BR      ACTME1
5227  031154  012737  000002  006516  ACTMS2: MOV     #2,MSGTYP       ;SET MESSAGE TYPE = ONES & ZEROS
5228  031162  000425                          BR      ACTME1
5229  031164  012737  000003  006516  ACTMS3: MOV     #3,MSGTYP       ;SET MESSAGE TYPE = ZEROS & ONES
5230  031172  000421                          BR      ACTME1
5231  031174  012737  000004  006516  ACTMS4: MOV     #4,MSGTYP       ;SET MESSAGE TYPE = CCITT
5232  031202  000415                          BR      ACTME1
5233  031204  012737  000005  006516  ACTMS5: MOV     #5,MSGTYP       ;SET MESS TYPE = QUICK FOX
5234  031212  013737  002162  006520          MOV     MSG5C,CURCC     ;SETUP DEFAULT SIZE FOR THIS TYPE
5235  031220  000430                          BR      ACTMEX
5236  031222  012737  000006  006516  ACTMS6: MOV     #6,MSGTYP       ;SET MESSAGE TYPE = ALPHA/NUM
5237  031230  013737  002164  006520          MOV     MSG6C,CURCC     ;SETUP DEFAULT SIZE FOR THIS TYPE
5238
5239  031236  012737  000100  006520  ACTME1: MOV     #64.,CURCC      ;SETUP DEFAULT SIZE FOR MSG0-4
5240  031244  000416                          BR      ACTMEX          ;BRANCH TO EXIT
5241
5242                                  ;REV B EC
5243  031246  022737  000010  003010  ACTSEX: CMP     #SETEXP,KEYWD1  ;DID WE GET HERE FROM ''SET E=''COMMAND?
5244  031254  001404                          BEQ     10$             ;YES,BRANCH
5245  031256  112737  000001  003147          MOVB    #1,P$GDBD       ;SET ERROR FLAG
5246  031264  000406                          BR      ACTMEX          ;GO EXIT SUBROUTINE
5247  031266  004737  023506          10$:    JSR     PC,FACSIMILE    ;GO COPY TRANSMIT LIST TO EXPECT LIST
5248  031272  012737  000060  003010          MOV     #SETET,KEYWD1   ;SET FLAG TO BE USED IN T1::
5249  031300  000400                          BR      ACTMEX          ;EXIT SUBROUTINE
5250
5251  031302  105037  003146  ACTMEX: CLRB    P$NNUF          ;CLEAR NOT-ENOUGH FLAG
5252  031306  000207                  RTS     PC
5253
```

```
 5254  031310  012737  000003  006566  ACTATV: MOV     #ACT,MODTYP             ;MODE = ACTIVE
 5255  031316  000432                  BR      ACTM2X
 5256
 5257  031320  012737  000002  006566  ACTPAS: MOV     #PAS,MODTYP             ;MODE = PASSIVE
 5258  031326  105037  003146                  CLRB    P$NNUF                  ;CLEAR NOT-ENOUGH FLAG
 5259  031332  005037  006570                  CLR     MLTYP                   ;CLEAR MAINT LOOP TYPE
 5260  031336  000207                  RTS     PC
 5261
 5262  031340  005037  006566          ACTREC: CLR     MODTYP                  ;MODE = RECEIVE
 5263  031344  000417                  BR      ACTM2X
 5264
 5265  031346  012737  000006  006566  ACTLIS: MOV     #LIS,MODTYP             ;MODE = LISTEN
 5266  031354  000413                  BR      ACTM2X
 5267
 5268  031356  012737  000004  006566  ACTDLL: MOV     #DOW,MODTYP             ;MODE = DOWNLINE LOAD
 5269  031364  000407                  BR      ACTM2X
 5270
 5271  031366  012737  000001  006566  ACTTRA: MOV     #TRA,MODTYP             ;MODE = TRANSMIT
 5272  031374  000403                  BR      ACTM2X
 5273
 5274  031376  012737  000005  006566  ACTTAL: MOV     #TAL,MODTYP             ;MODE = TALK
 5275
 5276  031404  042737  000004  006574  ACTM2X: BIC     #ECHOB,PARAM            ;DISABLE /ECHO (ALL BUT PASSIVE MODE)
 5277  031412  105037  003146                  CLRB    P$NNUF                  ;CLEAR NOT-ENOUGH FLAG
 5278  031416  005037  006570                  CLR     MLTYP                   ;CLEAR MAINT LOOP TYPE
 5279  031422  000207                  RTS     PC
 5280
```

```
 5281  031424  012737  000036  003012  ACTNO:  MOV     #NO,QUALFG
 5282  031432  000207                  RTS     PC
 5283
 5284  031434  022737  000036  003012  ACTECH: CMP     #NO,QUALFG
 5285  031442  001422                  BEQ     1$
 5286  031444  052737  000004  006574          BIS     #ECHOB,PARAM
 5287  031452  022737  000002  006566          CMP     #PAS,MODTYP             ;BE SURE IN PASSIVE MODE IF
 5288  031460  001416                  BEQ     2$                      ;IF TRYING TO SET /ECHO
 5289  031462                          PRINTF  #CLINPS
 5290  031462  012746  011764                                                  MOV     #CLINPS,-(SP)
 5291  031466  012746  000001                                                  MOV     #1,-(SP)
 5292  031472  010600                                                          MOV     SP,R0
 5293  031474  104417                                                          TRAP    C$PNTF
 5294  031476  062706  000004                                                  ADD     #4,SP
 5295  031502  112737  177777  003147          MOVB    #-1,P$GDBD
 5296  031510  042737  000004  006574  1$:     BIC     #ECHOB,PARAM
 5297  031516  005037  003012          2$:     CLR     QUALFG                  ;CLEAR 'NO' OUT OF QUALIFIER FLAG
 5298  031522  000501                  BR      ACTLXX
 5299
 5300  031524  012701  000002          ACTCHK: MOV     #DATCKB,R1              ;SET DATA CHECK BIT
 5301  031530  000413                  BR      ACTQFG
 5302
 5303  031532  012701  000001          ACTSTS: MOV     #STATB,R1       ;SET THE STATUS BIT
 5304  031536  000410                  BR      ACTQFG
 5305
 5306  031540  012701  000020          ACTCRC: MOV     #CRCB,R1        ;SET THE CRC BIT
 5307  031544  000405                  BR      ACTQFG
 5308
 5309  031546  012701  000010          ACTMOS: MOV     #MOCHK,R1       ;SET THE MODEM BIT
 5310  031552  000402                  BR      ACTQFG
 5311
 5312  031554  012701  000040          ACTPRO: MOV     #PROTOB,R1      ;SET THE PROTOCOL BIT
 5313
 5314  031560  050137  006574          ACTQFG: BIS     R1,PARAM
 5315  031564  022737  000036  003012          CMP     #NO,QUALFG
 5316  031572  001002                  BNE     1$
 5317  031574  040137  006574          BIC     R1,PARAM
 5318  031600  005037  003012          1$:     CLR     QUALFG                  ;CLEAR 'NO' OUT OF QUALIFIER FLAG
 5319  031604  000450                  BR      ACTLXX
 5320
 5321  031606  013737  003142  006576  ACTRPS: MOV     P$NUM,RPASS             ;GET NUMBER OF 'RUN PASSES'
 5322  031614  000444                  BR      ACTLXX
 5323
 5324  031616  012737  000005  006570  ACTMOP: MOV     #5,MLTYP
 5325  031624  000417                  BR      ACTLPX
 5326  031626  012737  000001  006570  ACTTLP: MOV     #1,MLTYP
 5327  031634  000413                  BR      ACTLPX
 5328  031636  012737  000002  006570  ACTCLP: MOV     #2,MLTYP
 5329  031644  000407                  BR      ACTLPX
 5330  031646  012737  000003  006570  ACTLLP: MOV     #3,MLTYP
 5331  031654  000403                  BR      ACTLPX
 5332  031656  012737  000004  006570  ACTRLP: MOV     #4,MLTYP
 5333
 5334  031664  022737  000003  006566  ACTLPX: CMP     #ACT,MODTYP     ;BE SURE IN ACTIVE IF TRYING TO SET LOOP
 5335  031672  001415                  BEQ     ACTLXX          ; BR IF IN ACTIVE
 5336  031674  112737  177777  003147          MOVB    #-1,P$GDBD
```

```
 5337  031702  005037  006570                  CLR     MLTYP           ;CLEAR ANY LOOP TYPE THAT MAY HAVE GOT SET
 5338  031706                                  PRINTF  #CLIBDL
 5339  031706  012746  011722                                                  MOV     #CLIBDL,-(SP)
 5340  031712  012746  000001                                                  MOV     #1,-(SP)
 5341  031716  010600                                                          MOV     SP,R0
 5342  031720  104417                                                          TRAP    C$PNTF
 5343  031722  062706  000004                                                  ADD     #4,SP
 5344  031726  105037  003146  ACTLXX:  CLRB    P$NNUF          ;CLEAR NOT-ENOUGH FLAG
 5345  031732  000207                          RTS     PC
 5346
```

```
5347
5348                                        ;REV B BY EC
5349  031734  005737  006570              GTR9:   TST     MLTYP           ;LOOP MODE ?
5350  031740  001422                              BEQ     10$             ;NO,BRANCH
5351  031742  032737  000002  006574              BIT     #DATCKB,PARAM   ;DATA CHECK ?
5352  031750  001416                              BEQ     10$             ;NO,BRANCH
5353  031752  023737  006446  006462              CMP     CMPTOT,TXMTOT   ;TX = EX ?
5354  031760  001412                              BEQ     10$             ;YES,BRANCH
5355  031762                                      PRINTF  #CLIPW          ;PRINT WARNING
5356  031762  012746  012125                                                        MOV     #CLIPW,-(SP)
5357  031766  012746  000001                                                        MOV     #1,-(SP)
5358  031772  010600                                                              MOV     SP,R0
5359  031774  104417                                                              TRAP    C$PNTF
5360  031776  062706  000004                                                      ADD     #4,SP
5361  032002  000137  026762                      JMP     GETCL           ;TRY AGAIN
5362
5363
5364                                        ; RX ALLOCATE CODE
5365  032006  012737  006150  006442  10$:     MOV     #PTRTAB,TXPTR   ;INIT TRANSMIT MESSAGE POINTER
5366  032014  012737  006244  006444          MOV     #PTR13,CMPPTR   ;INIT CUMPARE MESSAGE POINTER
5367  032022  012737  006340  006440          MOV     #PTR23,RXPTR    ;INIT RECEIVE MESSAGE POINTER
5368
5369  032030  013737  006446  006476          MOV     CMPTOT,RXMTOT   ;MAKE COMPARE AND RX MESSAGE COUNTS EQUAL
5370
5371
5372  032036  005037  006600            GTREX:  CLR     FLAG            ;CLEAR FLAG
5373  032042  005037  006502                    CLR     OPVAR           ;CLEAR OPTIONAL VARIABLE COUNTER
5374  032046  005037  006504                    CLR     PSCNT           ;CLEAR PASS COUNT
5375  032052  005037  006506                    CLR     ERRCNT          ;CLEAR ERROR COUNT
5376  032056  005037  011520                    CLR     MGLCNT          ;CLEAR GLITCH COUNT
5377  032062  005037  011522                    CLR     MHRCNT          ;CLEAR HARD ERR. COUNT
5378  032066  005037  006500                    CLR     LNCNT           ;CLEAR LINE COUNTER
5379  032072  012737  000626  011510            MOV     #626,SYNCW      ;SET UP SYNCW FOR 226 SYNC +TSOM
5380  032100  052737  000200  011472            BIS     #BIT7,DPVP1     ;SET UP PARAM WORD FOR 226 RX SYNC
5381  032106  005737  011524                    TST     RNODE
5382  032112  001406                            BEQ     1$              ;IF NON ITEP GO TO 1
5383  032114  042737  000200  011472            BIC     #BIT7,DPVP1     ;SET UP FOR 26 SYNC WORD ON RX.
5384  032122  012737  000426  011510            MOV     #426,SYNCW      ;ELSE SET UP SYNC FOR 26 AND TSOM
5385  032130  004737  020176              1$:   JSR     PC,LOGDVI       ;LOG ABOUT TO INIT DEVICE
5386  032134  004737  034304                    JSR     PC,DVINIT       ;INIT DEVICE
5387
5388  032140  012737  001000  006520  GTRX2:  MOV     #BUFLIM,CURCC   ;SET CHAR COUNT TO 'BUFLIM" NO. OF BYTES
5389  032146  012737  004150  006526          MOV     #RXBUF,CURADD   ;SET UP RX BUFFER AS CURRRENT ADD.
5390  032154  013737  006440  006524          MOV     RXPTR,CPTR
5391  032162  012737  000010  006516          MOV     #10,MSGTYP      ;SET UP FOR 33 TO FILL RX BUFFERS
5392  032170  004737  023362                    JSR     PC,BLDBUF       ;CLEAR RX BUFFER
5393  032174  013702  006566                    MOV     MODTYP,R2
5394  032200  006302                            ASL     R2
5395  032202  000172  006602                    JMP     @MODE(R2)       ;MODE DISPATCH
5396
```

```
5397                                    .SBTTL          RECEIVE MODE SECTION
5398                                    ;++
5399                                    ; FUNCTIONAL DESCRIPTION:
5400                                    ;       RECEIVE-ONLY (OR ONE-WAY-IN) ROUTINE
5401                                    ;       IN THIS MODE OF TESTING THE DEVICE'S RECEIVER IS ENABLED IN EXPECTATION
5402                                    ;       OF RECEIVING A MESSAGE.  AFTER RECEIVING AN "EXPECTED" NUMBER OF
5403                                    ;       MESSAGES, THE DATA RECEIVED CAN BE COMPARED AGAINST A LIST OF "EXPECT
5404                                    ;       TO RECEIVE" MESSAGES IF DATA-CHECKING IS ENABLED.
5405                                    ;
5406                                    ; SUBORDINATE ROUTINES USED:
5407                                    ;               "ALLTR"
5408                                    ;
5409                                    ; CALLING SEQUENCE:
5410                                    ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
5411                                    ;--
5412
5413   032206                           RXONLY:
5414   032206  013737  006440  006522  RXON2:  MOV     RXPTR,CPTRR
5415   032214  013737  006476  006474          MOV     RXMTOT,DVRCT    ;SET UP MESSAGE COUNT
5416   032222  052737  000104  006600          BIS     #QRX+#ERX,FLAG  ;SET UP RX QUE
5417   032230  005037  006524                  CLR     CPTR            ;CLEAR THE TX POINTER
5418   032234  000137  032376                  JMP     ALLTR           ;GO RX.
5419
```

```
                                .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
                                ;--

 5436  032240  042737  000002  006574  TXONLY: BIC     #DATCKB,PARAM   ;SET NOCHECK
 5437  032246  013737  006442  006524  TXON2:  MOV     TXPTR,CPTR
 5438  032254  013737  006462  006460          MOV     TXMTOT,DVTCT    ;COPY COUNTER FOR THIS PASS
 5439  032262  052737  000210  006600          BIS     #QTX+#ETX,FLAG  ;SET THE QUE TX FLAG
 5440  032270  005037  006522                  CLR     CPTRR           ;CLEAR RX POINTER
 5441  032274  000137  032376                  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
                                ;--

032300                          PLCK:
032300  013737  006462  006460 PLCK2:  MOV     TXMTOT,DVTCT    ;SET UP  THE TRANSMIT COUNT
032306  013737  006442  006524         MOV     TXPTR,CPTR      ;SET UP CPTR TO TRANSMIT POINTER
032314  013737  006440  006522 PLCK3:  MOV     RXPTR,CPTRR     ;SET UP CPTRR TO REC POINTER
032322  052737  000104  006600         BIS     #QRX+#ERX,FLAG  ;SET UP Q AND EXPECT RX
032330  000137  032376                 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
                                ;--

032334  013737  006462  006460  ALCK:   MOV     TXMTOT,DVTCT
032342  013737  006442  006524          MOV     TXPTR,CPTR      ;SET UP TX COUNTS
032350  013737  006476  006474          MOV     RXMTOT,DVRCT    ;SET UP COUNTS
032356  013737  006440  006522          MOV     RXPTR,CPTRR
032364  052737  000314  006600          BIS     #QRX+#QTX+#ETX+#ERX,FLAG
032372  000137  032376                  JMP     ALLTR
```

```
5495                                    .SBTTL          TRANSMIT - RECEIVE FOR ALL STANDARD MODES
5496                                    
5497                                    ;++
5498                                    ; FUNCTIONAL DESCRIPTION:
5499                                    ;       THIS CODE PERFORMS THE FOLLOWING FUNCTIONS
5500                                    ;       1.) IF RX BUFFERS ARE TO BE QUED,TELL DEVICE
5501                                    ;           CODE TO QUE THEM ,LOG RECEIVE QUED.
5502                                    ;       2.) IF TX BUFFERS ARE TO BE QUED ,TELL DEVICE
5503                                    ;           CODE TO QUE THEM, LOG TRANSMIT QUED.
5504                                    ;       3.) WAIT FOR EITHER RECIVE BUFFER OR TRANSMIT BUFFER OR
5505                                    ;           BOTH TO COMPLETE
5506                                    ;       4.) IF RECEIVE COMPLETE LOG IT UPDATE RX TABLE IF DATA
5507                                    ;           CHECKING.
5508                                    ;       5.) IF TRANSMIT COMPLETE LOG IT.
5509                                    ;       6.) WHEN BOTH TRANSMIT AND RECIEVE LISTS ARE DONE
5510                                    ;           GO TO THE COMPARE BUFFER CODE
5511                                    ;
5512                                    ; SUBORDINATE ROUTINES USED:
5513                                    ;               'DVRXQ'' -QUE RECEIVE BUFFER SPACE TO DEVICE
5514                                    ;               'LOGRXQ''-LOG RECEIVE BUFFER SPACE TO EVENT LOG
5515                                    ;               'LOGTXQ''-LOG TRANSMIT BUFFER QUED TO EVENT LOG
5516                                    ;               'DVTXRX''-QUE TRANSMIT BUFFER AND WAIT FOR RX
5517                                    ;                       OR TX TO COMPLETE
5518                                    ;               'LOGRXC''-LOG RECEIVE BUFFER COMPLETED TO EVENT LOG
5519                                    ;               'LOGTXC''-LOG TRANSMIT BUFFER COMPLETED TO EVENT LOG
5520                                    ;
5521                                    ; USE OF FLAG BITS:
5522                                    ;               QRX - SET ON INPUT TO ALLTR IF REC IS TO BE QUED TO
5523                                    ;                     DEVICE. CLEARED BY DVRXQ AND THEN SET BY DVTXRX
5524                                    ;                     WHEN RX BUFFER IS COMPLETED.
5525                                    ;               QTX - SET ON INPUT TO ALLTR IF TRANSMIT IS TO BE QUED TO
5526                                    ;                     DEVICE. CLEARED ON ENTRY TO DVTXRX AND SET BY DVTXRX
5527                                    ;                     WHEN TX BUFFER IS COMPLETED.
5528                                    ;               ETX - USED BY DVTXRX TO DETERMINE IF TX BUFFER COMPLETED IS
5529                                    ;                     EXPECTED.
5530                                    ;               ERX - USED BY DVTXRX TO DETERMINE IF RX BUFFER COMPLETED IS
5531                                    ;                     EXPECTED.
5532                                    ;
5533                                    ;
5534                                    ; CALLING SEQUENCE:
5535                                    ;       JMP     ALLTR           ;GO TO TRANSMIT-RECEIVE FOR ALL STANDARD MODES
5536                                    ;--
5537                                    
5538                                    
5539  032376                            ALLTR:
5540  032376  032737  000004  006600    ALCK5:  BIT     #QRX,FLAG
5541  032404  001424                            BEQ     ALCK1           ;IF NOT RX GO TO TX'S
5542  032406  013702  006522                    MOV     CPTRR,R2
5543  032412  011237  006540                    MOV     (R2),TEMP2
5544  032416  012237  006470                    MOV     (R2)+,DVRXA
5545  032422  011237  006542                    MOV     (R2),TEMP3
5546  032426  011237  006472                    MOV     (R2),DVRCC
5547  032432  010237  006522                    MOV     R2,CPTRR
5548  032436  004737  020132                    JSR     PC,LOGRXQ       ;LOG REC QUED
5549  032442  032737  000040  006574    10$:    BIT     #PROTOB,PARAM   ;'/PROTOCOL/ ?
5550  032450  001002                            BNE     ALCK1           ;YES,BRANCH
```

```
5551  032452  004737  035166                  JSR     PC,DVRXQ        ;GO QUE RX BUFFERS & ENABLE RX
5552  032456  032737  000010  006600  ALCK1:  BIT     #QTX,FLAG
5553  032464  001416                          BEQ     ALCK2           ;IF NO TX'S GO TO 2
5554  032466  013702  006524                  MOV     CPTR,R2
5555  032472  011237  006540                  MOV     (R2),TEMP2
5556  032476  012237  006454                  MOV     (R2)+,DVTXA
5557  032502  011237  006542                  MOV     (R2),TEMP3
5558  032506  012237  006456                  MOV     (R2)+,DVTCC
5559  032512  010237  006524                  MOV     R2,CPTR
5560  032516  004737  020076                  JSR     PC,LOGTXQ
5561
5562  032522  032737  000040  006574  ALCK2:  BIT     #PROTOB,PARAM   ;'/PROTOCOL'?
5563  032530  001410                          BEQ     10$             ;NO,BRANCH
5564  032532  004737  040702                  CALL    PROTOCOL        ;GO DO DDCMP MESSAGE PROCESSING
5565  032536  032737  000200  006574          BIT     #ABORT,PARAM    ;PROTOCOL ABORT?
5566  032544  001404                          BEQ     20$             ;NO,BRANCH
5567  032546  000137  026642                  JMP     GTRA5           ;ABORT!!AND RETURN TO 'DCLT>' PROMPT
5568
5569  032552  004737  035270          10$:    JSR     PC,DVTXRX       ;GO TO TX AND RX SUB ROUT.
5570
5571  032556  032737  000004  006600  20$:    BIT     #QRX,FLAG       ;CHECK FOR REC. MSG.
5572  032564  001514                          BEQ     ALCK3
5573  032566  013737  006470  006540          MOV     DVRXA,TEMP2
5574  032574  013737  006472  006542          MOV     DVRCC,TEMP3
5575  032602  004737  020150                  JSR     PC,LOGRXC       ;LOG REC COMPLETE
5576  032606  032737  000004  006574  UPTABL: BIT     #ECHOB,PARAM    ;IS THIS ECHO MODE(PASSIVE)
5577  032614  001406                          BEQ     UPTA4           ;IF NOT GO TO 4
5578  032616  013702  006524                  MOV     CPTR,R2         ;ELSE SET R2 TO PRESENT TX fABL
5579  032622  013722  006540                  MOV     TEMP2,(R2)+     ;STORE OFF RX ADD
5580  032626  013712  006542                  MOV     TEMP3,(R2)      ;AND CC
5581  032632  032737  000002  006574  UPTA4:  BIT     #DATCKB,PARAM   ;IS DATA CHECKING ASKED FOR
5582  032640  001015                          BNE     UPTA1           ;IF SO GO TO 1
5583  032642  012737  000001  006474          MOV     #01,DVRCT       ;ELSE SET DVRCT TO A 1
5584  032650  013737  006440  006522          MOV     RXPTR,CPTRR     ;RESET POINTER
5585  032656  022737  000003  006566          CMP     #ACT,MODTYP     ;IS THIS ACTIVE
5586  032664  001002                          BNE     UPTA3
5587  032666  005237  006474                  INC     DVRCT           ;IF YES BUMP COUNT
5588  032672  000424                  UPTA3:  BR      UPTEX
5589  032674  013702  006522          UPTA1:  MOV     CPTRR,R2
5590  032700  011237  006534                  MOV     (R2),TEMP       ;LOAD TEMP WITH PREV. COUNT
5591  032704  163737  006542  006534          SUB     TEMP3,TEMP      ;LOAD TEMP WITH PREV.COUNT-CURRENT
5592  032712  013722  006542                  MOV     TEMP3,(R2)+
5593  032716  063737  006542  006540          ADD     TEMP3,TEMP2
5594  032724  013722  006540                  MOV     TEMP2,(R2)+     ;STORE OF NEW ADD
5595  032730  013712  006534                  MOV     TEMP,(R2)       ;AND NEW CC
5596  032734  162702  000002                  SUB     #2,R2           ;PUT POINTER BACK TO ADDR.
5597
5598  032740  010237  006522                  MOV     R2,CPTRR        ;AND RESTORE IT.
5599
5600  032744                          UPTEX:
5601  032744  022737  000002  006566          CMP     #PAS,MODTYP
5602  032752  001007                          BNE     ALCK2A          ;IF NOT PASSIVE LOOP THEN GO TO 2A
5603  032754  042737  000104  006600          BIC     #QRX+#ERX,FLAG  ;CLEAR BOTH EXPECTED AND COMPLETED FLAGS
5604  032762  052737  000210  006600          BIS     #QTX+#ETX,FLAG  ;SET THE TX FLAGS
5605  032770  000632                          BR      ALCK1
5606
```

```
 5607  032772  005337  006474          ALCK2A: DEC     DVRCT           ;DEC REC COUNT
 5608  032776  005737  006474                  TST     DVRCT           ;IS IT ALL DONE
 5609  033002  001005                          BNE     ALCK3           ;NO. GO CHECK TX
 5610  033004  042737  000004  006600          BIC     #QRX,FLAG       ;CLEAR THE RX FLAG
 5611  033012  005037  006522                  CLR     CPTRR           ;YES. CLEAR POINTER
 5612  033016  032737  000010  006600  ALCK3:  BIT     #QTX,FLAG       ;IS IT TX
 5613  033024  001447                          BEQ     ALCK4           ;IF NOT TX THEN GO BACK
 5614  033026  013737  006454  006540          MOV     DVTXA,TEMP2
 5615  033034  013737  006456  006542          MOV     DVTCC,TEMP3     ;LOG TX COMPLETED
 5616  033042  004737  020114                  JSR     PC,LOGTXC
 5617  033046  005337  006460                  DEC     DVTCT           ;DEC TX COUNT
 5618  033052  022737  000002  006566          CMP     #PAS,MODTYP
 5619  033060  001013                          BNE     ALCK3A          ;IF NOT PASSIVE MODE GO TO 3A
 5620  033062  042737  000210  006600          BIC     #QTX+ETX,FLAG   ;CLEAR THE TX FLAGS
 5621  033070  052737  000104  006600          BIS     #QRX+ERX,FLAG   ;AND SET THE RX FLAGS
 5622  033076  005737  006460                  TST     DVTCT
 5623  033102  001005                          BNE     ALCK3C          ;IF MORE RX'S DO IT
 5624  033104  000137  033164                  JMP     CMPSR           ; ELSE COMPARE
 5625  033110  005737  006460  ALCK3A: TST     DVTCT           ;IS IT ALL DONE
 5626  033114  001402                          BEQ     ALCK3B          ;IF NOT GO BACK TO 5
 5627  033116  000137  032376  ALCK3C: JMP     ALCK5
 5628  033122  005037  006524  ALCK3B: CLR     CPTR            ;IF SO CLEAR POINTER
 5629  033126  042737  000010  006600          BIC     #QTX,FLAG       ;CLEAR TX FLAG
 5630  033134  032737  000002  006574          BIT     #DATCKB,PARAM   ;IS IT DAT CK
 5631  033142  001403                          BEQ     ALCK4A          ;IF NOT THEN END WO CKING RX.
 5632  033144  005737  006522  ALCK4:  TST     CPTRR
 5633
 5634  033150  001362                          BNE     ALCK3C          ;IF SOME RX''S LEFT GO BACK
 5635  033152  005737  006524  ALCK4A: TST     CPTR
 5636  033156  001402                          BEQ     ALCK4B          ;BRANCH IF ANY TX'S LEFT
 5637  033160  000137  032522                  JMP     ALCK2
 5638  033164          ALCK4B:
 5639
 5640
 5641
```

```
                                .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
                                ;--
                                ;
033164  032737  000002  006574  CMPSR:  BIT     #DATCKB,PARAM   ;IS DATA CHECKING TO BE DONE
033172  001522                          BEQ     CMPSEX          ;IF NOT THEN EXIT
033174  013737  006440  006524          MOV     RXPTR,CPTR      ;PUT START OF RX POINTERS TO CPTR
033202  013737  006444  006522          MOV     CMPPTR,CPTRR    ; AND START OF COMPARE POINTS TO CPTRR
033210  013737  006476  006474          MOV     RXMTOT,DVRCT
                                ;
033216                          CMPS3:
033216  013702  006524                  MOV     CPTR,R2         ;MOVE CURRET RX PT.TO R2
033222  011237  006540                  MOV     (R2),TEMP2      ;MOVE RX ADD TO EVENT LOG
033226  012201                          MOV     (R2)+,R1        ;SET R1 TO START ADD OF RX
033230  012237  006542                  MOV     (R2)+,TEMP3     ;SET CHAR  COUNT TO EVENT LOG
033234  010237  006524                  MOV     R2,CPTR         ;RESTORE RX POINT
                                ;
033240  013702  006522                  MOV     CPTRR,R2        ;PUT R2 AT COMPARE TABLE
033244  012203                          MOV     (R2)+,R3        ;SET R3 TO COMPARE ADD
033246  012204                          MOV     (R2)+,R4        ;SET R4 TO COMP CC
033250  010237  006522                  MOV     R2,CPTRR        ;RESTORE POINTER
033254  010437  006544                  MOV     R4,TEMP4
033260  004737  020244                  JSR     PC,LOGCMP       ;LOG COMPARE START.
                                ;
033264  020437  006542                  CMP     R4,TEMP3        ;IS COMPARE COUNT = TO RX COUNT
033270  001410                          BEQ     CMPS7           ;IF SO GO TO 7
033272  005237  006506                  INC     ERRCNT
033276                                  ERRSOFT 1,EDDLE,ERR10   ;PRINT ERROR
```

```
                                                                                TRAP    C$ERSOFT
                                                                                .WORD   1
                                                                                .WORD   EDDLE
                                                                                .WORD   ERR10
 033276  104457
 033300  000001
 033302  015322
 033304  017552
 033306  004737  020262                JSR     PC,LOGCML       ;LOG LENGTH ERROR
 033312  005037  006544        CMPS7:  CLR     TEMP4           ;CLEAR BAD BYTE COUNTER
 033316  012737  000001 006532        MOV     #1,OFSET        ;SET OFSET BYTE COUNT TO 1
 033324  122123                CMPS1:  CMPB    (R1)+,(R3)+     ;COMPARE RX WITH EXPETED
 033326  001422                        BEQ     CMPS6           ;IF EQUAL THEN GO TO 6
 033330  005237  006544        CMPS2:  INC     TEMP4           ;INC BAD COUNT
 033334  023727  006544 000005        CMP     TEMP4,#5        ;IS IT MORE THEN 5
 033342  101014                        BHI     CMPS6           ;IF SO GO  FOR MORE
 033344  114337  006554                MOVB    -(R3),GOOD      ;STORE GOOD BYTE FOR ERROR
 033350  114137  006555                MOVB    -(R1),BAD       ;STORE BAD BYTE FOR ERROR
 033354  005237  006506                INC     ERRCNT
 033360                                ERRSOFT 2,EDDDE,ERR1    ;REPORT COMPARISON FAILURE TO OPR.
                                                                                TRAP    C$ERSOFT
                                                                                .WORD   2
                                                                                .WORD   EDDDE
                                                                                .WORD   ERR1
 033360  104457
 033362  000002
 033364  015357
 033366  017462
 033370  005201                        INC     R1
 033372  005203                        INC     R3
 033374  005237  006532        CMPS6:  INC     OFSET           ;INC OFFSET
 033400  005304                        DEC     R4              ;ELSE DEC CHAR COUNT AND SEE IF 0
 033402  001350                        BNE     CMPS1           ;IF NOT GO BACK
 033404  005737  006544                TST     TEMP4           ;SEE IF ANY CMP ERRS FOR THIS MSG
 033410  001410                        BEQ     CMPS5A          ;BR IF NONE
 033412  005237  006506                INC     ERRCNT
 033416                                ERRSOFT 3,EDDDE,ERR2    ;REPORT # OF MISMATCHES FOR MESSAGE
                                                                                TRAP    C$ERSOFT
                                                                                .WORD   3
                                                                                .WORD   EDDDE
                                                                                .WORD   ERR2
 033416  104457
 033420  000003
 033422  015357
 033424  017524
 033426  004737  020300        CMPS5:  JSR     PC,LOGCMD       ;LOG DATA ERROR IN COMPARE
 033432                        CMPS5A:
 033432  005337  006474                DEC     DVRCT
 033436  001267                        BNE     CMPS3           ;IF NOT ALL DONE GO BACK
```

```
                                .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
                                ;
                                ;;
                                ;;--

033440  005737  011520  CMPSEX: TST     MGLCNT          ;CHECK FOR ANY GLITCH ERRORS
033444  001003                   BNE     MCREP           ;IF NON ZERO REPORT THEM
033446  005737  011522           TST     MHRCNT          ;CHECK FOR ANY HARD ERRORS
033452  001412                   BEQ     ENDPS           ;IF NONE GO TO END OF PASS
                                 ;
                                 ;REPORT ANY MODEM ERRORS HERE
                                 ;
033454  005237  006506  MCREP:   INC     ERRCNT          ;BUMP ERROR COUNT
033460                           ERRSOF  4,MSCMS,ERR4
                                                                 TRAP    C$ERSOFT
033460  104457                                                   .WORD   4
033462  000004                                                   .WORD   MSCMS
033464  014756                                                   .WORD   ERR4
033466  017602
033470  005037  011520           CLR     MGLCNT          ;CLEAR GLITCH COUNT
033474  005037  011522           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
                                ;------

033500  005237  006504          ENDPS:  INC     PSCNT           ;BUMP PASS COUNT

033504  013737  006502  006544          MOV     OPVAR,TEMP4
033512  013737  006504  006540          MOV     PSCNT,TEMP2
033520  013737  006506  006542          MOV     ERRCNT,TEMP3
033526  004737  020316                  JSR     PC,LOGEOP       ;LOG END OF PASS

033532  022737  177777  006576          CMP     #-1,RPASS       ;SEE IF RPASS=-1
033540  001403                          BEQ     1$              ;IF IT IS DON'T DECRMNT, LOOP FOREVER
033542  005337  006576                  DEC     RPASS           ;DEC PASS COUNT
033546  001402                          BEQ     2$              ;IF DONE EXIT TEST
033550  000137  032140          1$:     JMP     GTRX2           ;ELSE GO BACK AND DISPATCH
033554  042777  000120  155670  2$:     BIC     #RINTEN!RXENA,@RXCSR    ;TURN OFF RX
033562  000137  026642                  JMP     GTRA5           ;WHEN RPASS=0 GO BACK TO "DCLT>"
```

```
                        .SBTTL          DOWN-LINE-LOAD SECTION
                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       DOWN LINE LOAD IS NOT SUPPORTED BY THIS DEVICE..
                        ;       IF THIS MODE IS CALLED BY THE COMMAND LINE INTERPRETER
                        ;       THEN A MESSAGE WILL BE PRINTED ....THAT SAYS DOWN LINE
                        ;       LOAD IS NOT!! SUPPORTED BY THIS DEVICE.
                        ;--


033566                  DLL:
                                PRINTF  #DLLCM
033566                                                          MOV     #DLLCM,-(SP)
033566  012746  014122                                         MOV     #1,-(SP)
033572  012746  000001                                         MOV     SP,R0
033576  010600                                                 TRAP    C$PNTF
033600  104417                                                 ADD     #4,SP
033602  062706  000004
033606  000137  026642          JMP     GTRA5
```

```
                        .SBTTL          TALK MODE SECTION

                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       TALK MODE SECTION
                        ;       IN THIS MODE, THE "TALK" END OF THE LINK TRANSMITS OPERATOR
                        ;       SPECIFIED MESSAGES UNTIL A "EXIT" MESSAGE IS TYPE.  AT THAT POINT,
                        ;       THIS END OF THE LINK GOES INTO "LISTEN" MODE.
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;
                        ;               "LOGTXQ" - LOG TX BUFFER QUED TO EVENT LOG
                        ;               "DVTXRX" - QUE TX BUFFER TO DEVICE AND WAIT FOR COMPLETE
                        ;               "LOGTXC" - LOG TX COMPLETE TO EVENT LOG
                        ;
                        ; CALLING SEQUENCE:
                        ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
                        ;--

033612                  TALCK:
033612  042737  000002  006574          BIC     #DATCKB,PARAM   ;SET NOCHECK
033620  012702  002520                  MOV     #OPBUF,R2
033624  012722  177777  1$:             MOV     #-1,(R2)+       ;CLEAR OUT OPBUFFER FIRST
033630  022702  002642                  CMP     #OPEND,R2
033634  001373                          BNE     1$
033636                                  GMANID  OPRMM,OPBUF,A,-1,1,72.,NO       ;GET TALK MESSAGE
033636  104443                                                                          TRAP    C$GMAN
033640  000406                                                                          BR      10001$
033642  002520                                                                          .WORD   OPBUF
033644  000142                                                                          .WORD   T$CODE
033646  014056                                                                          .WORD   OPRMM
033650  177777                                                                          .WORD   -1
033652  000001                                                                          .WORD   T$LOLIM
033654  000110                                                                          .WORD   T$HILIM
033656                                                                          10001$:
033656  005002                          CLR     R2              ;NOW GET CHAR COUNT
033660  122762  000377  002520  2$:     CMPB    #377,OPBUF(R2)
033666  001402                          BEQ     3$
033670  005202                          INC     R2
033672  000772                          BR      2$
033674  010237  002166          3$:     MOV     R2,OPCNT

033700  012737  002520  006454          MOV     #OPBUF,DVTXA    ;SET UP TX ADDR.
033706  012737  002520  006540          MOV     #OPBUF,TEMP2
033714  013737  002166  006542          MOV     OPCNT,TEMP3
033722  013737  002166  006456          MOV     OPCNT,DVTCC     ;SET UP TX CC
033730  004737  020076                  JSR     PC,LOGTXQ
033734  052737  000210  006600          BIS     #QTX+#ETX,FLAG  ;SET UP FLAGS
033742  005037  006522                  CLR     CPTRR           ;CLEAR RX POINTER

                        ;;THIS CODE ADDED FOR PROTOCOL
033746  032737  000040  006574          BIT     #PROTOB,PARAM   ;'/PROTOCOL' ?
033754  001406                          BEQ     20$             ;NO,BRANCH
033756  042737  000004  006600          BIC     #RXQ,FLAG       ;CLEAR RX BIT
033764  004737  040702                  CALL    PROTOCOL        ;DO DDCMP TRANSMIT
033770  000402                          BR      25$             ;BRANCH
```

```
 5887
 5888  033772  004737  035270          20$:    JSR     PC,DVTXRX
 5889  033776  013737  006454  006540  25$:    MOV     DVTXA,TEMP2
 5890  034004  013737  006456  006542          MOV     DVTCC,TEMP3
 5891  034012  004737  020114                  JSR     PC,LOGTXC
 5892  034016  022737  054105  002520          CMP     #''EX,OPBUF     ;CHECK FOR EXIT
 5893  034024  001272                          BNE     TALCK
 5894  034026  022737  052111  002522          CMP     #''IT,OPBUF+2
 5895  034034  001266                          BNE     TALCK
 5896  034036  042737  000210  006600          BIC     #QTX+#ETX,FLAG  ;CLEAR THE TX BITS
 5897  034044  012737  000006  006566          MOV     #LIS,MODTYP     ;CHANGE TO LISTEN MODE
 5898  034052  000137  032140                  JMP     GTRX2           ;AND GO BACK TO DISPATCH
```

```
 5899                                   .SBTTL          LISTEN MODE SECTION
 5900
 5901                                   ;++
 5902                                   ; FUNCTIONAL DESCRIPTION:
 5903                                   ;       LISTEN MODE SECTION
 5904                                   ;       IN THIS MODE, THE "LISTEN" END OF THE LINK PRINTS ALL OF THE MESSAGES
 5905                                   ;       RECEIVED BY THE DEVICE ON THE OPERATOR'S CONSOLE.  IF THE MESSAGE
 5906                                   ;       RECEIVED IS AN "EXIT" MESSAGE, THEN THE NODE ENTERS "TALK" MODE.
 5907                                   ;
 5908                                   ; SUBORDINATE ROUTINES USED:
 5909                                   ;
 5910                                   ;               "DVRXQ" - QUE RECEIVE BUFFER SPACE TO DEVICE
 5911                                   ;              "LOGRXQ" - LOG RECEIVE BUFFER QUED TO EVENT LOG
 5912                                   ;              "DVTXRX" - WAIT FOR RX TO COMPLETE
 5913                                   ;              "LOGRXC" - LOG RX COMPLETE TO EVENT LOG
 5914                                   ;
 5915                                   ; CALLING SEQUENCE:
 5916                                   ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
 5917                                   ;--
 5918
 5919  034056  042737  000002  006574  LISCK:  BIC     #DATCKB,PARAM   ;CLEAR CHECK BIT
 5920  034064                          PRINTF  #LISP           ;PRINT PROMPT FOR OPR.
 5921  034064  012746  014045                                                 MOV     #LISP,-(SP)
 5922  034070  012746  000001                                                 MOV     #1,-(SP)
 5923  034074  010600                                                         MOV     SP,R0
 5924  034076  104417                                                         TRAP    C$PNTF
 5925  034100  062706  000004                                                 ADD     #4,SP
 5926  034104  012737  002520  006470  LISCKA: MOV     #OPBUF,DVRXA    ;SET DEVICE UP TO REC AT OPBUF
 5927  034112  012737  002520  006540          MOV     #OPBUF,TEMP2
 5928  034120  012737  000122  006472          MOV     #82.,DVRCC      ;SET UP CHAR COUNT TO 82.
 5929  034126  012737  000122  006542          MOV     #82.,TEMP3
 5930  034134  052737  000104  006600          BIS     #QRX+#ERX,FLAG  ;SET UP FLAG
 5931  034142  005037  006524                  CLR     CPTR            ;CLEAR THE TX.
 5932
 5933                          ;; WAS PROTOCOL SELECTED ?
 5934  034146  032737  000040  006574          BIT     #PROTOB,PARAM   ;'/PROTOCOL' ?
 5935  034154  001007                          BNE     20$             ;YES,BRANCH
 5936
 5937  034156  004737  035166                  JSR     PC,DVRXQ        ;QUE RX
 5938  034162  004737  020132                  JSR     PC,LOGRXQ
 5939  034166  004737  035270                  JSR     PC,DVTXRX       ;GO TO DEVICE RX. SUBROUTINE
 5940  034172  000402                          BR      25$             ;SKIP PROTOCOL
 5941  034174  004737  040702          20$:    CALL    PROTOCOL        ;GO DDCMP PROTOCOL
 5942  034200  013737  006470  006540  25$:    MOV     DVRXA,TEMP2
 5943  034206  013737  006472  006542          MOV     DVRCC,TEMP3     ;SET UP ADDR.AND CC.
 5944  034214  004737  020150                  JSR     PC,LOGRXC       ;LOG COMPLETED
 5945  034220  063737  006470  006472          ADD     DVRXA,DVRCC
 5946  034226  105077  152240                  CLRB    @DVRCC
 5947  034232                          PRINTF  #OPBFPT
 5948  034232  012746  002514                                                 MOV     #OPBFPT,-(SP)
 5949  034236  012746  000001                                                 MOV     #1,-(SP)
 5950  034242  010600                                                         MOV     SP,R0
 5951  034244  104417                                                         TRAP    C$PNTF
 5952  034246  062706  000004                                                 ADD     #4,SP
 5953  034252  022737  054105  002520          CMP     #'EX,OPBUF      ;COMPARE FOR EX OF "EXIT"
 5954  034260  001311                          BNE     LISCKA          ;IF NOT EXIT THEN GO BACK
```

```
 5955  034262  022737  052111  002522          CMP     #''IT,OPBUF+2   ;IF FIRST HALF OK CHECK NEXT PART
 5956  034270  001305                          BNE     LISCKA          ;IF NOT EXIT THE GO BACK
 5957  034272  012737  000005  006566          MOV     #TAL,MODTYP     ;CHANGE MODE TO TALK
 5958  034300  000137  032140                  JMP     GTRX2           ;RETURN TO DISPATCHER
 5959
 5960
```

```
                        .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
                        ;--
034304                  DVINIT:
                                ;MASTER CLEAR DEVICE
034304  012777  000001  155146          MOV     #RESET,@TXCSR   ;DO A MASTER CLEAR
034312  105777  155134                  TSTB    @RXCSR          ;SEE IF IT WORKED
034316  001423                          BEQ     DVIN1           ;BRANCH IF OK
034320                                  BREAK
034320  104422                                                                  TRAP    C$BRK
                                ;REPORT ERROR IF RESET
                                ;DOES NOT WORK
034322  012737  016747  006540          MOV     #DVEM0,TEMP2
034330  017737  155116  006542          MOV     @RXCSR,TEMP3
034336  017737  155116  006544          MOV     @TXCSR,TEMP4    ;LOAD UP ERRM. AND REG OUTPUTS
034344  004737  020160                  JSR     PC,LGDVE        ;LOG TIME OUT WAITING FOR RUN
034350  005237  006506                  INC     ERRCNT
034354                                  ERRSOFT 5,DVEM0,ERR13
034354  104457                                                                  TRAP    C$ERSOFT
034356  000005                                                                  .WORD   5
034360  016747                                                                  .WORD   DVEM0
```

```
034362 017634                                                              .WORD   ERR13
034364 000747                       BR DVINIT                  ;GO BACK AND TRY MSTR CLR AGAIN IF ERROR

                                    ;SET TTL LOOP IF REQU'D

034366 042737 000003 006600 DVIN1:  BIC     #3,FLAG            ;CLEAR INPUT AND OUTPUT INT FLAGS
034374 042777 000010 155056         BIC     #TTLL,@TXCSR       ;CLEAR INTERNAL LOOP
034402 022737 000001 006570         CMP     #TTL,MLTYP         ;IS TTL SELECTED
034410 001004                       BNE     DVIN3              ; IF NOT GO TO 3
034412 052777 000010 155040         BIS     #TTLL,@TXCSR       ;ELSE SET INTERNAL LOOP
034420 000455                       BR      DVIN37

034422 022737 000002 006570 DVIN3:  CMP     #CABLE,MLTYP
034430 001451                       BEQ     DVIN37             ;IF CABLE LOOP SKIP CHECK
                                                               ;FOR MODEM READY

034432 022737 000004 006566         CMP     #DOW,MODTYP        ;CHECK IF DLL
034440 001002                       BNE     DVIN3A             ;BRANCH IF NOT DLL
034442 000137 035102                JMP     DVINEX             ;ELSE EXIT

034446 012777 000002 154776 DVIN3A: MOV     #DTR,@RXCSR        ;SET UP DTR.

034454 012737 002000 006640         MOV     #2000,TIMER1
034462 005737 006640        DVIN38: TST     TIMER1
034466 001022                       BNE     DVIN39             ;IF TIMER NOT OUT GO TO 39

                                    ;SET ERROR FOR NO MODEM READY

034470 012737 017375 006540         MOV     #DVEM6,TEMP2
034476 017737 154750 006542         MOV     @RXCSR,TEMP3
034504 017737 154750 006544         MOV     @TXCSR,TEMP4
034512 004737 020160                JSR     PC,LGDVE
034516 005237 006506                INC     ERRCNT
034522                              ERRSOFT 11,DVEM6,ERR13
034522 104457                                                              TRAP    C$ERSOFT
034524 000013                                                              .WORD   11
034526 017375                                                              .WORD   DVEM6
034530 017634                                                              .WORD   ERR13
034532 000745                       BR      DVIN3A             ;THEN TRY TO SET DTR AGAIN
034534                      DVIN39: BREAK
034534 104422                                                              TRAP    C$BRK
034536 017737 154710 011476         MOV     @RXCSR,IRXCSR      ;GET COPY OR RXCSR
034544 032737 001000 011476         BIT     #BIT9,IRXCSR       ;IS MODEM READY SET
034552 001743                       BEQ     DVIN38
034554 013777 011472 154672 DVIN37: MOV     DPVP1,@PCSAR       ;SET UP PCASAR
034562 005737 011524                TST     RNODE              ;CHECK REMOTE NODE
034566 001145                       BNE     DVINEX             ;EXIT IF ITEP
034570 005737 006572                TST     FHDPLX             ;IS THIS FULL DUPLEX
034574 001542                       BEQ     DVINEX             ;BANCH IF NOT
034576 032737 000040 006574         BIT     #PROTOB,PARAM      ;'/PROTOCOL' ?
034604 001136                       BNE     DVINEX             ;YES,EXIT

                            ;;THIS START-STACK ROUTINE USED IN NON PROTOCOL,NON ITEP,FULL DUPLEX MODE
                                    ;SET UP TO SEND STRT
034606 112737 000005 002645         MOVB    #5,HDMSG+1         ;SET UP ENQ
034614 052737 000060 006600         BIS     #RXM!TXM,FLAG      ;SET FLAG WORD
```

```
 6073  034622  012737  000074  006644          MOV     #60.,TIMERS     ;SET TIMER FOR 1 MINUTE
 6074  034630  004737  036574                  JSR     PC,CTSSR        ;SET CTS IF NESC.
 6075  034634  012737  000006  002646  DVIN41: MOV     #6,HDMCC        ;SET UP STRT CODE
 6076  034642  004737  036444                  JSR     PC,DVIN31       ;GO TX STRT AND CHK FOR RX.
 6077  034646  005737  006644                  TST     TIMERS
 6078  034652  001466                          BEQ     DVIN81          ;IF TIMER EXPIERED EXIT
 6079
 6080  034654  022737  000006  002660  DVIN4:  CMP     #6,RHDMCC       ;IS THE RCVD=STRT
 6081  034662  001441                          BEQ     DVIN8           ;IF SO GO TO ASTRT
 6082  034664  022737  000007  002660          CMP     #7,RHDMCC       ;IS IT A STACK
 6083  034672  001360                          BNE     DVIN41          ;IF NOT STACK ETIHER GO BACK
 6084
 6085  034674  004737  036574          DVIN9:  JSR     PC,CTSSR        ;SET REQUEST TO SEND
 6086  034700  042737  001010  006600          BIC     #QTX!PAD,FLAG   ;CLEAR TX COMPT FLAG.
 6087  034706  012737  000001  002646          MOV     #1,HDMCC        ;SET UP ACK
 6088  034714  012737  002645  011502          MOV     #HDMSG+1,MSGPTR ;SET UP POINTER
 6089  034722  013737  002654  011504          MOV     HDMC,MSGCC
 6090  034730  012737  000010  011506          MOV     #8.,SYNCC       ;SET UP SYNC COUNT
 6091  034736  052777  000120  154514          BIS     #TXENA!TINTEN,@TXCSR
 6092  034744  032737  000010  006600  DIVN91: BIT     #QTX,FLAG
 6093  034752  001053                          BNE     DVINEX          ;EXIT IF ACK SENT
 6094  034754                                  BREAK
 6095  034754  104422                                                                  TRAP    C$BRK
 6096  034756  005737  006644                  TST     TIMERS
 6097  034762  001370                          BNE     DIVN91          ;IF NOT TIMER EXPIRED RECHK TX.
 6098  034764  000421                          BR      DVIN81          ;IF TIMER OUT REPORT IT
 6099
 6100  034766  012737  000007  002646  DVIN8:  MOV     #7,HDMCC        ;SET POTINTER TO STACK
 6101  034774  004737  036444                  JSR     PC,DVIN31               ;AND GO SEND STACK
 6102  035000  005737  006644                  TST     TIMERS
 6103  035004  001411                          BEQ     DVIN81          ;REPORT ERROR IF TIME OUT
 6104  035006  022737  000001  002660          CMP     #1,RHDMCC       ;IS IT ACK RCVD?
 6105  035014  001432                          BEQ     DVINEX          ;IF SO EXIT
 6106  035016  022737  000007  002660          CMP     #7,RHDMCC       ;IS IT STACK RCVD
 6107  035024  001723                          BEQ     DVIN9           ;IF SO SEND ACK
 6108  035026  000757                          BR      DVIN8           ;IF NEITHER SEND ANOTHER ACK
 6109
 6110                                          ;DO ERROR AND REPEAT
 6111
 6112  035030  012737  017311  006540  DVIN81: MOV     #DVEM5,TEMP2
 6113  035036  013737  002660  006542          MOV     RHDMCC,TEMP3
 6114  035044  013737  002646  006544          MOV     HDMCC,TEMP4     ;LOAD UP ERRM. AND REG OUTPUTS
 6115  035052  004737  020160                  JSR     PC,LGDVE        ;LOG TIME OUT WAITING FOR RUN
 6116  035056  005237  006506                  INC     ERRCNT
 6117  035062                                  ERRSOFT 10.,DVEM5,ERR13
 6118  035062  104457                                                          TRAP    C$ERSOFT
 6119  035064  000012                                                          .WORD   10
 6120  035066  017311                                                          .WORD   DVEM5
 6121  035070  017634                                                          .WORD   ERR13
 6122  035072  005237  006502                  INC     OPVAR           ;COUNT HOW MANY TIMES WE DO THIS.
 6123  035076  000137  034304                  JMP     DVINIT          ;TRY ALL OVER AGAIN
 6124
 6125  035102  004737  037006          DVINEX: JSR     PC,CLRRTS       ;CLEAR RTS IF NESC
 6126  035106  042737  173777  006600          BIC     #173777,FLAG    ;CLEAR FLAG WORD
 6127  035114  052737  002000  006600          BIS     #INOVR,FLAG     ;SET THE INITT OVER FLAG
 6128  035122  000207                          RTS     PC              ;RETURN TO CALLER
```

```
 6129
 6130
```

```
                                .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
                                ;--

035124  017737  154322  007554  DVMODS: MOV     @RXCSR,MODS
035132  042737  000040  007554          BIC     #BIT5,MODS      ;CLEAR BIT 5
035140  032777  000040  154312          BIT     #BIT5,@TXCSR    ;SEE IT TM OR SQ SET
035146  001403                          BEQ     DVMEX           ;IF NOT EXIT
035150  052737  000040  007554          BIS     #BIT5,MODS      ;IF SET SET BIT 5 IN MODS
035156  042737  106720  007554  DVMEX:  BIC     #106720,MODS    ;CLEAR ALL UNUSED BITS
035164  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
                                ;--

035166                          DVRXQ:
035166  032737  000004  006600          BIT     #QRX,FLAG
035174  001434                         BEQ     DVREX                   ;IF NOT RX THEN EXIT
                                                                        ;ELSE QUE RX
035176  042737  000444  006600          BIC     #QRX+#BCC+#RXM,FLAG     ;CLEAR FLAG FOR RX
035204  005737  011524                 TST     RNODE                   ;IF NON ITEP GO TO 2
035210  001415                         BEQ     DVRX2
035212  052737  000440  006600          BIS     #RXM+#BCC,FLAG          ;GET JUST THE DATA NO CRC.
035220  013737  006470  011512          MOV     DVRXA,RMSGPT
035226  012737  000072  011514          MOV     #72,RMSGCC              ;SET UP RX TO GET ITEP MSG.
035234  012737  000070  006472          MOV     #70,DVRCC
035242  000406                         BR      DVRX3

                                ;ENABLE RX, RX INTERRUPTS,AND DATA SET INTERRUPTS

035244  012737  002657  011512  DVRX2:  MOV     #RHDMSG+1,RMSGPT        ;SET UP POINTER
035252  013737  002654  011514          MOV     HDMC,RMSGCC             ;AND CC
035260  052777  000160  154164  DVRX3:  BIS     #RINTEN!RXENA!#DSITEN,@RXCSR
035266  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
                        ;--

035270  032737  000010  006600  DVTXRX: BIT     #QTX,FLAG               ;ANY TX TO QUE
035276  001444                        BEQ     DVTR3                   ;IF NOT GO WAIT FOR OUPUT
035300  042737  001030  006600        BIC     #QTX+#TXM+PAD,FLAG      ;CLEAR FLAG
035306  004737  036574                JSR     PC,CTSSR                ;GO SET CTS
035312  005737  011524                TST     RNODE
035316  001412                        BEQ     DVTR1                   ;IF NON-ITEP GO TO 1
035320  052737  000020  006600        BIS     #TXM,FLAG               ;SET THE BODY BIT
035326  013737  006454  011502        MOV     DVTXA,MSGPTR
035334  013737  006456  011504        MOV     DVTCC,MSGCC             ;AND SET UP FOR ACTUAL DATA
035342  000414                        BR      DVTR2
                                        ;ENABLE TX AND TX INTER.

035344  112737  000201  002645  DVTR1:  MOVB    #201,HDMSG+1            ;SET UP SOH
035352  012737  002645  011502        MOV     #HDMSG+1,MSGPTR         ;SET POINTER TO HEADER
035360  013737  006456  002646        MOV     DVTCC,HDMCC
035366  013737  002654  011504        MOV     HDMC,MSGCC              ;SET CC FOR HEADER
035374  012737  000177  011506  DVTR2:  MOV     #177,SYNCC              ;SET UP FOR 177 SYNCS.
035402  052777  000120  154050        BIS     #TXENA!#TINTEN,@TXCSR
```

```
 6256  035410  012737  000074  006644  DVTR3:  MOV     #60.,TIMERS             ;SET TIMER FOR 60 SECS
 6257
 6258  035416                          DVTR8:  BREAK                                                   TRAP    C$BRK
 6259  035416  104422
 6260  035420  005737  006644                  TST     TIMERS                  ;IS TIMER EXPIRED
 6261  035424  001022                          BNE     TOINOT
 6262
 6263                                          ;LOG ERROR TIME OUT RX OR TX NOT COMPLETED
 6264
 6265  035426  012737  017110  006540          MOV     #DVEM2,TEMP2
 6266  035434  017737  154012  006542          MOV     @RXCSR,TEMP3
 6267  035442  017737  154012  006544          MOV     @TXCSR,TEMP4
 6268  035450  004737  020160                  JSR     PC,LGDVE
 6269  035454  005237  006506                  INC     ERRCNT
 6270  035460                                  ERRSOFT 7,DVEM2,ERR13
 6271  035460  104457                                                                  TRAP    C$ERSOFT
 6272  035462  000007                                                                  .WORD   7
 6273  035464  017110                                                                  .WORD   DVEM2
 6274  035466  017634                                                                  .WORD   ERR13
 6275  035470  000747                          BR      DVTR3           ;RETURN TO CHECK TIMER
 6276
 6277  035472  032737  000010  006600  TOINOT: BIT     #QTX,FLAG       ;IS IT TX COMPL?
 6278  035500  001406                          BEQ     DVTR4           ;BRANCH IF TX NOT DONE.
 6279  035502  004737  037006                  JSR     PC,CLRRTS
 6280  035506  032737  000100  006600          BIT     #ERX,FLAG       ;ARE WE EXPECTING TO RX
 6281  035514  001416                          BEQ     DVTREX          ;BRANCH IF NOT.
 6282
 6283  035516  032737  000004  006600  DVTR4:  BIT     #QRX,FLAG       ;IS RX DONE
 6284  035524  001734                          BEQ     DVTR8           ;GO BACK AND TIME IF NOT
 6285
 6286  035526  032737  000200  006600          BIT     #ETX,FLAG       ;ARE WE EXPECTG TO TX.
 6287  035534  001406                          BEQ     DVTREX          ;BRANCH IF NOT.
 6288
 6289  035536  032737  000010  006600          BIT     #QTX,FLAG       ;IS IT TX COMPLETED
 6290  035544  001724                          BEQ     DVTR8           ;GO BACK AND TIME OUT
 6291  035546  004737  037006                  JSR     PC,CLRRTS       ;CLEAR RTS IF NESC.
 6292  035552  000207          DVTREX: RTS     PC                      ;AND EXIT
 6293
```

```
                                ; 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
                                ;--
                                ;
035554                          BGNSRV  DVRXI
035554                                                                  DVRXI::
035554  010246                  MOV     R2,-(SP)        ;SAVE R2
035556  017737  153670  011476  MOV     @RXCSR,IRXCSR   ;MOV RX CSR TO IMAGE
035564  032737  000010  006574  BIT     #MOCHK,PARAM    ;ANY MODEM CHANGES TO REPORT
035572  001456                  BEQ     RXIN21          ;IF NOT IGNORE DS CHANGE.
035574  032737  002000  006600  BIT     #INOVR,FLAG     ;IS INIT OVER
035602  001452                  BEQ     RXIN21          ;NO THEN IGNORE DS CHANGE.
035604  005737  011476          TST     IRXCSR
035610  100047                  BPL     RXIN21          ;IF DATA SET CHANGE IS NOT SET BR
035612  013737  011476  011474  MOV     IRXCSR,CMODS    ;MOV THE NEW MODEM STATUS IN
035620  042737  106760  011474  BIC     #106760,CMODS
```

```
6350  035626  032777  000040  153624          BIT     #TM,@TXCSR
6351  035634  001403                          BEQ     RXIN2           ;IF TEST MODE SET
6352  035636  052737  000040  011474          BIS     #TM,CMODS       ;SET IT IN NEW STATUS
6353  035644  013737  011474  006542  RXIN2:  MOV     CMODS,TEMP3
6354  035652  013737  007554  006544          MOV     MODS,TEMP4
6355  035660  023737  006544  006542          CMP     TEMP4,TEMP3     ;COMPARE OLD TO CURRENT
6356  035666  001406                          BEQ     GLINC           ;INC GLITCH COUNT
6357  035670  005237  011522                  INC     MHRCNT          ;INC HARD COUNT
6358  035674  012737  016717  006540          MOV     #HRDMSG,TEMP2   ;SET UP HARD MESG.
6359  035702  000405                          BR      RXIN1
6360  035704  005237  011520          GLINC:  INC     MGLCNT          ;INC GLITCH COUNT
6361  035710  012737  016671  006540          MOV     #GLMSG,TEMP2    ;SET UP GLITCH
6362  035716  004737  020334          RXIN1:  JSR     PC,LOGMSC       ;GO LOG MODEM STATUS CHANGE
6363  035722  013737  011474  007554          MOV     CMODS,MODS      ;MOVE CURRENT TO OLD
6364                                          ;
6365                                          ;TEST FOR STATUS OR DATA
6366
6367  035730  032737  002200  011476  RXIN21: BIT     #RSTARY!RDATRY,IRXCSR
6368  035736  001544                          BEQ     RXINEX          ;IF NEITHER EXIT
6369  035740  017737  153512  011500          MOV     @RDSR,IRDSR
6370  035746  032737  000200  011476          BIT     #RDATRY,IRXCSR  ;IS THIS DATA
6371  035754  001455                          BEQ     RXIN3           ;IF NOT GO TO 3
6372
6373                                          ;GET HERE WITH GOOD DATA
6374
6375  035756  013702  011512          RXIN4:  MOV     RMSGPT,R2
6376  035762  113722  011500                  MOVB    IRDSR,(R2)+     ;STORE DATA AWAY
6377  035766  010237  011512                  MOV     R2,RMSGPT       ;PUT POINTER BACK
6378
6379
6380  035772  005337  011514                  DEC     RMSGCC
6381  035776  001124                          BNE     RXINEX          ;GET OUT IF NOT ALL DONE
6382  036000  032737  000400  006600          BIT     #BCC,FLAG       ;IS THE BCC FLAG ALREADY SE
6383  036006  001066                          BNE     RXIN5           ;BRANCH IF YES.
6384  036010  032737  100000  011500          BIT     #RERR,IRDSR     ;IS THE ERR CHK BIT SET INDICATING
6385                                                                  ;GOOD BCC.
6386  036016  001022                          BNE     RXIN6           ;BRANCH IF GOO
6387  036020  013737  011500  006542          MOV     IRDSR,TEMP3
6388  036026  013737  011476  006544          MOV     IRXCSR,TEMP4
6389  036034  012737  017205  006540          MOV     #DVEM3,TEMP2
6390  036042  004737  020160                  JSR     PC,LGDVE
6391  036046  005237  006506                  INC     ERRCNT
6392  036052                                  ERRSOFT 8,DVEM3,ERR13
6393  036052  104457                                                  TRAP    C$ERSOFT
6394  036054  000010                                                  .WORD   8
6395  036056  017205                                                  .WORD   DVEM3
6396  036060  017634                                                  .WORD   ERR13
6397
6398  036062  000467                          BR      RXIN8           ;DISABLE INTERRUPTS AND EXIT
6399
6400  036064  052737  000400  006600  RXIN6:  BIS     #BCC,FLAG       ;SET FLAG
6401  036072  012737  000002  011514          MOV     #2.,RMSGCC      ;SET THE COUNT TO 2
6402  036100  012737  011516  011512          MOV     #BCCW,RMSGPT    ;SET POINTER TO BCC WORD
6403  036106  000460                          BR      RXINEX
6404
6405                                          ;STATUS CHECK
```

```
 6406
 6407  036110  032737  002000  011476  RXIN3:  BIT     #RSTARY,IRXCSR  ;IS THIS A STATUS INT.
 6408  036116  001454                  BEQ     RXINEX          ;EXIT IF NOT
 6409
 6410                                  ;LOG OVERRUN ERROR
 6411
 6412  036120  012737  017245  006540          MOV     #DVEM4,TEMP2
 6413  036126  013737  011500  006542          MOV     IRDSR,TEMP3
 6414  036134  013737  011476  006544          MOV     IRXCSR,TEMP4
 6415  036142  004737  020160                  JSR     PC,LGDVE
 6416  036146  005237  006506                  INC     ERRCNT
 6417  036152                                  ERRSOFT 9,DVEM4,ERR13
 6418  036152  104457                                                          TRAP    C$ERSOFT
 6419  036154  000011                                                          .WORD   9
 6420  036156  017245                                                          .WORD   DVEM4
 6421  036160  017634                                                          .WORD   ERR13
 6422  036162  000424                  BR      RXIN7
 6423
 6424  036164  032737  000040  006600  RXIN5:  BIT     #RXM,FLAG       ;IS THE RX M BODY BIT SET
 6425  036172  001020                  BNE     RXIN7           ;IF YES THEN ALL DONE
 6426  036174  052737  000040  006600          BIS     #RXM,FLAG
 6427  036202  042737  000400  006600          BIC     #BCC,FLAG       ;CLEAR BCC AND SET RXM
 6428  036210  013737  006470  011512          MOV     DVRXA,RMSGPT    ;MOVE ADDRESS TO POINTER
 6429  036216  013737  002660  011514          MOV     RHDMCC,RMSGCC   ;MOVE THE CHAR COUNT IN
 6430  036224  013737  002660  006472          MOV     RHDMCC,DVRCC    ;SET THE CC TO AMOUNT IN HEADER
 6431  036232  000406                  BR      RXINEX          ;AND FINISH.
 6432
 6433  036234  052737  000004  006600  RXIN7:  BIS     #QRX,FLAG       ;SET FLAG BIT
 6434
 6435  036242  042777  000120  153202  RXIN8:  BIC     #RINTEN+RXENA,@RXCSR    ;CLEAR INTAND RX ENABLE
 6436
 6437  036250  012602          RXINEX: MOV     (SP)+,R2                ;RESTORE R2
 6438  036252                  ENDSRV
 6439  036252                                                                  L10020:
 6440  036252  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.
                        ;--

                                BGNSRV  DVTXI
                                                                                DVTXI::
                                MOV     R2,-(SP)        ;SAVE R2
                                TST     SYNCC           ;ANY SYNCS TO SEND
                                BEQ     TXIN1           ;IF NOT GO TO 1
                                MOV     SYNCW,@TDSR     ;ELSE SET TSOM AND SYNC WORD
                                DEC     SYNCC           ;DEC SYNC COUNT
                                BNE     TXINEX          ;IF NOT ZERO EXIT
TXIN1:                          BIC     #TEOM!TSOM,@TDSR
                                BIT     #PAD,FLAG       ;IS THE PAD BIT SET
                                BEQ     TXIN2           ;GO TO 2 IF NOT SET
                                MOV     #377,@TDSR      ;LOAD FF TO TX DATA REG.
                                BIC     #TINTEN,@TXCSR  ;CLEAR TX INT ENABLE
                                BIS     #QTX,FLAG       ;SET THE TX COMPLETE IN FLAG
                                INC     TXREADY         ;TELL PROTOCOL MODULE THAT WE ARE DONE
                                BR      TXINEX          ;AND EXIT
TXIN2:                          MOV     MSGPTR,R2       ;LOAD R2 WITH TX ADDR.
                                MOVB    (R2)+,@TDSR     ;LOAD DATA BYTE
                                MOV     R2,MSGPTR       ;RESTORE POINTER
                                DEC     MSGCC           ;DEC CC
                                BNE     TXINEX
```

```
 6497  036370  052777  001000  153064          BIS     #TEOM,@TDSR
 6498  036376  032737  000020  006600          BIT     #TXM,FLAG       ;IS THIS THE END OF DATA MSG.
 6499  036404  001012                          BNE     TXIN3           ;IF SO SET THE PAD BIT
 6500  036406  052737  000020  006600          BIS     #TXM,FLAG       ;IF NOT MUST BE END OF HEADER
 6501  036414  013737  006454  011502          MOV     DVTXA,MSGPTR    ;SO SET UP MSGPTR FOR MSG
 6502  036422  013737  006456  011504          MOV     DVTCC,MSGCC     ;AND THE CC FOR MSG.
 6503  036430  000403                          BR      TXINEX
 6504  036432  052737  001000  006600  TXIN3:  BIS     #PAD,FLAG       ;SET THE PAD BIT
 6505
 6506  036440  012602                  TXINEX: MOV     (SP)+,R2        ;RESTORE R2
 6507  036442                                  ENDSRV
 6508  036442                                                                  L10021:
 6509  036442  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.
                                ;
                                ;--

036444  042737  000004  006600  DVIN31: BIC     #QRX,FLAG               ;CLEAR RX COMPLE.

036452  012737  002657  011512          MOV     #RHDMSG+1,RMSGPT        ;SET UP POINTER
036460  013737  002654  011514          MOV     HDMC,RMSGCC             ;AND CC
                                                                        ;ENABLE RCVR.

036466  052777  000160  152756          BIS     #RINTEN!RXENA!DSITEN,@RXCSR

                                        ;SET UP TXMITTER TO SEND

036474  004737  036574          DVIN32: JSR     PC,CTSSR                ;SET RTS .
036500  042737  001010  006600          BIC     #QTX!PAD,FLAG           ;CLEAR TX COMPT FLAG.
036506  012737  002645  011502          MOV     #HDMSG+1,MSGPTR         ;MOVE THE CURRENT POINTER TO MSGPTR.
036514  013737  002654  011504          MOV     HDMC,MSGCC
036522  012737  000010  011506          MOV     #8.,SYNCC               ;SET UP SYNC COUNT
036530  052777  000120  152722          BIS     #TXENA!TINTEN,@TXCSR

                                        ;NOW WAIT FOR TIME OUT OR TX COMPLETE

036536                          DVIN35: BREAK
036536  104422                                                          TRAP    C$BRK
036540  005737  006644                  TST     TIMERS          ;IS IT TIMED OUT
036544  001412                  BEQ     DVIN34          ;IF YES EXIT
036546  032737  000010  006600          BIT     #QTX,FLAG       ;IS TX DONE
036554  001770                  BEQ     DVIN35          ;IF NOT GO BACK AND CK TIME OUT
036556  004737  037006                  JSR     PC,CLRRTS       ;GO CLEAR RTS IF NESC.
036562  032737  000004  006600          BIT     #QRX,FLAG       ;DID WE RX ANYTHING
```

```
 6566  036570  001741                    BEQ     DVIN32          ;IF NOT RETRANSMIT LAST
 6567  036572  000207          DVIN34:   RTS     PC              ;RETURN TO CALLER
 6568
```

```
                                .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
                                ;--

036574  022737  000001  006570  CTSSR:  CMP     #1,MLTYP        ;IS THIS TTL LOOP
036602  001500                          BEQ     DVTXR9          ;BR IF YES
                                ;SET RTS AND WAIT FOR CTS
036604  032737  004000  006600  DVTXR3: BIT     #FIRST,FLAG
036612  001014                          BNE     CTSS3           ;IF NOT FIRST TIME SKIP DELY
036614  012737  177777  006534          MOV     #-1,TEMP
036622  005237  006534          CTSS4:  INC     TEMP
036626                          BREAK                                   TRAP    C$BRK
036626  104422
036630  005737  006534                  TST     TEMP
036634  001372                          BNE     CTSS4           ;IF NOT ZERO GO BACK
036636  052737  004000  006600          BIS     #FIRST,FLAG     ;SET FIRST FLAG.
036644  012737  001750  006640  CTSS3:  MOV     #1000.,TIMER1   ;1000 TICKS
036652  005737  006572                  TST     FHDPLX          ;FULL DUPLEX ?
036656  001012                          BNE     CTSS7           ;YES,BRANCH
                                ;;CHECK FOR CARRIER
036660  004737  035124          10$:    CALL    DVMODS          ;GET MODEM STATUS
036664  032737  010000  007554          BIT     #DCD,MODS       ;CARRIER PRESENT?
036672  001404                          BEQ     CTSS7           ;NO,BRANCH
036674  005737  006640                  TST     TIMER1          ;TIME DONE?
036700  001401                          BEQ     CTSS7           ;YES,BRANCH
036702  000766                          BR      10$             ;DO IT AGAIN

036704  052777  000004  152540  CTSS7:  BIS     #RTS,@RXCSR     ;SET REQUEST TO SEND
036712  012737  001750  006640          MOV     #1000.,TIMER1   ;SET UP TIMER
036720                          DVTXR2: BREAK                                   TRAP    C$BRK
036720  104422
036722  032777  020000  152522          BIT     #CTS,@RXCSR     ;IS CLEAR TO SEND BACK
036730  001025                          BNE     DVTXR1          ;BR. IF CTS IS SET
036732  005737  006640                  TST     TIMER1          ;ELSE TEST IF TIME EXPIRED
036736  001370                          BNE     DVTXR2          ;BR IF TIME NOT EXPRIED.

                                ;SET ERROR FOR NO CTS
```

```
 6625  036740  012737  017026  006540          MOV     #DVEM1,TEMP2
 6626  036746  017737  152500  006542          MOV     @RXCSR,TEMP3
 6627  036754  017737  152500  006544          MOV     @TXCSR,TEMP4
 6628  036762  004737  020160                  JSR     PC,LGDVE
 6629  036766  005237  006506                  INC     ERRCNT
 6630  036772                                  ERRSOFT 6,DVEM1,ERR13
 6631  036772  104457                                                  TRAP    C$ERSOFT
 6632  036774  000006                                                  .WORD   6
 6633  036776  017026                                                  .WORD   DVEM1
 6634  037000  017634                                                  .WORD   ERR13
 6635  037002  000700                          BR      DVTXR3          ;THEN TRY TO SET RTS AGAIN
 6636  037004                          DVTXR1:
 6637  037004  000207                  DVTXR9: RTS     PC              ;
```

```
 CVCLHC DPV-11 DATA COMM. LINK TEST     MACY11 30A(1052)  23-MAR-82  16:43  PAGE 166
CVCLHC.P11    22-MAR-82 11:09                      DEVICE CLEAR REQUEST TO SEND

  6638                                  .SBTTL                    DEVICE CLEAR REQUEST TO SEND
  6639                                  ;++
  6640                                  ; FUNCTIONAL DESCRIPTION:
  6641                                  ;       THIS ROUTINE CLEARS REQUEST TO SEND IF
  6642                                  ;       IN HALF DUPLEX MODE OR MULTI-POINT.
  6643                                  ; CALLING SEQUENCE:
  6644                                  ;       JSR     PC,CLRRTS
  6645                                  ;--
  6646
  6647   037006                         CLRRTS:
  6648   037006  005737  037126                 TST     MPPTP           ;MULTI-POINT?
  6649   037012  001003                         BNE     20$             ;YES,BRANCH
  6650   037014  005737  006572                 TST     FHDPLX          ;IS THIS FULL DUPLEX
  6651   037020  001011                         BNE     DVTR5           ;BRANCH IF YES
  6652   037022  012737  002000  006540 20$:    MOV     #2000,TEMP2
  6653   037030  005337  006540 CLRR1:  DEC     TEMP2
  6654   037034  001375                         BNE     CLRR1           ;DELAY A WHILE TO ALLOW CRC
  6655                                                                  ;TO BE TXMITT'D BEFORE YOU
  6656   037036  042777  000004  152406         BIC     #RTS,@RXCSR     ;CLEAR REQUEST TO SEND
  6657   037044  000207                 DVTR5:  RTS     PC              ;RETURN TO CALLER
  6658
```

```
                                        .SBTTL  DDCMP PROTOCOL MODULE
                                ;*****************************************************************************
                                ;
                                ;       DCLT DDCMP PROTOCOL MODULE:
                                ;       THIS CODE WAS WRITTEN TO BE USED ONLY WITH DCLT.
                                ;
                                        .EVEN
                                ;; LOCAL STORAGE
                                ;; TABLE OF STATISTICS AND ERRORS
                                ;;; NOTE: KEEP THE VARIABLES TOGETHER AND IN SEQUENCE
                                ;;;       OTHERWISE THE RPT> ROUTINE WILL PRINT WRONG INFO.
037046  000000                  PRSTAT: .WORD 0                 ;STATUS FLAGS
                                                                ;BIT0 = BCCOK
                                                                ;BIT1 = BCCBAD
                                                                ;BIT2 = SNAK
                                                                ;BIT3 = SACK
                                                                ;BIT4 = SDATA
                                                                ;SPARE
                                                                ;BIT6 = RXD
                                                                ;BIT7 = SPARE
                                                                ;BIT8 = NAKRX
                                                                ;BIT9 = MYDATA
                                                                ;BIT10 = SSTACK
                                                                ;BIT11 =SSTART
037050  000000                  TMESTX: .WORD 0                 ;TOTAL DATA MESSAGES TRANSMITTED(16 BIT COUNTER
037052  000000                  TMESRX: .WORD 0                 ;TOTAL DATA MESSAGES RECEIVED(16 BIT COUNTER)

037054      000                 N:      .BYTE 0                 ;# OF HIGHEST SEQUENTIAL DATA MESSAGE TRANS
                                                                ;; MITTED BY THIS STATION
037055      000                 A:      .BYTE 0                 ;# OF THE HIGHEST SEQUENTIAL DATA MESSAGE
                                                                ;; THAT HAS BEEN ACKNOWLEDGE TO THIS STATION

037056      000                 T:      .BYTE 0                 ;# OF THE NEXT DATA MESSAGE TO BE TRANSMITTED
037057      000                 X:      .BYTE 0                 ;LAST MESSAGE NUMBER TRANSMITTED

037060      000                 R:      .BYTE 0                 ;LAST MESSAGE RECEIVED
037061      000                 TRIBN:  .BYTE 0                 ;TRIB ADDRESS  PT TO PT = 1

                                ;; ERROR COUNTERS
037062      000                 REMTMO: .BYTE 0                 ;REMOTE REPLY TIMEOUTS(ACKS SENT NUM=R)
037063      000                 GLOBCC: .BYTE 0                 ;GLOBAL CRC ERRORS

037064      000                 REANAK: .BYTE 0                 ;REASON FOR LAST NAK SENT
037065      000                 SELTHER:.BYTE 0                 ;SELECTION THRESHOLD ERROR

037066      000                 RXTHER: .BYTE 0                 ;RECEIVE THRESHOLD ERRORS
037067      000                 TXTHER: .BYTE 0                 ;TRANSMIT THRESHOLD ERRORS
```

```
6709
6710  037070  000          DEROUT: .BYTE 0          ;DATA ERRORS OUTBOUND (NAKS RECEIVED
6711  037071  000          OUTMASK:.BYTE 0          ;; REASONS = 1,2,OR 3)
6712                                                ;MASK VALUES -- BIT0 = HEADER CRC ERROR
6713                                                ;             -- BIT1 = DATA FIELD CRC ERROR
6714                                                ;             -- BIT2 = REP RESPONSE NUM<>R
6715
6716  037072  000          DERIN:  .BYTE 0          ;DATA ERRORS INBOUND (NAKS TRANSMITTED
6717  037073  000          INMASK: .BYTE 0          ;; REASONS = 1,2,OR 3)
6718                                                ;MASK VALUES -- BIT0 = HEADER CRC ERROR
6719                                                ;             -- BIT1 = DATA FIELD CRC ERROR
6720                                                ;             -- BIT2 = REP RESPONSE NUM<>R
6721
6722  037074  000          LBUFER: .BYTE 0          ;LOCAL BUFFER ERRORS (NAKS SENT
6723  037075  000          LBMASK: .BYTE 0          ;; REASONS = 8. OR 16.)
6724                                                ;MASK VALUES -- BIT0 = BUFFER NOT AVAILABLE
6725                                                ;             -- BIT1 = MESSAGE TOO LONG
6726
6727  037076  000          RBUFER: .BYTE 0          ;REMOTE BUFFER ERRORS (NAKS RECEIVED
6728  037077  000          RBMASK: .BYTE 0          ;REASONS 8. OR 16.)
6729                                                ;MASK VALUES -- BIT0 = BUFFER NOT AVAILBLE
6730                                                ;             -- BIT1 = MESSAGE TOO LONG
6731
6732  037100  000          RMSTER: .BYTE 0          ;REMOTE STATION ERRORS (NAKS RECEIVED
6733  037101  000          RMMASK: .BYTE 0          ;REASON 9. OR 17.)
6734                                                ;MASK VALUES-- BIT0 = RECEIVER OVERRUN
6735                                                ;              BIT1 = FORMAT ERROR
6736
6737  037102  000          LOSTER: .BYTE 0          ;LOCAL STATION ERRORS (NAKS SENT
6738  037103  000          LSMASK: .BYTE 0          ;; REASON 9. OR 17.)
6739                                                ;MASK VALUES -- BIT0 = RECEIVER OVERRUN
6740                                                ;             -- BIT1 = FORMAT ERROR
6741
6742  037104  000000       RXTXTE: .WORD 0          ;RX AND TX THRESHOLD ERRORS (OVERFLOWS)
6743  037106  000          SPARE0: .BYTE 0
6744  037107  000          SPARE1: .BYTE 0
6745  037110  000000       PROEND: .WORD 0          ;END OF PROTOCOL COUNTERS
6746  037112  000000       IMFLAG: .WORD 0          ;IMAGE OF MAIN CODE FLAG WORD
6747  037114  000000       RXPRC:  .WORD 0          ;-1 = MESSAGE RX'ED & 'ACK' SENT
6748  037116  000000       TXPRC:  .WORD 0          ;-1 = MESSAGE TX'ED & 'ACK' RECEIVED
6749  037120  000000       ASTRT:  .WORD 0          ;-1 = STACK SENT
6750  037122  000000       TXREADY:.WORD 0          ; 1 = READY TO SEND ANOTHER MESSAGE
6751  037124  000000       PRUN:   .WORD 0          ; 1 = PROTOCOL RUNNING. USED IN THIS MODULE
6752  037126  000000       MPPTP:  .WORD 0          ; 1 = MULTI POINT NETWORK
6753  037130  000000       SELECT: .WORD 0          ; 1 =THIS STATION CAN NOW TRANSMIT(HALF/DUPLEX)
6754  037132  000000       IMPRSTAT:.WORD 0         ; COPY OF PROTOCOL STATUS WORD
6755  037134  000000       PRFLAG: .WORD 0          ; USED TO COMMUNICATE WITH RX INTER. ROUTINE
6756  037136  000000       HDXMTP: .WORD 0          ; 1 = HALF DUPLEX OR MULTI-POINT
6757  037140  000000       PRTEMP: .WORD 0          ; TEMPORARY WORK LOCATION
6758  037142  000000       TURNON: .WORD 0          ; 1 = RECEIVER IS ALREADY ON
6759  037144  000000       TIMEOUT: .WORD 0         ; 20 = PRINT 'TX OR RX NOT COMPLETE'
6760
```

```
6761
6762
6763                            ;; NAK REASONS VALUES AS USED IN NAK CONTROL MESSAGES
6764
6765    000001                      HEADBCC = 1             ;HEADER BCC ERROR
6766    000002                      DATABCC = 2             ;DATA BCC ERROR
6767    000003                      REPSENT = 3             ;REP RESPONSE
6768    000010                      BUFFNA  = 10            ;BUFFER TEMPORARILY NOT AVAILABLE
6769    000011                      RXOVRUN = 11            ;RECEIVER OVERRUN
6770    000020                      MESLONG = 20            ;MESSAGE TOO LONG
6771    000021                      FORMERR = 21            ;HEADER FORMAT ERROR
6772
6773
6774                            ;; ADDITIONAL NAK BIT MASKS AS USED IN COUNTERS
6775
6776    000004                      REPMSK = BIT2           ;REPLY RESPONSE
6777    000001                      RXOVMSK= BIT0           ;RECEIVER OVERRUN
6778    000002                      FMTMSK = BIT1           ;FORMAT ERROR
6779    000002                      MTLMSK = BIT1           ;MESSAGE TOO LONG
6780    000001                      BNAMSK = BIT0           ;BUFFER NOT AVAILABLE
6781
6782                            ;; MESSAGE TYPE DEFINITIONS
6783
6784    000201                      SOH = 201               ;DATA MESSAGE
6785    000144                      MAINT = 144             ;MAINTENANCE MESSAGE
6786    000005                      ENQ = 5                 ;CONTROL MESSAGE
6787
6788
6789                            ;;SUBTYPES OF CONTROL MESSAGES
6790
6791    000001                      ACK = 1                 ;ACKNOWLEDGE MESSAGE
6792    000002                      NAK = 2                 ;NEGATIVE ACKNOWLEDGE MESSAGE
6793    000003                      REP = 3                 ;REPLY TO MESSAGE NUMBER
6794    000006                      STRT = 6                ;START MESSAGE
6795    000007                      STACK = 7               ;START ACKNOWLEDGE MESSAGE
6796
6797
6798                            ;;STATUS WORD BIT DEFINITIONS
6799
6800    000001                      BCCOK = BIT0            ;BCC CHECKED GOOD
6801    000002                      BCCBAD = BIT1           ;BCC CHECKED BAD
6802    000004                      SACK = BIT2             ;SEND ACK
6803    000010                      SNAK = BIT3             ;SEND NAK
6804    000020                      SDATA = BIT4            ;SEND DATA
6805    000100                      RXD = BIT6              ;RECEIVER DONE
6806    000400                      NAKRX = BIT8            ;NAK RECEIVED
6807    001000                      MYDATA = BIT9           ;MY DATA
6808    002000                      SSTACK = BIT10          ;SEND START ACKNOWLEDGE
6809    004000                      SSTART = BIT11          ;SEND START
6810
```

```
6811
6812                            ;*****************************************************************************
6813                            ;       THE BELOW TABLES AND ASCIZ MESSAGES ARE USED IN DCLT
6814                            ;               REPORTING OF ERROR COUNTERS. THEY MUST REMAIN IN THE
6815                            ;               CURRENT SEQUENCE ELSE WE'LL BE REPORTING ERRONEOUS
6816                            ;               DATA.
6817                            ;*****************************************************************************
6818
6819  037146  037256    STALST: .WORD   STA0A           ;POINTER FOR OFFSET 0 ASCII
6820  037150  037305            .WORD   STA1A           ;POINTER FOR OFFSET 1 ASCII
6821  037152  037343            .WORD   STA2A           ;POINTER FOR OFFSET 2 ASCII
6822  037154  037377            .WORD   STA3A           ;POINTER FOR OFFSET 3 ASCII
6823  037156  037471            .WORD   STA4A           ;POINTER FOR OFFSET 4 ASCII
6824  037160  037555            .WORD   STA5A           ;POINTER FOR OFFSET 5 ASCII
6825  037162  037635            .WORD   STA6A           ;POINTER FOR OFFSET 6 ASCII
6826  037164  037721            .WORD   STA7A           ;POINTER FOR OFFSET 7 ASCII
6827  037166  040003            .WORD   STA10A          ;POINTER FOR OFFSET 10 ASCII
6828  037170  040065            .WORD   STA11A          ;POINTER FOR OFFSET 11 ASCII
6829  037172  040162            .WORD   STA12A          ;POINTER FOR OFFSET 12 ASCII
6830  037174  040256            .WORD   STA13A          ;POINTER FOR OFFSET 13 ASCII
6831  037176  040354            .WORD   STA14A          ;POINTER FOR OFFSET 14 ASCII
6832  037200  040452            .WORD   STA15A          ;POINTER FOR OFFSET 15 ASCII
6833  037202  040540            .WORD   STA16A          ;POINTER FOR OFFSET 16 ASCII
6834  037204  040625            .WORD   STA17A          ;POINTER FOR OFFSET 17 ASCII
6835
6836
6837                                    ;TABLE FOR PRINT ROUTINES
6838                                    ;PRIW: WORD ROUTINE
6839                                    ;PRIBB: BYTE/BYTE ROUTINE
6840                                    ;PRIBS: BYTE SPECIAL ROUTINE
6841
6842  037206  021640    STAIND: .WORD   PRIW
6843  037210  021640            .WORD   PRIW
6844  037212  021640            .WORD   PRIW
6845  037214  021666            .WORD   PRIBB
6846  037216  021666            .WORD   PRIBB
6847  037220  021666            .WORD   PRIBB
6848  037222  021666            .WORD   PRIBB
6849  037224  021666            .WORD   PRIBB
6850  037226  021666            .WORD   PRIBB
6851  037230  021724            .WORD   PRIBS
6852  037232  021724            .WORD   PRIBS
6853  037234  021724            .WORD   PRIBS
6854  037236  021724            .WORD   PRIBS
6855  037240  021724            .WORD   PRIBS
6856  037242  021724            .WORD   PRIBS
6857  037244  021640            .WORD   PRIW
6858
6859  037246  000000    LAST:   .WORD   0               ;LAST MESSAGE TO PRINT
6860  037250  000000    FIR:    .WORD   0               ;FIRST MESSAGE TO PRINT
6861  037252  000000    MES:    .WORD   0               ;HOLDS MESSAGE
6862  037254  000000    MESDATA:.WORD   0               ;DATA PART OF MESSAGE
6863
```

```
                                ;*****************************************************************************************
                                ;       THE BELOW ASCIZ MESSAGES USED IN 'RPT>' LEVEL OF DCLT
                                ;
                                .NLIST BEX
037256  047045  047445  022466  STA0A:  .ASCIZ  /%N%06%S2%ASTATUS FLAGS/
037305     045  022516  033117  STA1A:  .ASCIZ  /%N%06%S2%ADATA MSGS. TX'MITTD/
037343     045  022516  033117  STA2A:  .ASCIZ  /%N%06%S2%ADATA MSGS. RX'CVD/
037377     045  022516  031517  STA3A:  .ASCIZ  /%N%03%S5%AHIGHEST MSG # TX'D%N%03%S5%AHIGHEST MSG # ACK'D/
037471     045  022516  031517  STA4A:  .ASCIZ  /%N%03%S5%ANEXT MSG # TO TX%N%03%S5%ALAST MSG # TX'D/
037555     045  022516  031517  STA5A:  .ASCIZ  /%N%03%S5%AHIGHEST MSG # RX'D%N%03%S5%ATRIB ADDR/
037635     045  022516  031517  STA6A:  .ASCIZ  /%N%03%S5%AREMOTE TIME OUTS%N%03%S5%AGLOBAL CRC ERRS/
037721     045  022516  031517  STA7A:  .ASCIZ  /%N%03%S5%ANAK REASON%N%03%S5%ASELECT THRESH. ERRS/
040003     045  022516  031517  STA10A: .ASCIZ  /%N%03%S5%ARX THRESH ERRS%N%03%S5%ATX THRESH. ERRS/
040065     045  022516  031517  STA11A: .ASCIZ  /%N%03%S5%ADATA ERRORS OUT%N%S8%AHBCC %01%A BCC %01%A REP %01/
040162  047045  047445  022463  STA12A: .ASCIZ  /%N%03%S5%ADATA ERRORS IN%N%S8%AHBCC %01%A BCC %01%A REP %01/
040256  047045  047445  022463  STA13A: .ASCIZ  /%N%03%S5%ALOCAL BUFFER ERRS%N%S8%A NO BUFF %01%A TOO BIG  %01/
040354  047045  047445  022463  STA14A: .ASCIZ  /%N%03%S5%AREMOTE BUFFER ERRS%N%S8%A NO BUFF %01%A TOO BIG %01/
040452  047045  047445  022463  STA15A: .ASCIZ  /%N%03%S5%AREMOTE STA ERRS%N%S8%AOVRN %01%A FORMAT %01/
040540  047045  047445  022463  STA16A: .ASCIZ  /%N%03%S5%ALOCAL STA ERRS%N%S8%AOVRN %01%A FORMAT %01/
040625     045  022516  033117  STA17A: .ASCIZ  /%N%06%S2%ATX & RX THRESHOLD ERRORS(OVERFLOW)/
                                        .EVEN
                                .LIST BEX
                                ;*****************************************************************************************
```

```
6881
6882                            ;*****************************************************************************
6883                            ; PROTOCOL ROUTINE:
6884                            ;
6885                            ;       DESCRIPTION: IF THE USER SPECIFIES THE '/PROTOCOL' SWITCH THIS
6886                            ;                    ROUTINE WILL BE CALLED. THIS ROUTINE DECIDES IF
6887                            ;                    WE ARE TRANSMITTING AND/OR RECEIVING AND CALLS
6888                            ;                    THE NECESSARY DDCMP PROTOCOL ROUTINES.
6889                            ;
6890                            ;*****************************************************************************
6891
6892
6893  040702  013737  006600  037112  PROTOC: MOV     FLAG,IMFLAG     ;SAVE COPY OF MAIN CODE 'FLAG' VARIABLE
6894  040710  012737  000001  037122           MOV     #1,TXREADY      ;INIT TRANSMIITER DONE FLAG
6895  040716  005037  037114                   CLR     RXPRC           ;INIT RX PROCOTOL DONE
6896  040722  005037  037116                   CLR     TXPRC           ;INIT TX PROCOTOL DONE
6897  040726  005037  037144                   CLR     TIMEOUT         ;INIT PRINT TIMER
6898  040732  032737  000100  006574           BIT     #PRORUN,PARAM   ;PROTOCOL RUNNING ?
6899  040740  001067                           BNE     7$              ;YES,BRANCH
6900
6901                            ;; PROTOCOL NOT RUNNING -- SO FIRE UP THE LINK
6902  040742                            SETVEC  INVEC,#PRRXI,#PRI04     ;LOAD RX PROTOCOL INTERRUPT ROUTINE
6903  040742  012746  000200                                                    MOV     #PRI04,-(SP)
6904  040746  012746  044504                                                    MOV     #PRRXI,-(SP)
6905  040752  013746  011464                                                    MOV     INVEC,-(SP)
6906  040756  012746  000003                                                    MOV     #3,-(SP)
6907  040762  104437                                                            TRAP    C$SVEC
6908  040764  062706  000010                                                    ADD     #10,SP
6909  040770  005037  037136                   CLR     HDXMTP          ;INIT HALF DUPLEX/MULTI-POINT FLAG
6910  040774  005737  006572                   TST     FHDPLX          ;HALF DUPLEX ?
6911  041000  001403                           BEQ     2$              ;YES,BRANCH
6912  041002  005737  037126                   TST     MPPTP           ;MULTI POINT ?
6913  041006  001403                           BEQ     3$              ;NO,BRANCH
6914  041010  012737  000001  037136  2$:      MOV     #1,HDXMTP       ;SET HALF DUPLEX/MULTI-POINT
6915
6916  041016  012737  000036  006644  3$:      MOV     #30.,TIMERS     ;30 SECONDS TO START
6917  041024  012737  000001  037130           MOV     #1,SELECT       ;INIT SELECT
6918  041032  005037  037142                   CLR     TURNON          ;INIT YET ANOTHER FLAG
6919  041036  005037  037124                   CLR     PRUN            ;INIT ANOTHER FLAG
6920  041042  005037  037120                   CLR     ASTRT           ;INIT 'STACK SENT' FLAG
6921  041046  005037  037046                   CLR     PRSTAT          ;INIT STATUS WORD
6922  041052  004737  041730                   JSR     PC,PROINT       ;INIT PROTOCOL COUNTERS AND VARIABLES
6923  041056  005737  037126                   TST     MPPTP           ;MULTI - POINT MODE ?
6924  041062  001005                           BNE     4$              ;YES,BRANCH
6925  041064  052737  004000  037046           BIS     #SSTART,PRSTAT  ;TELL TX ROUTINE TO SEND 'START'
6926  041072  004737  045350                   JSR     PC,TXPROTO      ;GO SEND IT
6927  041076  004737  042100           4$:     JSR     PC,RXPROTO      ;GO WAIT FOR 'STACK' OR 'START'
6928  041102  032737  000100  006574           BIT     #PRORUN,PARAM   ;DID PROTOCOL START ?
6929  041110  001742                           BEQ     3$              ;NO,TRY AGAIN
6930  041112  012737  000001  037124           MOV     #1,PRUN         ;THIS FLAG USED IN RXPROTO ROUTINE
6931
6932                            ;; IF HALF DUPLEX OR MULTI POINT, WE MUST MANAGE THE LINK DIFFERENTLY
6933  041120  012737  000003  006644  7$:      MOV     #3.,TIMERS      ;SET UP TIMER
6934  041126  005737  037136                   TST     HDXMTP          ;HALF DUPLEX OR MULTI - POINT?
6935  041132  001076                           BNE     PROHDX          ;YES,BRANCH
6936
```

```
                                        ;; IF FULL DUPLEX AND ACTIVE MODE-- JUMP
041134  022737  000003  006566                CMP     #ACT,MODTYP     ;ACTIVE MODE?
041142  001440                                BEQ     200$            ;YES, BRANCH

                                        ;; PROTOCOL IS RUNNING -- LINK IS HOT SO SEND DATA

041144  032737  000010  037112  10$:    BIT     #QTX,IMFLAG     ;TRANSMITTING A MESSAGE ?
041152  001414                          BEQ     100$            ;NO,BRANCH
041154  052737  000020  037046  20$:    BIS     #SDATA,PRSTAT   ;SEND DATA FLAG
041162  004737  045350                  CALL    TXPROTO         ;GO SEND THE MESSAGE
041166  004737  042100                  CALL    RXPROTO         ;CHECK THE REPLY
041172  005737  037116                  TST     TXPRC           ;MESSAGE TRANSMITTED & "ACK'ED"?
041176  001766                          BEQ     20$             ;NO,BRANCH
041200  005237  037050                  INC     TMESTX          ;BUMP "TOTAL MESSAGES TRANSMITTED" COUNTER

041204  005737  037114          100$:   TST     RXPRC           ;RECEIVE PROTOCOL FINISHED ?
041210  001011                          BNE     110$            ;YES,BRANCH
041212  032737  000004  037112          BIT     #QRX,IMFLAG     ;RECEIVING A MESSAGE ?
041220  001002                          BNE     105$            ;YES,BRANCH
041222  000137  041670                  JMP     PROTEX          ;EXIT

041226  004737  042100          105$:   CALL    RXPROTO         ;GO PROCESS INCOMING MESSAGE
041232  000764                          BR      100$            ;SEE IF RECEIVE PROTOCOL COMPLETE
041234  005237  037052          110$:   INC     TMESRX          ;BUMP "TOTAL MESSAGES RECEIVED" COUNTER
041240  000137  041670                  JMP     PROTEX          ;EXIT


                                        ;; ACTIVE MODE (FULL DUPLEX AND POINT TO POINT LINKS)

041244  004737  041774          200$:   CALL    RXON            ;TURN ON RECEIVER
041250  052737  000020  037046  210$:   BIS     #SDATA,PRSTAT   ;SEND DATA FLAG
041256  004737  045350                  CALL    TXPROTO         ;DO SEND DATA MESSAGE
041262  004737  042100          215$:   CALL    RXPROTO         ;GO PROCESS INCOMING MESSAGE
041266  005737  037116                  TST     TXPRC           ;TX PROTOCOL DONE ?
041272  001766                          BEQ     210$            ;NO,BRANCH
041274  005737  037114                  TST     RXPRC           ;RX PROTOCOL DONE ?
041300  001770                          BEQ     215$            ;NO,BRANCH
041302  005237  037052                  INC     TMESRX          ;BUMP 'TOTAL MESSAGES RECEIVED'
041306  005237  037050                  INC     TMESTX          ;BUMP "TOTAL MESSAGE SENT" COUNTER

                                        ;; TXREADY SET IN TX INTERRUPT ROUTINE
041312  005737  037122          220$:   TST     TXREADY         ;MESSAGE SENT ?
041316  001775                          BEQ     220$            ;NO,BRANCH
041320  004737  041774                  CALL    RXON            ;TURN ON RECEIVER
041324  000137  041670                  JMP     PROTEX          ;EXIT


                                        ;;; THIS ROUTINE(PROHDX) IS USE IN HALF-DUPLEX PT-PT & MTP

041330                                  PROHDX:
041330  005737  006572          10$:    TST     FHDPLX          ;FULL DUPLEX ?
041334  001072                          BNE     PROFDX          ;YES,BRANCH
041336  032737  000010  037112          BIT     #QTX,IMFLAG     ;TRANSMITTING ?
041344  001424                          BEQ     100$            ;NO,BRANCH
041346  005737  037130          20$:    TST     SELECT          ;DO WE HAVE THE SELECT BIT ?
```

```
 6993  041352  001005                          BNE     30$             ;YES,BRANCH
 6994  041354  004737  041774                  CALL    RXON            ;TURN ON RX
 6995  041360  004737  042104          25$:    CALL    RXWAIT          ;TURN ON RX AND WAIT FOR SELECT BIT
 6996  041364  000770                          BR      20$             ;DID WE GET THE SELECT BIT ?
 6997  041366  052737  000020  037046  30$:    BIS     #SDATA,PRSTAT   ;SEND DATA FLAG
 6998  041374  004737  045350                  CALL    TXPROTO         ;GO SENT IT
 6999  041400  004737  042100                  CALL    RXPROTO         ;CHECK REPLY
 7000  041404  005737  037116                  TST     TXPRC           ;TX PROTOCOL DONE ?
 7001  041410  001756                          BEQ     20$             ;NO,BRANCH
 7002  041412  005237  037050                  INC     TMESTX          ;BUMP TOTAL MESSAGES SENT
 7003  041416  012737  000001  037116  100$:   MOV     #1,TXPRC        ;SET TX PROTOCOL DONE
 7004  041424  005737  037114          103$:   TST     RXPRC           ;RX PROTOCOL DONE ?
 7005  041430  001026                          BNE     150$            ;YES,BRANCH
 7006  041432  032737  000004  037112          BIT     #QRX,IMFLAG     ;RECEIVING ?
 7007  041440  001002                          BNE     110$            ;YES,BRANCH
 7008  041442  000137  041670                  JMP     PROTEX          ;EXIT
 7009                                  ;;WAS THE BALL TOSSED BACK IN OUR COURT ?
 7010  041446  005737  037130          110$:   TST     SELECT          ;HAVE WE RECEIVED THE SELECT BIT YET?
 7011  041452  001005                          BNE     130$            ;YES,BRANCH
 7012  041454  004737  041774                  CALL    RXON            ;TURN ON RECEIVER
 7013  041460  004737  042104          115$:   CALL    RXWAIT          ;PROCESS DATA
 7014  041464  000757                          BR      103$            ;TRY AGAIN
 7015  041466  052737  000004  037046  130$:   BIS     #SACK,PRSTAT    ;SEND ACK TO TURN THE LINE AROUND
 7016  041474  004737  045350                  CALL    TXPROTO         ;SEND IT
 7017  041500  004737  042100                  CALL    RXPROTO         ;GO RECEIVE THE PENDING MESSAGE
 7018  041504  000747                          BR      103$            ;BRANCH
 7019  041506  005237  037052          150$:   INC     TMESRX          ;BUMP'RECIEVED MESSAGE COUNTER'
 7020  041512  004737  041774                  CALL    RXON            ;TURN ON RX
 7021  041516  000137  041670                  JMP     PROTEX          ;EXIT
 7022
 7023                                  ;;THIS ROUTINE(PROFDX:) USED WITH FULL DUPLEX-MULTI POINT LINKS
 7024
 7025  041522  032737  000010  037112  PROFDX: BIT     #QTX,IMFLAG     ;TRANSMITTING ?
 7026  041530  001003                          BNE     10$             ;YES,BRANCH
 7027  041532  012737  000001  037116          MOV     #1,TXPRC        ;SET TRANSMIT PROTOCOL COMPLETE
 7028  041540  005737  037114          10$:    TST     RXPRC           ;WAS THE 1ST MESSAGE RX'ED DURING STARTUP?
 7029  041544  001015                          BNE     30$             ;YES,BRANCH
 7030  041546  032737  000004  037112          BIT     #QRX,IMFLAG     ;RECEIVING ?
 7031  041554  001004                          BNE     20$             ;YES,BRANCH
 7032  041556  012737  000001  037114          MOV     #1,RXPRC        ;SET RECEIVE PROTOCOL COMPLETE
 7033  041564  000410                          BR      100$            ;BRANCH
 7034  041566  004737  042100          20$:    CALL    RXPROTO         ;PROCESS INCOMING MESSAGE
 7035  041572  005737  037114                  TST     RXPRC           ;DONE ?
 7036  041576  001773                          BEQ     20$             ;NO,BRANCH
 7037  041600  005237  037052          30$:    INC     TMESRX          ;BUMP RX MESSAGE COUNT
 7038  041604  000400                          BR      100$            ;BRANCH
 7039
 7040  041606  005737  037116          100$:   TST     TXPRC           ;ANYTHING TO SEND ?
 7041  041612  001024                          BNE     135$            ;NO,BRANCH
 7042
 7043  041614  005737  037130          120$:   TST     SELECT          ;DO WE HAVE PERMISSION TO SEND ?
 7044  041620  001005                          BNE     130$            ;YES,BRANCH
 7045  041622  004737  041774                  CALL    RXON            ;TURN ON TX
 7046  041626  004737  042104          125$:   CALL    RXWAIT          ;WAIT ON SELECT BIT
 7047  041632  000770                          BR      120$            ;TRY AGAIN
 7048  041634  052737  000020  037046  130$:   BIS     #SDATA,PRSTAT   ;SEND DATA FLAG
```

```
 7049  041642  004737  045350                    CALL    TXPROTO         ;SEND DATA
 7050  041646  004737  042100                    CALL    RXPROTO         ;CHECK REPLY
 7051  041652  005737  037116                    TST     TXPRC           ;TRANSMIT PROTOCOL DONE ?
 7052  041656  001756                            BEQ     120$            ;NO,TRY AGAIN
 7053  041660  005237  037050                    INC     TMESTX          ;BUMP 'TOTAL MESSAGE SENT' COUNTER
 7054  041664  004737  041774          135$:     CALL    RXON            ;TURN ON RX SO THAT WE WON'T MISS ANY MESSAGES
 7055                                                                    ;; WHILE WE AWAY.
 7056  041670  022737  177777  037052  PROTEX:   CMP     #-1,TMESRX      ;;MAX?
 7057  041676  001002                            BNE     10$             ;NO,BRANCH
 7058  041700  005337  037052                    DEC     TMESRX          ;LATCH AT 177776
 7059  041704  022737  177777  037050  10$:      CMP     #-1,TMESTX      ;MAX?
 7060  041712  001002                            BNE     15$             ;NO,BRANCH
 7061  041714  005337  037050                    DEC     TMESTX          ;LATCH AT 177776
 7062  041720  013737  037112  006600  15$:      MOV     IMFLAG,FLAG     ;RESTORE 'FLAG' TO ITS ORIGINAL VALUE
 7063  041726  000207                            RTS     PC              ;EXIT
 7064
 7065
```

```
7066
7067                            ;*****************************************************************************
7068                            ;
7069                            ;       PROTOCOL INIT ROUTINE:
7070                            ;
7071                            ;               THIS ROUTINE WILL INITIALIZE THE ERROR COUNTERS AND  MESSAGE
7072                            ;               COUNTERS AS NEEDED FOR PROPER DDCMP PROTOCOL OPERATION.
7073                            ;               DURING NORMAL OPERATION THIS CODE WILL BE CALLED ONCE FROM
7074                            ;               PROTOCOL STARTUP ROUTINE.
7075                            ;
7076                            ;*****************************************************************************
7077
7078
7079  041730  010146            PROINT: MOV     R1,-(SP)        ;SAVE R1
7080  041732  113737  037061  006534        MOVB    TRIBN,TEMP      ;SAVE TRIB NUMBER
7081  041740  012701  037046            MOV     #PRSTAT,R1      ;FIRST LOCATION TO CLEAR
7082  041744  005021            10$:    CLR     (R1)+           ;CLEAR AND INCREMENT
7083  041746  020127  037110            CMP     R1,#PROEND      ;LAST LOCATION TO CLEAR
7084  041752  001374                    BNE     10$             ;NO,BRANCH
7085  041754  113737  006534  037061  20$:    MOVB    TEMP,TRIBN      ;RESTORE TRIB #
7086  041762  112737  000001  037056        MOVB    #1,T            ;FIRST MESSAGE # TO BE TRANMITTED
7087  041770  012601                    MOV     (SP)+,R1        ;RESTORE R1
7088  041772  000207                    RETURN                  ;EXIT
7089
7090                            ;*****************************************************************************
7091                            ; TURN ON RECEIVER ROUTINE:
7092                            ;
7093                            ;       DESCRIPTION: THIS ROUTINE SIMPLY ENABLES THE RECEIVER AND
7094                            ;                    INITIALIZES VARIABLES.
7095                            ;
7096                            ;*****************************************************************************
7097
7098  041774  005737  037142    RXON:   TST     TURNON          ;RX ALREADY ON ?
7099  042000  001036                    BNE     RXONEX          ;YES,BRANCH
7100  042002  005037  037046            CLR     PRSTAT          ;INIT STATUS WORD
7101  042006  005037  002656            CLR     RHDMSG          ;INIT 1ST WORD OF RX BUFFER
7102  042012  042737  000444  037134        BIC     #QRX!#BCC!#RXM,PRFLAG   ;FLAGS USED IN RX INTERRUPT ROUTINE
7103  042020  052737  001000  037046        BIS     #MYDATA,PRSTAT  ;ASSUME MESSAGE FOR ME
7104  042026  012737  002657  011512        MOV     #RHDMID,RMSGPT  ;BUFFER ADDRESS FOR HEADER PART ON MESSAGE
7105  042034  013737  002654  011514        MOV     HDMC,RMSGCC     ;INIT CHARACTER COUNT = 6
7106  042042  032737  000010  006574        BIT     #MOCHK,PARAM    ;MODEM CHANGES WANTED ?
7107  042050  001004                    BNE     20$             ;YES,BRANCH
7108  042052  052777  000120  147372        BIS     #RINTEN!RXENA,@RXCSR    ;TURN ON RX
7109  042060  000403                    BR      25$             ;BRANCH
7110  042062  052777  000160  147362  20$:    BIS     #RINTEN!RXENA!DSITEN,@RXCSR     ;TURN ON RX
7111  042070  012737  000001  037142  25$:    MOV     #1,TURNON       ;RX IS ON FLAG
7112  042076  000207            RXONEX: RETURN
7113
```

```
                        ;*****************************************************************************
                        ; RECEIVER PROTOCOL ROUTINE:
                        ;
                        ;       DESCRIPTION: THIS ROUTINE WILL PROCESS AN INCOMING MESSAGE
                        ;                    AND DETERMINE IF IT'S A VALID CONTROL OR DATA
                        ;                    MESSAGE. IF AN ERROR IS DETECTED THE APPROIATE
                        ;                    ERROR COUNTERS WILL BE UPDATED BY THE ERROR
                        ;                    ROUTINE.
                        ;
                        ;       SUBORDINATE ROUTINES USED: 'TXPROTO'
                        ;                                  'ERROR PROCESSOR'
                        ;
                        ;*****************************************************************************


042100                  RXPROTO:
042100  004737  041774          CALL    RXON            ;TURN ON RECEIVER
                        ;; WAIT FOR FIRST CHARACTER TO APPEAR IN RX BUFFER
042104                  RXWAIT:
042104                  20$:    BREAK                   ;CHECK FOR ^C
042104  104422                                                          TRAP    C$BRK
042106  105737  002657          TSTB    RHDMID          ;FIRST CHARACTER READ ?
042112  001007          BNE     30$                     ;YES, BRANCH
042114  005737  006644          TST     TIMERS          ;60 SECONDS ELAPSED ?
042120  001371          BNE     20$                     ;NO,BRANCH
042122  004737  043726          JSR     PC,ERRPRC       ;CALL ERROR PROCESSOR
042126  000137  043724          JMP     RXPREX          ;EXIT

                        ;; IF A CONTROL MESSAGE THEN TELL RX INTR. TO PROCESS HEADER ONLY
042132  123727  002657  000005  30$:    CMPB    RHDMID,#ENQ     ;CONTROL MESSAGE ?
042140  001003          BNE     40$                     ;NO,BRANCH
042142  052737  000040  037134          BIS     #RXM,PRFLAG     ;PROCESS HEADER ONLY

                        ;; WAIT FOR CRC TO BE CHECKED
042150  032737  000003  037046  40$:    BIT     #BCCOK!BCCBAD,PRSTAT    ;CRC CHECKED ?
042156  001021          BNE     50$                             ;YES,BRANCH
042160  005737  006644          TST     TIMERS          ;60 SECONDS ELAPSED ?
042164  001004          BNE     45$                     ;NO,BRANCH
042166  004737  043726          JSR     PC,ERRPRC       ;GO PROCESS ERROR
042172  000137  042100          JMP     RXPROTO         ;TRY AGAIN
```

```
 7158
 7159                                   ;; CHECK THAT RX WAS SERVICED QUICK ENOUGH(DETERMINED BY RX INTER. ROUTINE)
 7160  042176  032737  000010  037046  45$:    BIT     #SNAK,PRSTAT    ;RX OVERRUN ?
 7161  042204  001761                          BEQ     40$             ;NO,BRANCH
 7162  042206  004737  043726                  JSR     PC,ERRPRC       ;GO PROCESS ERROR
 7163  042212  004737  045350                  JSR     PC,TXPROTO      ;GO SEND NAK
 7164  042216  000137  042100                  JMP     RXPROTO         ;TRY AGAIN
 7165
 7166                                   ;; IF HEADER CRC ERROR THEN LOG IT AND SEND NAK
 7167  042222  032737  000002  037046  50$:    BIT     #BCCBAD,PRSTAT  ;CRC ERROR ?
 7168  042230  001430                          BEQ     60$             ;NO,BRANCH
 7169  042232  052737  000010  037046          BIS     #SNAK,PRSTAT    ;SET SNAK (SEND NAK)
 7170  042240  152737  000001  037073          BISB    #HEADBCC,INMASK ;SET THE MASK
 7171  042246  112737  000001  037064          MOVB    #HEADBCC,REANAK ;NAK REASON = 1
 7172  042254  105237  037072                  INCB    DERIN           ;LOG DATA ERROR INBOUND
 7173  042260  001003                          BNE     55$             ;BRANCH IF NOT OVERFLOW
 7174  042262  112737  000377  037072          MOVB    #377,DERIN      ;LATCH COUNTER AT 256.
 7175  042270  004737  043726  55$:    JSR     PC,ERRPRC       ;GO PROCESS ERROR
 7176  042274  012737  000001  037130          MOV     #1,SELECT       ;ASSUME S-BIT WAS SET IN FAULTY MESSAGE
 7177  042302  004737  045350                  JSR     PC,TXPROTO      ;GO SEND NAK
 7178  042306  000137  042100                  JMP     RXPROTO         ;TRY AGAIN
 7179
 7180                                   ;; NOW CHECK THE ADDRESS OF THE MESSAGE- IS IT FOR ME ?
 7181  042312  123737  037061  002664  60$:    CMPB    TRIBN,RHDADR    ;MY ADDRESS ?
 7182  042320  001422                          BEQ     70$             ;YES, BRANCH
 7183
 7184                                   ;; ITS NOT FOR ME, BUT COUNT IT OUT TO KEEP RX IN SYNC
 7185  042322  042737  001000  037046  62$:    BIC     #MYDATA,PRSTAT  ;MESSAGE NOT FOR ME
 7186  042330  032737  000100  037046          BIT     #RXD,PRSTAT     ;RECEIVER DONE ?
 7187  042336  001003                          BNE     65$             ;YES,BRANCH
 7188  042340  005737  006644                  TST     TIMERS          ;HAVE WE DAWDLED LONG ENOUGH ?
 7189  042344  001366                          BNE     62$             ;NO,BRANCH
 7190
 7191  042346  032737  000001  037046  65$:    BIT     #BCCOK,PRSTAT   ;DATA CRC OK ?
 7192  042354  001002                          BNE     67$             ;YES,BRANCH
 7193  042356  105237  037063                  INCB    GLOBCC          ;LOG GLOBAL CRC ERROR
 7194  042362  000137  042100          67$:    JMP     RXPROTO         ;GO RE-QUE BUFFER
 7195
 7196
 7197                                   ;; IS IT A CONTROL MESSAGE ? IF IT IS PROCESS IT
 7198
 7199  042366  105037  037066  70$:    CLRB    RXTHER          ;INIT RX THRESHOLD ERROR COUNTER
 7200  042372  122737  000005  002657          CMPB    #ENQ,RHDMID     ;CONTROL MESSAGE ?
 7201  042400  001402                          BEQ     75$             ;YES,BRANCH
 7202  042402  000137  043252                  JMP     200$            ;GO PROCESS DATA MESSAGE
 7203
 7204                                   ;; IS IT A NAK ?
 7205  042406  122737  000002  002660  75$:    CMPB    #NAK,RHDTYP     ;NAK?
 7206  042414  001022                          BNE     90$             ;NO,BRANCH
 7207  042416  032737  000100  006574          BIT     #PRORUN,PARAM   ;PROTOCOL RUNNING ?
 7208  042424  001002                          BNE     80$             ;YES,BRANCH
 7209  042426  000137  042100                  JMP     RXPROTO         ;IGNORE THIS MESSAGE
 7210  042432  052737  000400  037046  80$:    BIS     #NAKRX,PRSTAT   ;FLAG NAK RECEIVED
 7211  042440  004737  043726                  JSR     PC,ERRPRC       ;GO LOG NAK REASON
 7212  042444  052737  000020  037046          BIS     #SDATA,PRSTAT   ;SEND DATA
 7213  042452  004737  045350                  JSR     PC,TXPROTO      ;GO RE-TRANSMIT PREVIOUS MESSAGE
```

```
 7214  042456  000137  042100                JMP     RXPROTO         ;GO RE-QUE RX
```

```
7215
7216                                        ;; IS IT A ACK ?
7217  042462  122737  000001  002660  90$:    CMPB    #ACK,RHDTYP     ;ACK ?
7218  042470  001057                           BNE     100$            ;NO,BRANCH
7219  042472  032737  000100  006574           BIT     #PRORUN,PARAM   ;PROTOCOL RUNNING ?
7220  042500  001004                           BNE     93$             ;YES,BRANCH
7221  042502  052737  000100  006574           BIS     #PRORUN,PARAM   ;TELL THE WORLD THAT LINK HAS STARTED
7222  042510  000445                           BR      97$             ;EXIT
7223  042512  123737  037056  002662  93$:     CMPB    T,RHDREP        ;CORRECT MESSAGE # ACKNOWLEDGED ?
7224  042520  001405                           BEQ     95$             ;YES,BRANCH
7225  042522  005737  037136                   TST     HDXMTP          ;HALF DUPLEX/MULTI -POINT ?
7226  042526  001036                           BNE     97$             ;YES,BRANCH
7227  042530  000137  042100                   JMP     RXPROTO         ;TRY AGAIN
7228  042534  105037  037067           95$:    CLRB    TXTHER          ;INIT. TX THRESHOLD COUNTER
7229  042540  113737  037056  037054           MOVB    T,N             ;HIGHEST SEQUENTIAL MESSAGE # SENT
7230  042546  113737  037056  037057           MOVB    T,X             ;HIGHEST MESSAGE # SENT
7231  042554  113737  002662  037055           MOVB    RHDREP,A        ;HIGHEST MESSAGE # ACKNOWLEDGED TO THIS STATION
7232  042562  105237  037056                   INCB    T               ;# OF NEXT DATA MESSAGE TO BE TRANSMITTED
7233  042566  012737  177777  037116           MOV     #-1,TXPRC       ;TRANSMIT PROTOCOL COMPLETE
7234  042574  022737  000003  006566           CMP     #ACT,MODTYP     ;ACTIVE MODE ?
7235  042602  001010                           BNE     97$             ;NO,BRANCH
7236  042604  005737  037114                   TST     RXPRC           ;RX PROTOCOL COMPLETE?
7237  042610  001005                           BNE     97$             ;YES,BRANCH
7238  042612  005737  006572                   TST     FHDPLX          ;HALF DUPLEX?
7239  042616  001402                           BEQ     97$             ;YES,BRANCH
7240  042620  000137  042100                   JMP     RXPROTO         ;GO PROCESS INCOMING MESSAGE
7241
7242  042624  000137  043724           97$:    JMP     RXPREX          ;EXIT
7243
7244                                        ;; IS IT A REP ?
7245  042630  122737  000003  002660  100$:    CMPB    #REP,RHDTYP     ;REP ?
7246  042636  001054                           BNE     150$            ;NO,BRANCH
7247
7248                                        ;; NUM = R ?
7249  042640  032737  000100  006574           BIT     #PRORUN,PARAM   ;PROTOCOL RUNNING ?
7250  042646  001002                           BNE     110$            ;YES,BRANCH
7251  042650  000137  042100                   JMP     RXPROTO         ;IGNORE MESSAGE- TRY AGAIN
7252  042654  123737  002663  037060  110$:    CMPB    RHDNUM,R        ;HAVE WE RECEIVED THIS MESSAGE ?
7253  042662  001015                           BNE     120$            ;NO, BRANCH
7254  042664  052737  000004  037046           BIS     #SACK,PRSTAT    ;SET SEND ACK
7255  042672  105237  037062                   INCB    REMTMO          ;BUMP REMOTE TIME OUT COUNTER
7256  042676  001003                           BNE     115$            ;BRANCH IF NOT OVERFLOW
7257  042700  112737  000377  037062           MOVB    #377,REMTMO     ;LATCH COUNTER AT 256.
7258  042706  004737  045350           115$:   JSR     PC,TXPROTO      ;GO SEND ACK
7259  042712  000137  042100                   JMP     RXPROTO         ;TRY AGAIN
7260
7261                                        ;; NUM <> R
7262  042716  052737  000010  037046  120$:    BIS     #SNAK,PRSTAT    ;SET SEND NAK
7263  042724  112737  000003  037064           MOVB    #REPSENT,REANAK ;SET REASON FOR NAK
7264  042732  105237  037072                   INCB    DERIN           ;BUMP DATA ERROR INBOUND
7265  042736  001003                           BNE     125$            ;BRANCH IF NOT OVERFLOW
7266  042740  112737  000377  037072           MOVB    #377,DERIN      ;LATCH AT 256.
7267  042746  152737  000004  037073  125$:    BISB    #REPMSK,INMASK  ;ERROR REASON IS REMOTE TIME OUT
7268  042754  004737  043726                   JSR     PC,ERRPRC       ;PROCESS NAK
7269  042760  004737  045350                   JSR     PC,TXPROTO      ;GO SEND NAK
7270  042764  000137  042100                   JMP     RXPROTO         ;TRY AGAIN
```

```
7271
7272                                        ;; IS IT A START ?
7273  042770  122737  000006  002660  150$:  CMPB    #STRT,RHDTYP    ;START ?
7274  042776  001071                       BNE     170$            ;NO,BRANCH
7275  043000  032737  000100  006574       BIT     #PRORUN,PARAM   ;PROTOCOL RUNNING ?
7276  043006  001007                       BNE     160$            ;YES,BRANCH
7277  043010  052737  002000  037046       BIS     #SSTACK,PRSTAT  ;SEND START ACKNOWLEDGE
7278  043016  004737  045350               JSR     PC,TXPROTO      ;GO SEND STACK
7279  043022  000137  042100               JMP     RXPROTO         ;GO TO RX ROUTINE AND EXPECT ACK OR DATA
7280                                                                ;; IF DMV OR DMP.
7281
7282  043026  052737  000200  006574  160$:  BIS     #ABORT,PARAM    ;TELL MAIN CODE TO ABORT!!
7283  043034  012737  177777  037114       MOV     #-1,RXPRC       ;RECEIVE PROTCOL DONE
7284  043042  012737  177777  037116       MOV     #-1,TXPRC       ;TRANSMIT PROTOCOL DONE
7285  043050                               PRINTF  #165$           ;FATAL ERROR
7286  043050  012746  043074                                                   MOV     #165$,-(SP)
7287  043054  012746  000001                                                   MOV     #1,-(SP)
7288  043060  010600                                                           MOV     SP,R0
7289  043062  104417                                                           TRAP    C$PNTF
7290  043064  062706  000004                                                   ADD     #4,SP
7291  043070  000137  043724               JMP     RXPREX          ;EXIT
7292
7293                                      .NLIST BEX
      043074  047045  040445  052123  165$:  .ASCIZ  /%N%ASTART RECEIVED WITH PROTOCOL RUNNING--ABORTING!!/
              043162                      .EVEN
                                         .LIST BEX
7294
7295                                        ;; IS IT A STACK ? IF SO SEND AN 'ACK'
7296
7297  043162  122737  000007  002660  170$:  CMPB    #STACK,RHDTYP   ;STACK ?
7298  043170  001012                       BNE     180$            ;NO, BRANCH
7299  043172  052737  000004  037046       BIS     #SACK,PRSTAT    ;TELL TX ROUTINE TO SEND ACK
7300  043200  004737  045350               JSR     PC,TXPROTO      ;SEND ACK
7301  043204  052737  000100  006574       BIS     #PRORUN,PARAM   ;SET 'PROTOCOL RUNNING' FLAG
7302  043212  000137  043724               JMP     RXPREX          ;EXIT
7303
7304
7305                                        ;; IF WE GOT TO HERE, WE HAVE A STRANGE PROBLEM !
7306  043216  052737  000010  037046  180$:  BIS     #SNAK,PRSTAT    ;SET SEND NAK FLAG
7307  043224  105237  037102               INCB    LOSTER          ;LOCAL STATION ERROR
7308  043230  152737  000021  037103       BISB    #FORMERR,LSMASK ;FORMAT ERROR
7309  043236  004737  043726               JSR     PC,ERRPRC       ;PROCESS ERROR
7310  043242  004737  045350               JSR     PC,TXPROTO      ;SEND NAK
7311  043246  000137  042100               JMP     RXPROTO         ;TRY AGAIN
7312                                      ;; END OF CONTROL MESSAGE PROCESSOR
7313
7314
7315
```

```
7316
7317                                    ;;;HERE WE BEGIN PROCESSING DATA PART OF MESSAGE
7318
7319  043252  005737  037114            200$:   TST     RXPRC           ;ALREADY PROCESSED A MESSAGE?
7320  043256  001432                            BEQ     215$            ;NO,BRANCH
7321  043260  042737  001000  037046            BIC     #MYDATA,PRSTAT  ;TELL RX INTERRUPT ROUTINE NOT TO STORE THIS
7322  043266  005737  037116                    TST     TXPRC           ;TX PROTOCOL COMPLETE ?
7323  043272  001037                            BNE     220$            ;YES,BRANCH
7324                                    ;; SEE IF IMPLICIT ACK IMBEDDED IN THIS MESSAGE
7325  043274  123737  037056  002662            CMPB    T,RHDREP        ;RESP = MESSAGE SENT?
7326  043302  001033                            BNE     220$            ;NO,BRANCH
7327  043304  113737  037056  037054            MOVB    T,N             ;HIGHEST # SENT
7328  043312  113737  037056  037057            MOVB    T,X             ;HIGHEST # SENT
7329  043320  113737  037056  037055            MOVB    T,A             ;HIGHEST MESSAGE ACK'ED
7330  043326  105237  037056                    INCB    T               ;NEXT MESSAGE TO SEND
7331  043332  012737  177777  037116            MOV     #-1,TXPRC       ;TX PROTOCOL FINISHED
7332  043340  000137  043724                    JMP     RXPREX          ;EXIT
7333
7334  043344  105237  037060            215$:   INCB    R               ;EXPECTED #?
7335  043350  123737  037060  002663            CMPB    R,RHDNUM        ;CORRECT MESSAGE #?
7336  043356  001423                            BEQ     300$            ;YES,PROCESS IT
7337  043360  105337  037060                    DECB    R               ;SUBTRACT 1
7338  043364  042737  001000  037046            BIC     #MYDATA,PRSTAT  ;JUST COUNT OUT MESSAGE-DON'T PUT IN BUFFER
7339  043372  032737  000100  037046  220$:     BIT     #RXD,PRSTAT     ;WAIT FOR DONE
7340  043400  001003                            BNE     250$            ;BRANCH
7341  043402  005737  006644                    TST     TIMERS          ;TIME OUT?
7342  043406  001371                            BNE     220$            ;NO,BRANCH
7343
7344                                    ;;SEND AN "ACK"
7345  043410  052737  000004  037046  250$:     BIS     #SACK,PRSTAT    ;SEND ACK
7346  043416  004737  045350                    CALL    TXPROTO         ;GO SEND IT
7347  043422  000137  042100                    JMP     RXPROTO         ;TRY AGAIN
7348
7349                                    ;; IS DATA PART OF MESSAGE COMPLETE ?
7350  043426  032737  000100  037046  300$:     BIT     #RXD,PRSTAT     ;MESSAGE COMPLETE ?
7351  043434  001021                            BNE     330$            ;YES,BRANCH
7352
7353                                    ;; IS THE LINE DEAD ?
7354  043436  005737  006644                    TST     TIMERS          ;TIMED-OUT ?
7355  043442  001004                            BNE     305$            ;NO,BRANCH
7356  043444  004737  043726                    JSR     PC,ERRPRC       ;GO PROCESS TIMER ERROR
7357  043450  000137  042100                    JMP     RXPROTO         ;TRY AGAIN
7358
7359                                    ;; CHECK FOR RECEIVER OVERRUN OR BUFFER PROBLEM
7360  043454  032737  000010  037046  305$:     BIT     #SNAK,PRSTAT    ;DID RX INTERRUPT SET THIS ?
7361  043462  001761                            BEQ     300$            ;NO,BRANCH
7362
7363                                    ;;RX ERROR SEND A NAK AND TRY AGAIN
7364  043464  004737  043726                    JSR     PC,ERRPRC       ;GO PROCESS ERROR
7365  043470  004737  045350                    JSR     PC,TXPROTO      ;SEND NAK
7366  043474  000137  042100                    JMP     RXPROTO         ;TRY AGAIN
7367
7368                                    ;;CHECK FOR DATA CRC ERROR
7369  043500  032737  000001  037046  330$:     BIT     #BCCOK,PRSTAT   ;DATA CRC GOOD ?
7370  043506  001022                            BNE     400$            ;YES,BRANCH
7371
```

```
 7372                                   ;; LOG CRC ERROR AND SEND A NAK
 7373  043510  052737  000010  037046          BIS     #SNAK,PRSTAT    ;SET SEND NAK FLAG
 7374  043516  105237  037072                  INCB    DERIN           ;BUMP DATA ERROR INBOUND COUNTER
 7375  043522  001003                          BNE     340$            ;BRANCH IF NOT OVERFLOW
 7376  043524  112737  000377  037072          MOVB    #377,DERIN      ;LATCH AT 256.
 7377  043532  152737  000002  037073  340$:   BISB    #DATABCC,INMASK ;SET DATA CRC BIT
 7378  043540  004737  043726                  JSR     PC,ERRPRC       ;GO PROCESS ERROR
 7379  043544  004737  045350                  JSR     PC,TXPROTO      ;GO SEND NAK
 7380  043550  000137  042100                  JMP     RXPROTO         ;TRY AGAIN
 7381
 7382
 7383                                   ;; WE HAVE A GOOD MESSAGE !!! SO ACKNOWLEDGE IT
 7384  043554  032737  000100  006574  400$:   BIT     #PRORUN,PARAM   ;PROTOCOL RUNNING?
 7385  043562  001007                          BNE     420$            ;YES,BRANCH
 7386  043564  005737  037120                  TST     ASTRT           ;DID WE SEND A STACK?
 7387  043570  001001                          BNE     415$            ;YES,BRANCH
 7388  043572  000454                          BR      RXPREX          ;EXIT
 7389
 7390                                   ;; NOTE: DMV/DPM WILL SEND 'START - STACK - DATA' FOR STARTUP SEQUENCE
 7391  043574  052737  000100  006574  415$:   BIS     #PRORUN,PARAM   ;SET PROTOCOL RUNNING
 7392
 7393                                   ;; CHECK FOR AN IMPLICIT 'ACK'
 7394  043602  123737  037056  002662  420$:   CMPB    T,RHDREP        ;RESP = MESSAGE SENT ?
 7395  043610  001016                          BNE     450$            ;NO,BRANCH
 7396  043612  113737  037056  037054          MOVB    T,N             ;HIGHEST SEQ MESSAGE # SENT
 7397  043620  113737  037056  037057          MOVB    T,X             ;HIGHEST MESSAGE SENT
 7398  043626  113737  037056  037055          MOVB    T,A             ;HIGHEST MESSAGE 'ACK'ED'
 7399  043634  105237  037056                  INCB    T               ;NEXT MESSAGE # TO TRANSMIT
 7400  043640  012737  177777  037116          MOV     #-1,TXPRC       ;SET TRANSMIT PROTOCOL COMPLETE
 7401  043646  052737  000004  037046  450$:   BIS     #SACK,PRSTAT    ;SET SEND ACK FLAG
 7402  043654  004737  045350                  JSR     PC,TXPROTO      ;SEND ACK
 7403  043660  012737  177777  037114          MOV     #-1,RXPRC       ;RECEIVE MESSAGE PROTOCOL FINISHED
 7404  043666  005737  037124                  TST     PRUN            ;PROTOCOL RUNNING ?
 7405  043672  001414                          BEQ     RXPREX          ;NO,BRANCH
 7406  043674  005737  037136                  TST     HDXMTP          ;FULL DUPLEX PT-PT?
 7407  043700  001011                          BNE     RXPREX          ;NO,BRANCH
 7408  043702  022737  000003  006566          CMP     #ACT,MODTYP     ;ACTIVE MODE ?
 7409  043710  001005                          BNE     RXPREX          ;NO,BRANCH
 7410  043712  005737  037116                  TST     TXPRC           ;TRANSMIT PROTOCOL COMPLETE ?
 7411  043716  001002                          BNE     RXPREX          ;YES,BRANCH
 7412  043720  000137  042100                  JMP     RXPROTO         ;GO PROCESS MESSAGE
 7413
 7414  043724  000207          RXPREX: RETURN                          ;DONE !!
 7415
```

```
7416
7417                            ;**********************************************************************
7418                            ;       ERROR PROCESSING ROUTINE (ERRPRC):
7419                            ;
7420                            ;       DESCRIPTION: THIS ROUTINE IS USED TO PROCESS INBOUND AND
7421                            ;                    OUTBOUND ERRORS. ALSO THE 60 SECOND 'WATCHDOG'
7422                            ;                    TIMER IS CHECKED.
7423                            ;
7424                            ;                  THE MAJORITY OF THE CODE IS USED IN PROCESSING
7425                            ;                    OUTBOUND ERRORS (NAKS RECEIVED). THE NAK REASON
7426                            ;                    TYPE IS DETERMINED AND THE APPROIATE ERROR
7427                            ;                    COUNTER IN INCREMENTED. IF THE TRANSMIT THRESHOLD
7428                            ;                    COUNTER (TXTHER) OR RECEIVE THRESHOLD COUNTER
7429                            ;                    REACHES 7, IT IS CLEARED AND THE CUMULATIVE
7430                            ;                    RECEIVE/TRANSMIT THRESHOLD ERROR (RXTXTE) COUNTER
7431                            ;                    IS BUMPED.
7432                            ;**********************************************************************
7433
7434
7435
7436                            ;;CHECK THE WATCHDOG TIMER
7437
7438  043726  005737  006644   ERRPRC: TST     TIMERS          ;60 SECONDS ELAPSED
7439  043732  001075                   BNE     10$             ;NO,BRANCH
7440  043734  032737  000100  006574         BIT     #PRORUN,PARAM   ;PROTOCOL RUNNING ?
7441  043742  001034                   BNE     7$              ;YES,BRANCH
7442
7443                            ;; INFORM USER OF 'START - STACK' TIMEOUT
7444  043744  005037  006542           CLR     TEMP3           ;INIT IT
7445  043750  005037  006544           CLR     TEMP4           ;INIT IT
7446  043754  012737  017311  006540   MOV     #DVEM5,TEMP2    ;"TIME OUT IN START-STACK SEQUENCE"
7447  043762  113737  002660  006542   MOVB    RHDMCC,TEMP3    ;RECEIVED DATA
7448  043770  113737  002646  006544   MOVB    HDMCC,TEMP4     ;TRANSMITTED DATA
7449  043776  004737  020160           JSR     PC,LGDVE        ;LOG TIME OUT IN EVENT LOG
7450  044002  005237  006506           INC     ERRCNT          ;BUMP ERROR COUNT
7451  044006                           ERRSOFT 10.,DVEM5,ERR13 ;PRINT ERROR
7452  044006  104457                                                   TRAP    C$ERSOFT
7453  044010  000012                                                   .WORD   10
7454  044012  017311                                                   .WORD   DVEM5
7455  044014  017634                                                   .WORD   ERR13
7456  044016  005237  006502           INC     OPVAR           ;BUMP ERROR COUNTER
7457  044022  012737  000036  006644   MOV     #30.,TIMERS     ;RE-INIT TIMER
7458  044030  000137  044502           JMP     ERREXT          ;EXIT
7459
7460                            ;; INFORM USER OF "DATA MESSAGE" TIMEOUT
7461  044034  005237  037144   7$:     INC     TIMEOUT         ;BUMP COUNTER
7462  044040  022737  000024  037144   CMP     #20.,TIMEOUT    ;60 SECONDS ?
7463  044046  001023                   BNE     9$              ;NO,BRANCH
7464  044050  012737  017110  006540   MOV     #DVEM2,TEMP2    ;"TIME OUT WAITING FOR RX OR TX TO COMPLETE"
7465  044056  017737  145370  006542   MOV     @RXCSR,TEMP3    ;RECEIVER ADDRESS
7466  044064  017737  145370  006544   MOV     @TXCSR,TEMP4    ;TRANSMIT ADDRESS
7467  044072  004737  020160           JSR     PC,LGDVE        ;LOG ERROR
7468  044076  005237  006506           INC     ERRCNT          ;BUMP ERROR COUNT
7469  044102                           ERRSOFT 7,DVEM2,ERR13   ;PRINT ERROR
7470  044102  104457                                                   TRAP    C$ERSOFT
7471  044104  000007                                                   .WORD   7
```

```
                                                                                    .WORD   DVEM2
                                                                                    .WORD   ERR13
                        CLR     TIMEOUT         ;INIT TIMEOUT
                9$:     MOV     #3.,TIMERS      ;SET UP TIMER
                        BR      ERREXT          ;EXIT

                ;;IF A 'NAK' RECEIVED THEN PROCESS IT
                10$:    BIT     #NAKRX,PRSTAT   ;NAK RECEIVED?
                        BEQ     100$            ;NO,BRANCH

                ;;IF TRANSMIT THRESHOLD COUNTER = 7 THEN BUMP CUMULATIVE TXRX COUNTER
                        CMPB    #7,TXTHER       ;THRESHOLD REACHED?
                        BEQ     20$             ;YES,BRANCH
                        INCB    TXTHER          ;BUMP TRANSMIT THESHOLD
                        BR      30$             ;BRANCH
                20$:    INC     RXTXTE          ;BUMP TRANSMIT/RECEIVE THRESHOLD COUNTER
                        CLRB    TXTHER          ;SET TRANSMIT COUNTER TO ZERO

                ;;;;DETERMINE THE 'NAK' REASON

                ;;HEADER CRC ERROR ?
                30$:    BIC     #BIT15!BIT14,RHDTYP     ;CLEAR SELECT & QS FLAG
                        CMPB    #HEADBCC,RHDTYP+1       ;HEADER CRC ERROR?
                        BNE     35$                     ;NO,BRANCH
                        INCB    DEROUT                  ;LOG ERROR
                        BNE     32$                     ;BRANCH IF NOT OVERFLOW
                        MOVB    #377,DEROUT             ;LATCH AT 256.
                32$:    BISB    #HEADBCC,OUTMASK        ;SET MASK
                        BR      ERREXT                  ;EXIT

                ;;DATA CRC ERROR ?
                35$:    CMPB    #DATABCC,RHDTYP+1       ;DATA CRC ERROR ?
                        BNE     40$                     ;NO,BRANCH
                        INCB    DEROUT                  ;LOG ERROR
                        BNE     37$                     ;BRANCH IF NOT OVERFLOW
                        MOVB    #377,DEROUT             ;LATCH AT 256.
                37$:    BISB    #DATABCC,OUTMASK        ;SET MASK
                        BR      ERREXT                  ;EXIT

                ;;REMOTE STATION BUFFER NOT AVAILABLE?
                40$:    CMPB    #BUFFNA,RHDTYP+1        ;BUFFER NOT AVAILABLE?
                        BNE     45$                     ;NO,BRANCH
                        INCB    RBUFER                  ;LOG ERROR
                        BNE     43$                     ;BRANCH IF NOT OVERFLOW
                        MOVB    #377,RBUFER             ;LATCH AT 256.
                43$:    BISB    #BNAMSK,RBMASK          ;SET MASK
                        BR      ERREXT                  ;EXIT

                ;;REMOTE STATION RECEIVER OVERRUN?
                45$:    CMPB    #RXOVRUN,RHDTYP+1       ;RECEIVER OVERRUN?
                        BNE     50$                     ;NO, BRANCH
                        INCB    RMSTER                  ;LOG ERROR
                        BNE     47$                     ;BRANCH IF NO OVERFLOW
                        MOVB    #377,RMSTER             ;LATCH AT 256.
```

```
 7528  044342  152737  000001  037101  47$:    BISB    #RXOVMSK,RMMASK         ;SET MASK
 7529  044350  000454                          BR      ERREXT                  ;EXIT
 7530
 7531                                  ;;REMOTE STATION MESSAGE TOO LONG?
 7532  044352  122737  000020  002661  50$:    CMPB    #MESLONG,RHDTYP+1       ;MESSAGE TOO LONG?
 7533  044360  001012                          BNE     55$                     ;NO,BRANCH
 7534  044362  105237  037076                  INCB    RBUFER                  ;LOG REMOTE STATION BUFFER ERROR
 7535  044366  001003                          BNE     52$                     ;BRANCH IF NO OVERFLOW
 7536  044370  112737  000377  037076          MOVB    #377,RBUFER             ;LATCH AT 256.
 7537  044376  152737  000002  037077  52$:    BISB    #MTLMSK,RBMASK          ;SET MASK
 7538  044404  000436                          BR      ERREXT                  ;EXIT
 7539
 7540                                  ;;REMOTE STATION FORMAT ERROR?
 7541  044406  122737  000021  002661  55$:    CMPB    #FORMERR,RHDTYP+1       ;REMOTE STATION FORMAT ERROR?
 7542  044414  001012                          BNE     100$                    ;NO,BRANCH
 7543  044416  105237  037100                  INCB    RMSTER                  ;LOG ERROR
 7544  044422  001003                          BNE     57$                     ;BRANCH IF NO OVERFLOW
 7545  044424  112737  000377  037100          MOVB    #377,RMSTER             ;LATCH AT 256.
 7546  044432  152737  000002  037101  57$:    BISB    #FMTMSK,RMMASK          ;SET MASK
 7547  044440  000420                          BR      ERREXT                  ;EXIT
 7548
 7549
 7550                                  ;; IF SEND NAK (SNAK=1) THEN BUMP RECEIVER THRESHOLD ERROR COUNTER
 7551                                  ;;.  IF RXTHER = 7 THE BUMP RXTXTE COUNTER AND CLEAR RXTHER
 7552  044442  032737  000010  037046  100$:    BIT     #SNAK,PRSTAT            ;SEND NAK ?
 7553  044450  001414                          BEQ     ERREXT                  ;NO, BRANCH
 7554
 7555  044452  122737  000007  037066          CMPB    #7,RXTHER               ;RECEIVER THRESHOLD = 7?
 7556  044460  001403                          BEQ     120$                    ;YES,BRANCH
 7557  044462  105237  037066                  INCB    RXTHER                  ;BUMP COUNTER
 7558  044466  000405                          BR      ERREXT                  ;BRANCH
 7559
 7560  044470  005237  037104          120$:   INC     RXTXTE                  ;BUMP CUMULATIVE COUNTER
 7561  044474  105037  037066                  CLRB    RXTHER                  ;INIT RECEIVER THRESHOLD COUNTER
 7562  044500  000400                          BR      ERREXT                  ;EXIT
 7563
 7564
 7565
 7566  044502  000207                  ERREXT: RETURN
 7567
```

```
                                                .SBTTL          RECEIVER PROTOCOL INTERRUPT ROUTINE

                                                ;++
                                                ; FUNCTIONAL DESCRIPTION:
                                                ;       THIS ROUTINE IS USED ONLY WHEN THE "/PROTOCOL" SWITCH
                                                ;       IS SPECIFIED BY THE USER.
                                                ;
                                                ;       WHEN A RX INT. OCCURS THIS ROUTINE DECIDES IF IT IS A 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 INTERRUPT, THE ROUTINE CHECK FOR AN OVERRUN
                                                ;       CONDITION AND IF SET
                                                ;
                                                ; 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
                                                ;--

044504                                                  BGNSRV  PRRXI
044504                                                                                  PRRXI::
044504  010246                                          MOV     R2,-(SP)        ;SAVE R2
044506  017737  144740  011476                          MOV     @RXCSR,IRXCSR   ;MOV RX CSR TO IMAGE
044514  032737  000010  006574                          BIT     #MOCHK,PARAM    ;ANY MODEM CHANGES TO REPORT
044522  001462                                          BEQ     PRIN21          ;IF NOT IGNORE DS CHANGE.
044524  032737  002000  037112                          BIT     #INOVR,IMFLAG   ;IS INIT OVER
044532  001456                                          BEQ     PRIN21          ;NO THEN IGNORE DS CHANGE.
044534  032737  004000  037112                          BIT     #FIRST,IMFLAG   ;FIRST TIME HERE?
044542  001452                                          BEQ     PRIN21          ;YES,BRANCH
044544  005737  011476                                  TST     IRXCSR          ;DATA SET CHANGE ?
044550  100047                                          BPL     PRIN21          ;IF DATA SET CHANGE IS NOT SET BR
044552  013737  011476  011474                          MOV     IRXCSR,CMODS    ;MOV THE NEW MODEM STATUS IN
044560  042737  106760  011474                          BIC     #106760,CMODS   ;CLEAR BITS NOT RELATING TO MODEM STATUS
044566  032777  000040  144664                          BIT     #TM,@TXCSR      ;TEST MODE?
044574  001403                                          BEQ     PRIN2           ;NO,BRANCH
044576  052737  000040  011474                          BIS     #TM,CMODS       ;SET TM MODE IN CHANGE STATUS
044604  013737  011474  006542          PRIN2:          MOV     CMODS,TEMP3
044612  013737  007554  006544                          MOV     MODS,TEMP4
044620  023737  006544  006542                          CMP     TEMP4,TEMP3     ;COMPARE OLD TO CURRENT
044626  001406                                          BEQ     10$             ;INC GLITCH COUNT
044630  005237  011522                                  INC     MHRCNT          ;INC HARD COUNT
044634  012737  016717  006540                          MOV     #HRDMSG,TEMP2   ;SET UP HARD MESG.
044642  000405                                          BR      PRIN1
044644  005237  011520                  10$:            INC     MGLCNT          ;INC GLITCH COUNT
044650  012737  016671  006540                          MOV     #GLMSG,TEMP2    ;SET UP GLITCH
044656  004737  020334                  PRIN1:          JSR     PC,LOGMSC       ;GO LOG MODEM STATUS CHANGE
```

```
7624  044662  013737  011474  007554          MOV     CMODS,MODS      ;MOVE CURRENT TO OLD
7625
7626                                  ;;TEST FOR DATA
7627
7628  044670  032737  002200  011476  PRIN21: BIT     #RSTARY!RDATRY,IRXCSR   ;RX DONE OR STATUS AVAILABLE ?
7629  044676  001002                          BNE     10$             ;YES,BRANCH
7630  044700  000137  045344                  JMP     PRINEX          ;EXIT
7631  044704  017737  144546  011500  10$:    MOV     @RDSR,IRDSR     ;SAVE A COPY OF STATUS & DATA . .
7632  044712  032737  004000  011500          BIT     #RXOVER,IRDSR   ;OVERRUN ERROR ?
7633  044720  001414                          BEQ     PRIN4           ;NO,BRANCH
7634                                  ;;IF AN OVERRUN THEN LOG ERROR,SET NAK REASON,TURN OFF RX & EXIT
7635  044722  052737  000010  037046          BIS     #SNAK,PRSTAT    ;TELL MAIN CODE ABOUT OVERRUN ERROR
7636  044730  105237  037102                  INCB    LOSTER          ;LOG LOCAL STATION ERROR
7637  044734  152737  000001  037103          BISB    #RXOVMSK,LSMASK ;SET RX OVERRUN MASK BIT
7638  044742  112737  000011  037064          MOVB    #RXOVRUN,REANAK ;SET REASON FOR SENDING NAK
7639  044750  000570                          BR      PRIN8           ;GO TURN OFF RX AND EXIT
7640
7641
7642                                  ;;;IF IN MULTI-POINT MODE AND NOT MY ADDRESS THEN JUST BUMP CHAR COUNT
7643
7644                                  ;;STORE AWAY DATA
7645  044752  032737  001000  037046  PRIN4:  BIT     #MYDATA,PRSTAT  ;STORE THIS DATA ?
7646  044760  001406                          BEQ     10$             ;NO,BRANCH
7647  044762  013702  011512                  MOV     RMSGPT,R2       ;SET RX MESSAGE POINTER
7648  044766  113722  011500                  MOVB    IRDSR,(R2)+     ;STORE DATA AWAY
7649  044772  010237  011512                  MOV     R2,RMSGPT       ;SAVE UPDATED MESSAGE POINTER
7650
7651                                  ;;DECREMENT CHARACTER COUNT
7652  044776  005337  011514          10$:    DEC     RMSGCC          ;ALL DATA RECEIVED ?
7653  045002  001160                          BNE     PRINEX          ;NO,BRANCH
7654  045004  032737  000400  037134          BIT     #BCC,PRFLAG     ;CRC ALREADY CHECKED?
7655  045012  001022                          BNE     PRIN5           ;YES,BRANCH
7656  045014  032737  100000  011500          BIT     #CRCOK,IRDSR    ;CRC GOOD ?
7657  045022  001004                          BNE     PRIN6           ;YES,BRANCH
7658  045024  052737  000002  037046          BIS     #BCCBAD,PRSTAT  ;TELL MAIN CODE ABOUT CRC ERROR
7659  045032  000537                          BR      PRIN8           ;DISABLE INTERRUPTS AND EXIT
7660
7661                                  ;;; READ 2 MORE CHARACTERS TO FLUSH CRC
7662  045034  052737  000400  037134  PRIN6:  BIS     #BCC,PRFLAG     ;SET CRC ALREADY CHECKED FLAG
7663  045042  012737  000002  011514          MOV     #2,RMSGCC       ;COUNT TWO CHARACTERS
7664  045050  012737  011516  011512          MOV     #BCCW,RMSGPT    ;CRC STORAGE ADDRESS
7665  045056  000532                          BR      PRINEX          ;EXIT
7666
7667  045060  052737  000001  037046  PRIN5:  BIS     #BCCOK,PRSTAT   ;TELL MAIN CODE CRC HAS BEEN CHECKED
7668  045066  123737  037061  002664          CMPB    TRIBN,RHDADR    ;MY MESSAGE
7669  045074  001404                          BEQ     5$              ;YES,BRANCH
7670  045076  042737  001000  037046          BIC     #MYDATA,PRSTAT  ;DON'T STORE IT
7671  045104  000407                          BR      7$              ;BRANCH
7672                                  ;; SELECT BIT SET ?
7673  045106  032737  100000  002660  5$:     BIT     #BIT15,RHDMCC   ;SELECT BIT SET?
7674  045114  001403                          BEQ     7$              ;NO,BRANCH
7675  045116  012737  000001  037130          MOV     #1,SELECT       ;WE NOW HAVE THE RIGHT TO TRANSMIT,IF HALF-DUPL
7676
7677  045124  032737  000040  037134  7$:     BIT     #RXM,PRFLAG     ;READ DATA MESSAGE ?
7678  045132  001071                          BNE     PRIN7           ;NO,BRANCH
7679
```

```
                                ;;SET UP TO READ IN DATA PART OF MESSAGE
045134  042737  000003  037046          BIC     #BCCOK!BCCBAD,PRSTAT ;CLEAR FLAGS (USED IN PROTOCOL CODE)
045142  052737  000040  037134          BIS     #RXM,PRFLAG     ;SET DATA MESSAGE READ FLAG
045150  042737  000400  037134          BIC     #BCC,PRFLAG     ;CLEAR CRC CHECKED FLAG(USED BY THIS ROUTINE)
045156  042737  140000  002660          BIC     #BIT15!BIT14,RHDMCC ;CLEAR SELECT & QS BITS

                                ;;IS ALLOCATED BUFFER SPACE LARGE ENOUGH FOR MESSAGE?
045164  023727  002660  001000          CMP     RHDMCC,#512.    ;WILL MESSAGE FIT IN MAX BUFFER SPACE
045172  003414                          BLE     10$             ;YES,BRANCH

                                ;;MESSAGE TOO LONG !! LOG ERROR
045174  105237  037074                  INCB    LBUFER          ;LOG LOCAL BUFFER ERROR
045200  152737  000002  037075          BISB    #MTLMSK,LBMASK  ;SET MESSAGE TOO LONG BIT
045206  112737  000020  037064          MOVB    #MESLONG,REANAK ;SET REASON FOR NAK
045214  152737  000010  037046          BISB    #SNAK,PRSTAT    ;SET SEND NAK FLAG
045222  000443                          BR      PRIN8           ;TURN OFF RX & EXIT

                                ;; IF A NEW BUFFER IS AVAILABLE
                                ;;SET BUFFER AND CHARACTER COUNT FOR MESSAGE
045224  005737  037114          10$:    TST     RXPRC           ;IS RX PROTOCOL DONE?
045230  001420                          BEQ     15$             ;NO,BRANCH
045232  105237  037074                  INCB    LBUFER          ;LOCAL BUFFER ERROR
045236  001003                          BNE     12$             ;OVERFLOW?
045240  012737  000377  037074          MOV     #377,LBUFER     ;LATCH A 256.
045246  152737  000001  037075  12$:    BISB    #BNAMSK,LBMASK  ;SET MASK
045254  112737  000010  037064          MOVB    #BUFFNA,REANAK  ;SET NAK REASON
045262  152737  000010  037046          BISB    #SNAK,PRSTAT    ;SET "SEND NAK FLAG"
045270  000412                          BR      PRIN7           ;EXIT

045272  013737  006470  011512  15$:    MOV     DVRXA,RMSGPT    ;MESSAGE BUFFER ADDRESS
045300  013737  002660  011514          MOV     RHDMCC,RMSGCC   ;CHARACTER COUNT OF MESSAGE
045306  013737  002660  006472          MOV     RHDMCC,DVRCC    ;TELL MAIN CODE HOW LARGE MESSAGE IS
045314  000413                          BR      PRINEX          ;EXIT

                                ;;MESSAGE COMPLETE
045316  052737  000004  037134  PRIN7:  BIS     #QRX,PRFLAG     ;SET MESSAGE COMPLETE FLAG(USED BY MAIN CODE)
045324  052737  000100  037046          BIS     #RXD,PRSTAT     ;MESSAGE COMPLETE(USED BY PROTOCOL MODULE)

045332  005037  037142  PRIN8:  CLR     TURNON          ;RX NOT ON
045336  042777  000120  144106          BIC     #RINTEN+RXENA,@RXCSR    ;TURN OFF RECEIVER

045344  012602          PRINEX: MOV     (SP)+,R2        ;RESTORE R2
045346                          ENDSRV
                                                                L10022:
045346  000002                                                          RTI
```

```
                        .SBTTL  PROTOCOL TRANSMIT ROUTINE

                        ;++
                        ;; FUNCTIONAL DESCRIPTION:
                        ;;      THIS ROUTINE IS USED TO SETUP EITHER CONTROL MESSAGES OR
                        ;;      DATA MESSAGES FOR TRANSMISSION.
                        ;;      IF THE SEND ACK(SACK) IS SET AN 'ACK' MESSAGE WILL BE SETUP
                        ;;      AND TRANSMITTED.
                        ;;      IF THE SEND NAK(SNAK) IS SET A 'NAK' MESSAGE WILL BE SETUP
                        ;;      AND TRANSMITTED.
                        ;;      IF NO FLAGS ARE SET, A DATA MESSAGE WILL BE SETUP AND SENT.
                        ;;      IF THE NETWORK IS HALF-DUPLEX THEN REQUEST TO SEND(RTS) WILL
                        ;;      BE ASSERTED BEFORE TRANSMISSION.

045350 013737 037046 037132 TXPROT: MOV     PRSTAT,IMPRSTAT ;SAVE A COPY OF FLAGS
045356 032737 000100 006574         BIT     #PRORUN,PARAM   ;PROTOCOL RUNNING ?
045364 001407                       BEQ     7$              ;NO,BRANCH
045366 022737 000003 006566         CMP     #ACT,MODTYP     ;ACTIVE MODE?
045374 001003                       BNE     7$              ;NO,BRANCH
045376 005737 037122        5$:     TST     TXREADY         ;TRANSMITTER READY FOR MESSAGE ?
045402 001775                       BEQ     5$              ;NO,BRANCH

                        ;;IF HALF DUPLEX OR MULTI-POINT LINK, WE NEED THE SELECT BIT
                        ;;; BEFORE WE CAN SEND.
045404 005737 037136        7$:     TST     HDXMTP          ;FULL DUPLEX AND PT TO PT ?
045410 001416                       BEQ     9$              ;YES,BRANCH
045412 005737 037130        6$:     TST     SELECT          ;OK TO SEND ?
045416 001003                       BNE     8$              ;YES,BRANCH
045420 004737 042100                CALL    RXPROTO         ;GO WAIT ON SELECT BIT
045424 000772                       BR      6$              ;TRY AGAIN

                        ;; DETERMINE WHAT TO SEND
045426 005737 006572        8$:     TST     FHDPLX          ;FULL DUPLEX?
045432 001005                       BNE     9$              ;YES,BRANCH
                        ;; IF HALF DUPLEX WE MUST DISABLE RX BEFORE SENDING
045434 042777 000120 144010         BIC     #RINTEN!RXENA,@RXCSR ;TURN OFF RX
045442 005037 037142                CLR     TURNON          ;"RX NOT ON" FLAG
045446 013737 037132 037046 9$:     MOV     IMPRSTAT,PRSTAT ;RESTORE ORIGINAL FLAGS
045454 113737 037061 002652         MOVB    TRIBN,HDMADR    ;SET TRIB ADDRESS
045462 042737 001000 006600         BIC     #PAD,FLAG       ;THIS BIT USED IN TX INTER ROUTINE
045470 005037 037122                CLR     TXREADY         ;TRANSMITTER BUSY
045474 005037 037130                CLR     SELECT          ;IF HALF DUPLEX/MTP MODE
045500 032737 000004 037046         BIT     #SACK,PRSTAT    ;SEND ACK ?
045506 001021                       BNE     10$             ;YES,BRANCH
045510 032737 000010 037046         BIT     #SNAK,PRSTAT    ;SEND NAK ?
045516 001062                       BNE     50$             ;YES, BRANCH
045520 032737 004000 037046         BIT     #SSTART,PRSTAT  ;SEND START ?
045526 001106                       BNE     60$             ;YES,BRANCH
045530 032737 002000 037046         BIT     #SSTACK,PRSTAT  ;SEND START ACKNOWLEDGE ?
045536 001126                       BNE     70$             ;YES,BRANCH
045540 032737 000020 037046         BIT     #SDATA,PRSTAT   ;SEND DATA MESSAGE ?
045546 001151                       BNE     100$            ;YES,BRANCH
045550 000000                       HALT                    ;FATAL ERROR
```

```
7784
7785
7786
7787                                    ;; SETUP TO SEND AN 'ACK'
7788  045552  052737  000020  006600  10$:    BIS     #TXM,FLAG           ;SEND HEADER ONLY(USED IN TX INTER. ROUTINE)
7789  045560  112737  000005  002645          MOVB    #ENQ,HDMID          ;CONTROL MESSAGE
7790  045566  012737  000001  002646          MOV     #ACK,HDMTYP         ;ACK CONTROL MESSAGE
7791  045574  052737  140000  002646          BIS     #BIT15!BIT14,HDMTYP ;SET SELECT & QS FLAG
7792  045602  005737  037136                  TST     HDXMTP              ;HALF DUPLEX OR MULTI - POINT
7793  045606  001415                          BEQ     20$                 ;NO,BRANCH
7794  045610  005737  037116                  TST     TXPRC               ;ANY THING TO SENT ?
7795  045614  001012                          BNE     20$                 ;NO,BRANCH
7796  045616  032737  000100  006574          BIT     #PRORUN,PARAM       ;PROTOCOL RUNNING?
7797  045624  001406                          BEQ     20$                 ;NO,BRANCH
7798  045626  042737  100000  002646          BIC     #BIT15,HDMTYP       ;CLEAR SELECT BIT
7799  045634  012737  000001  037130          MOV     #1,SELECT           ;WE HAVE SOMETHING TO SEND, SO KEEP THE LINE
7800  045642  113737  037060  002650  20$:    MOVB    R,HDMREP            ;SET RESPONSE NUMBER
7801  045650  105037  002651                  CLRB    HDMNUM              ;FILLER
7802  045654  042737  000004  037046          BIC     #SACK,PRSTAT        ;CLEAR SEND ACK FLAG
7803  045662  000526                          BR      200$                ;GO SEND IT
7804
7805
7806                                    ;; SETUP TO SEND A 'NAK'
7807  045664  052737  000020  006600  50$:    BIS     #TXM,FLAG           ;TELL TX INTERRUPT TO SEND HEADER ONLY
7808  045672  112737  000005  002645          MOVB    #ENQ,HDMID          ;CONTROL MESSAGE
7809  045700  012737  000002  002646          MOV     #NAK,HDMTYP         ;'NAK'
7810  045706  113737  037064  002647          MOVB    REANAK,HDMTYP+1     ;REASON FOR NAK
7811  045714  052737  140000  002646  55$:    BIS     #BIT15!BIT14,HDMTYP ;SET SELECT & QS FLAGS
7812  045722  105037  002651                  CLRB    HDMNUM              ;FILLER
7813  045726  113737  037060  002650          MOVB    R,HDMREP            ;LAST MESSAGE RECEIVED CORRECTLY
7814  045734  042737  000010  037046          BIC     #SNAK,PRSTAT        ;CLEAR SEND NAK FLAG
7815  045742  000476                          BR      200$                ;GO SEND IT
7816
7817
7818                                    ;; SETUP TO SEND START MESSAGE
7819  045744  052737  000020  006600  60$:    BIS     #TXM,FLAG           ;TELL TX INT. ROUTINE TO SEND HEADER ONLY
7820  045752  112737  000005  002645          MOVB    #ENQ,HDMID          ;CONTROL MESSAGE
7821  045760  012737  000006  002646          MOV     #STRT,HDMTYP        ;START MESSAGE
7822  045766  052737  140000  002646          BIS     #BIT15!BIT14,HDMTYP ;SET SELECT & QS FLAGS
7823  045774  105037  002650                  CLRB    HDMREP              ;FILLER
7824  046000  105037  002651                  CLRB    HDMNUM              ;FILLER
7825  046004  042737  004000  037046          BIC     #SSTART,PRSTAT      ;CLEAR SEND START FLAG
7826  046012  000452                          BR      200$                ;GO SEND IT
7827
7828                                    ;; SETUP TO SEND STACK MESSAGE
7829  046014  052737  000020  006600  70$:    BIS     #TXM,FLAG           ;TELL TX INT. TO SEND HEADER ONLY
7830  046022  112737  000005  002645          MOVB    #ENQ,HDMID          ;CONTROL MESSAGE
7831  046030  012737  000007  002646          MOV     #STACK,HDMTYP       ;START ACKNOWLEDGE MESSAGE
7832  046036  052737  140000  002646          BIS     #BIT15!BIT14,HDMTYP     ;SET SELECT &QS FLAGS
7833  046044  105037  002650                  CLRB    HDMREP              ;FILLER
7834  046050  105037  002651                  CLRB    HDMNUM              ;FILLER
7835  046054  012737  177777  037120          MOV     #-1,ASTRT           ;START HAS BEEN ACKNOWLEDGED
7836  046062  042737  002000  037046          BIC     #SSTACK,PRSTAT      ;CLEAR SEND STACK FLAG
7837  046070  000423                          BR      200$                ;GO SEND IT
7838
7839
```

```
                                ;; SETUP TO SEND DATA
046072  042737  000020  006600  100$:   BIC     #TXM,FLAG       ;TELL TX INTERRUPT TO SEND HEADER + DATA
046100  112737  000201  002645          MOVB    #SOH,HDMID      ;DATA MESSAGE
046106  013737  006456  002646          MOV     DVTCC,HDMCC     ;CHARACTERS COUNT
046114  052737  140000  002646          BIS     #BIT15!BIT14,HDMCC  ;SET SELECT & QS FLAGS
046122  113737  037060  002650          MOVB    R,HDMREP        ;LAST MESSAGE RECEIVED CORRECTLY
046130  113737  037056  002651          MOVB    T,HDMNUM        ;THIS MESSAGE NUMBER
046136  000400                          BR      200$            ;GO SEND IT

                                ;; GO SET 'REQUEST TO SEND'
046140  004737  036574          200$:   JSR     PC,CTSSR        ;GO SET REQUEST TO SEND
046144  052737  004000  037112          BIS     #FIRST,IMFLAG   ;TELL THE CTSSR SUBROUTINE TO SKIP DELAY

                                ;; SETUP TO TRANSMIT HEADER PORTION OF MESSAGE
046152  012737  002645  011502  210$:   MOV     #HDMID,MSGPTR   ;HEADER MESSAGE ADDRESS
046160  012737  000006  011504          MOV     #6,MSGCC        ;CHARACTER COUNT OF HEADER = 6
046166  012737  000020  011506          MOV     #20,SYNCC       ;NUMBER OF SYNCS TO TRANSMIT

                                ;; SEND THE DATA
046174  052777  000120  143256  215$:   BIS     #TXENA!#TINTEN,@TXCSR  ;TURN ON TRANSMITTER


                                ;;IF ACTIVE MODE, TURN ON TX AND GET OUT IN A HURRY
                                ;; NOTE: START UP SEQUENCE OPERATES LIKE HALF-DUPLEX

046202  005737  037136          217$:   TST     HDXMTP          ;FULL DUPLEX PT-PT
046206  001005                          BNE     220$            ;NO,BRANCH
046210  022737  000003  006566          CMP     #ACT,MODTYP     ;ACTIVE MODE ?
046216  001001                          BNE     220$            ;NO,BRANCH
046220  000406                          BR      TXPREX          ;EXIT

046222                          220$:   BREAK                                   TRAP    C$BRK
046222  104422
046224  005737  037122                  TST     TXREADY         ;TX FINISHED ?
046230  001774                          BEQ     220$            ;NO, BRANCH

                                ;; IF HALF-DUPLEX OR MULTI-POINT REQUEST TO SEND WILL BE DROPPED
046232  004737  037006          230$:   JSR     PC,CLRRTS       ;DROP RTS IF HALF DUPLEX

046236  004737  041774          TXPREX: CALL    RXON            ;TURN ON RX IF NECESSARY
046242  000207                          RETURN                  ;WE ARE DONE !


                                        .EVEN

                                        ENDTST
046244                                                          L10017:
046244                                                                  TRAP    C$ETST
046244  104401
```

```
                        .SBTTL  HARDWARE PARAMETER CODING SECTION


                        ;++
                        ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
                        ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                        ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                        ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                        ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                        ; WITH THE OPERATOR.
                        ;--

046246                          BGNHRD
046246  000030                                                    .WORD L10023-L$HARD/2
046250                                                          L$HARD::


                        .SBTTL          DEVICE INDEPENDENT SECTION

046250                          GPRML   DPLX,0,1,YES
046250  000130                                                    .WORD   T$CODE
046252  046330                                                  .WORD   DPLX
046254  000001                                                  .WORD   1


                        .SBTTL          DEVICE DEPENDENT SECTION

046256                          GPRMA   CSRADR,2,0,160000,177776,YES
046256  001031                                                  .WORD   T$CODE
046260  046361                                                  .WORD   CSRADR
046262  160000                                                  .WORD   T$LOLIM
046264  177776                                                  .WORD   T$HILIM
046266                          GPRMA   VECTOR,4,0,300,776,YES
046266  002031                                                  .WORD   T$CODE
046270  046407                                                  .WORD   VECTOR
046272  000300                                                  .WORD   T$LOLIM
046274  000776                                                  .WORD   T$HILIM
046276                          GPRML   RNODM,14,1,YES
046276  006130                                                  .WORD   T$CODE
046300  046442                                                  .WORD   RNODM
046302  000001                                                  .WORD   1
046304                          XFERT   ENDHWL
046304  012024                                                  .WORD   T$CODE
046306                          GPRML   PTPMLP,10,1,YES
046306  004130                                                  .WORD   T$CODE
046310  046466                                                  .WORD   PTPMLP
046312  000001                                                  .WORD   1
046314                          XFERF   ENDHWL
046314  006044                                                  .WORD   T$CODE
046316                          GPRMD   TRIBNQ,12,D,-1,1,255.,YES
046316  005052                                                  .WORD   T$CODE
046320  046524                                                  .WORD   TRIBNQ
046322  177777                                                  .WORD   -1
046324  000001                                                  .WORD   T$LOLIM
```

```
 7951  046326  000377                                                          .WORD   T$HILIM
 7952  046330                        ENDHWL: ENDHRD
 7953                                                                          .EVEN
 7954  046330                                                          L10023:
 7955
 7956                                .NLIST  BEX

                                     ;DEVICE INDEPENDENT QUESTIONS

       046330  052506  046114  042040 DPLX:   .ASCIZ  /FULL DUPLEX OPERATION : /


                                     ;DEVICE DEPENDENT QUESTION

       046361     104  053105  041511 CSRADR: .ASCIZ  /DEVICE CSR ADDRESS : /
       046407     111  052116  051105 VECTOR: .ASCIZ  /INTERRUPT VECTOR ADDRESS: /
       046442  042522  047515  042524 RNODM:  .ASCIZ  /REMOTE NODE "ITEP":/
       046466  051511  052040  044510 PTPMLP: .ASCIZ  /IS THIS A MULTIPOINT NETWORK:/
       046524  042101  051104  051505 TRIBNQ: .ASCIZ  /ADDRESS THIS STATION:/
                                     .LIST   BEX
                                             .EVEN
 7957
 7958
 7959
```

```
                                ;.SBTTL SOFTWARE PARAMETER CODING SECTION

                                ;++
                                ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--

                                ;       BGNSFT



                                ;       ENDSFT



                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                ; TEMPORARY PATCH AREA - FOR DEBUG PURPOSES
                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

046552                          $PATCH:
046552  000030                          .BLKW   30


046632                                  LASTAD

                                                                        .EVEN
046632  000000                                                          .WORD   0
046634  000000                                                          .WORD   0
046636                          L$LAST::
046636                                  ENDMOD

000001                          .END
```

```
A       037055              6691#  7231*  7329*  7398*
ABO   = 000026              2247#  4679
ABORT = 000200              2227#  4823   5565   7282
ACK   = 000001              6791#  7217   7790
ACT   = 000003              2200#  4805   5254   5334   5585   6938   7234   7408   7748   7867
ACTATV  031310              5028   5254#
ACTBCR  031114              5046   5215#
ACTCHK  031524              5008   5300#
ACTCLB  030436              5104   5118#
ACTCLP  031636              5042   5328#
ACTCLR  030126              5006   5059#
ACTCOP  030734              5016   5178#
ACTCRC  031540              5037   5306#
ACTCSE  030262              5011   5090#
ACTCST  030354              5012   5106#
ACTDLL  031356              5032   5268#
ACTDME  030662              5048   5159   5162#
ACTDMQ  030654              5049   5161#
ACTDMS  030632              5047   5156#
ACTDMX  030670              5163#
ACTECH  031434              5036   5284#
ACTEQO  031056              5020   5204#
ACTERR  021412              3475   3514#
ACTEXT  030212              5052   5078#
ACTFUL  021370              3476   3508#
ACTHLP  030146              5010   5065#
ACTLIS  031346              5031   5265#
ACTLLP  031646              5043   5330#
ACTLPX  031664              5325   5327   5329   5331   5334#
ACTLXX  031726              5298   5319   5322   5335   5344#
ACTMEX  031302              5197   5213   5235   5240   5246   5249   5251#
ACTME1  031236              5224   5226   5228   5230   5232   5239#
ACTMOP  031616              5040   5324#
ACTMOS  031546              5051   5309#
ACTMS0  031136              5021   5223#
ACTMS1  031144              5022   5225#
ACTMS2  031154              5023   5227#
ACTMS3  031164              5024   5229#
ACTMS4  031174              5025   5231#
ACTMS5  031204              5026   5233#
ACTMS6  031222              5027   5236#
ACTM2X  031404              5255   5263   5266   5269   5272   5276#
ACTNO   031424              5035   5281#
ACTNUF  030116              5045   5056#
ACTNUL  030124              5005   5057#
ACTNUM  030744              5017   5181#
ACTOPM  031036              5018   5199#
ACTPAS  031320              5029   5257#
ACTPRO  031554              5038   5312#
ACTPRT  030222              5050   5081#
ACTQFG  031560              5301   5304   5307   5310   5314#
ACTREC  031340              5030   5262#
ACTREX  021344              3473   3499#
ACTRHL  021300              3472   3485#
ACTRLG  021354              3474   3503#
ACTRLP  031656              5044   5332#
```

```
ACTRNF  021270              3477    3481#
ACTRNL  021276              3471    3482#
ACTRPS  031606              5039    5321#
ACTRSO  021434              3478    3521#
ACTRUN  030236              5009    5085#
ACTSEX  031246              5053    5243#
ACTSHO  030136              5007    5062#
ACTSHW  030476              5096    5110    5129#   5148
ACTSTE  030676              5013    5168#
ACTSTS  031532              5019    5303#
ACTSTT  030706              5014    5171#
ACTSTX  030714              5169    5172#
ACTSZE  030724              5015    5175#
ACTTAL  031376              5034    5274#
ACTTLP  031626              5041    5326#
ACTTRA  031366              5033    5271#
ADDCC   023264              3921#   4937    4979
ADDC1   023360              3923    3938#
ADR   = 000020 G            2173#
ALCK    032334              2659    5486#
ALCK1   032456              5541    5550    5552#   5605
ALCK2   032522              5553    5562#   5637
ALCK2A  032772              5602    5607#
ALCK3   033016              5572    5609    5612#
ALCK3A  033110              5619    5625#
ALCK3B  033122              5626    5628#
ALCK3C  033116              5623    5627#   5634
ALCK4   033144              5613    5632#
ALCK4A  033152              5631    5635#
ALCK4B  033164              5636    5638#
ALCK5   032376              5540#   5627
ALLTR   032376              5418    5441    5465    5491    5539#
ASSEMB= 000010              1945
ASTRT   037120              6749#   6920*   7386    7835*
ATVMOD= 000027              2309#   2804
BAD     006555              2628#   3024    5713*
BADCHR= 000051              2327#   2900
BASM1   015501              3018#   3865
BASM2   015472              3018#   3884
BASM3   015463              3018#   3876
BCC   = 000400              2263#   6184    6187    6382    6400    6427    7102    7654    7662    7683
BCCBAD= 000002              6801#   7152    7167    7658    7681
BCCOK = 000001              6800#   7152    7191    7369    7667    7681
BCCW    011516              2972#   6402    7664
BDCLK   014202              3018#   4551
BIT0  = 000001 G            2146#   2220    2251    2368    3522    3593    6777    6780    6800
BIT00 = 000001 G            2135#   2146
BIT01 = 000002 G            2134#   2145
BIT02 = 000004 G            2133#   2144
BIT03 = 000010 G            2132#   2143
BIT04 = 000020 G            2131#   2142
BIT05 = 000040 G            2130#   2141
BIT06 = 000100 G            2129#   2140
BIT07 = 000200 G            2128#   2139
BIT08 = 000400 G            2127#   2138
BIT09 = 001000 G            2126#   2137
```

```
BIT1   = 000002 G          2145#   2221    2252    2361    2369    3596    6778    6779    6801
BIT10  = 002000 G          2125#   2267    2366    6808
BIT11  = 004000 G          2124#   2269    2367    2380    6809
BIT12  = 010000 G          2123#   2352
BIT13  = 020000 G          2122#   2350
BIT14  = 040000 G          2121#   2354    7495    7684    7791    7811    7822    7832    7844
BIT15  = 100000 G          2120#   2376    2377    2379    7495    7673    7684    7791    7798    7811    7822    7832    7844
BIT2   = 000004 G          2144#   2222    2253    2353    2370    3599    6776    6802
BIT3   = 000010 G          2143#   2223    2254    2371    6803
BIT4   = 000020 G          2142#   2224    2259    2362    2372    6804
BIT5   = 000040 G          2141#   2225    2261    2355    2356    2363    6152    6153    6155
BIT6   = 000100 G          2140#   2226    2255    2364    2373    6805
BIT7   = 000200 G          2139#   2227    2256    2365    5380    5383
BIT8   = 000400 G          2138#   2263    2374    6806
BIT9   = 001000 G          2137#   2265    2351    2375    6059    6807
BLDBEX   023472            3982    3986#
BLDBUF   023362            3964#   4796    4803    4938    4980    5125    5392
BLDB1    023372            3969#
BLDB2    023434            3977#   3984
BLDB3    023452            3980#   3985
BNAMSK=  000001            6780#   7519    7704
BOE    = 000400 G          2177#
BUFEX    014347            3018#   4910    4951
BUFFNA=  000010            6768#   7514    7705
BUFLIM=  001000            2188#   2572    2573    2574    3922    3934    4054    4907    4948    5118    5388
BYTBIT   006514            2607#   3870    5157*   5161*
CABLE  = 000002            2207#   6029
CBLLOP=  000045            2323#   2921
CCURAD   006452            2587#   4066*   4074    4078*   4792*   4800    4978    4983*   5103*
CHECK  = 000003            2289#   2852
CLEAR  = 000001            2287#   2790    4900    5059
CLIACT   027736            4858    4997#
CLIALN=  000007            2280#   2898
CLIALP=  000006            2279#   2838    2848    2903
CLIBCR   012027            3018#   5216
CLIBDL   011722            3018#   5339
CLIBIF=  000003            2276#
CLIBR  = 000002            2275#   2787    2789    2791    2794    2796    2805    2807    2809    2811    2813    2816    2818
                           2833    2835    2845    2853    2857    2859    2863    2865    2868    2871    2878    2880    2882
                           2884    2886    2888    2890    2892    2896    2899    2906    2908    2913    2920    2922    2924
                           2926    2931    3454
CLIBRX   011702            3018#   4310
CLIDEC=  000011            2282#   2912    2930
CLIERM   011605            3018#   3418    4864
CLIERR=  000000            2273#   2797    2900    3464
CLIEXI=  000001            2274#   2779    2781    2783    2785    2824    2826    2936    3446    3448    3450    3452    3457
                           3459    3463    3465
CLINBG   011660            3018#   4353
CLINPS   011764            3018#   5290
CLINUF   011635            3018#   3428    4873
CLINUM=  000005            2278#
CLIOCT=  000010            2281#   2840    2842    3462
CLIPW    012125            3018#   5356
CLIRAC   021232            3412    3466#
CLIRT    021030            3411    3443#
CLISEO   012074            3018#   5186
```

```
CLISPA= 000004          2277#  2777  2801  2822  2831  2839  2897  3444
CLISTR= 000012          2283#  2780  2782  2784  2786  2788  2790  2793  2795  2802  2804  2806  2808
                        2810   2812  2815  2817  2823  2825  2832  2834  2850  2852  2856  2858  2862
                        2864   2867  2870  2877  2879  2881  2883  2885  2887  2889  2891  2905  2907
                        2919   2921  2923  2925  3447  3449  3451  3453  3456  3458  3460
CLITRE  007716          2773#  4857
CLI$PM  011572          3018#  4851
CLI$RP  011600          3018#  3405
CLKBR   006622          2667#  4536*
CLKCSR  006620          2666#  3205*  3226*  4507  4525*  4526*  4527*  4537*  4671*  4759*
CLKEN   006630          2670#  3226   4515*  4528*  4537   4759
CLKHZ   006626          2669#  3208   3223   3360   4541   4559
CLKINT  017754  G       3203#  4609
CLKSET  017730          3158#  4514   4524
CLKVEC  006624          2668#  4505*  4535*  4610
CLRRTS  037006          6125   6279   6291   6564   6647#  7877
CLRR1   037030          6653#  6654
CMDBUF  002666          2509#  3403   3410   4849   4856
CMODS   011474          2963#  6348*  6349*  6352*  6353   6363   7609*  7610*  7513*  7614   7624
CMPBUF  005150          2574#  4052*  4066   4792   5102
CMPPTR  006444          2584#  4060*  4061*  4074*  4075*  4076*  4080*  4787*  4799   4962*  4976*  4977   4981*
                        5090*  5091   5100*  5101   5366*  5677
CMPSEX  033440          5675   5763#
CMPSR   033164          5624   5674#
CMPS1   033324          5706#  5724
CMPS2   033330          5709#
CMPS3   033216          5680#  5736
CMPS5   033426          5733#
CMPS5A  033432          5726   5734#
CMPS6   033374          5707   5711   5722#
CMPS7   033312          5695   5704#
CMPTOT  006446          2585#  4069*  4072  4081*  4804*  4960*  4963   4982*  5093   5097*  5353   5369
CMSG0 = 000020          2302#  2879
CMSG1 = 000021          2303#  2877
CMSG2 = 000022          2304#  2881
CMSG3 = 000023          2305#  2883
CMSG4 = 000024          2306#  2887
CMSG5 = 000025          2307#  2885
CMSG6 = 000026          2308#  2889
CONOTM  006550          2625#  3106
CONTIN  006552          2626#
CPTR    006524          2615#  3967   3971*  4795*  4799*  4935*  4939   4977*  4981   5101*  5114*  5390*  5417*
                        5437*  5462*  5487*  5554   5559*  5578   5628*  5635   5676*  5681   5685*  5931*
CPTRR   006522          2614#  5414*  5440*  5463*  5489*  5542   5547*  5584*  5589   5598*  5611*  5632   5677*
                        5687   5690*  5879*
CR      016534          3018#  3344
CRC   = 000040          2318#
CRCB  = 000020          2224#  5306
CRCOK = 100000          2379#  7656
CSHEXP= 000006          2292#  2823
CSHTRN= 000007          2293#  2825
CSRADR  046361          7926   7956#
CTOTCC  006450          2586#  4068*  4076  4078  4079*  4084*  4782*  4947   4958   4984*  5098*
CTS   = 020000          2350#  2693   6618
CTSSR   036574          6074   6085   6240  6549  6588#  7850
CTSS3   036644          6595   6603#
```

```
CTSS4   036622          6597#   6601
CTSS7   036704          6605    6609    6611    6614#
CURADD  006526          2616#   3969    3974    3976    3986*   4794*   4800*   4936*   4941    4978*   4983    5119*   5389*
CURCC   006520          2613#   3921    3933    3936*   3937    3970    3975    3986    4790*   4802*   5121*   5193*   5206*
                        5234*   5237*   5239*   5388*
C$AU  = 000052          1945#   4739
C$AUTO= 000061          1945#   4660
C$BRK = 000022          1945#   6003    6057    6095    6259    6559    6599    6617    7137    7872
C$BSEG= 000004          1945#
C$BSUB= 000002          1945#
C$CEFG= 000045          1945#
C$CLCK= 000062          1945#   4510    4520
C$CLEA= 000012          1945#   4696
C$CLOS= 000035          1945#
C$CLP1= 000006          1945#
C$CVEC= 000036          1945#
C$DCLN= 000044          1945#   4478
C$DODU= 000051          1945#
C$DRPT= 000024          1945#
C$DU  = 000053          1945#   4717
C$EDIT= 000003          1945#   2009
C$ERDF= 000055          1945#
C$ERHR= 000056          1945#
C$ERRO= 000060          1945#
C$ERSF= 000054          1945#
C$ERSO= 000057          1945#   5698    5716    5729    5772    6014    6051    6118    6271    6393    6418    6631    7452
                        7470
C$ESCA= 000010          1945#
C$ESEG= 000005          1945#
C$ESUB= 000003          1945#
C$ETST= 000001          1945#   7890
C$EXIT= 000032          1945#   4636    4688    4837    4897
C$GETB= 000026          1945#
C$GETW= 000027          1945#
C$GMAN= 000043          1945#   3401    4539    4847    5857
C$GPHR= 000042          1945#   4574
C$GPLO= 000030          1945#
C$GPRI= 000040          1945#
C$INIT= 000011          1945#   4644
C$INLP= 000020          1945#
C$MANI= 000050          1945#   4831
C$MEM = 000031          1945#
C$MSG = 000023          1945#   3035    3048    3062    3076    3096    3116
C$OPEN= 000034          1945#
C$PNTB= 000014          1945#   3031    3044    3058    3072    3092    3112
C$PNTF= 000017          1945#   3347    3355    3393    3421    3431    3491    3550    3868    3879    3887    3931    4313
                        4356    4554    4772    4814    4867    4876    4914    4929    4955    4971    5071    5144    5189
                        5219    5293    5342    5359    5827    5924    5951    7289
C$PNTS= 000016          1945#   3536    3571    3585    3614    3637    3660    3674    3687    3700    3708    3735    3751
                        3766    3781    3789    3809    3829    4131    4167
C$PNTX= 000015          1945#
C$QIO = 000377          1945#
C$RDBU= 000007          1945#   4532
C$REFG= 000047          1945#   4483    4488    4493    4499
C$RESE= 000033          1945#   4686
C$REVI= 000003          1945#   2008
```

```
C$RFLA= 000021            1945#
C$RPT = 000025            1945#  4446
C$SEFG= 000046            1945#
C$SPRI= 000041            1945#  4634  4674
C$SVEC= 000037            1945#  4612  4622  4629  4821  6907
C$TPRI= 000013            1945#
DATABC= 000002            6766#  7377  7505  7510
DATCKB= 000002            2221#  4140  5300  5351  5436  5581  5630  5674  5851  5919
DCD   = 010000            2352#  2695  6608
DCK   = 000014            2242#  3301
DCLFLG  006562            2634#  4474  4476*  4895*
DDE   = 000022            2245#  3309
DER   = 000010            2240#  3287
DERIN   037072            6716#  7172*  7174*  7264*  7266*  7374*  7376*
DEROUT  037070            6710#  7498*  7500*  7507*  7509*
DEV1    007706            2753#  3693*  3704  3712*  3714*  4112  4824*  5149*
DEV2    007710            2754#  3694*  3703  3713*  3715*  4115  4825*  5150*
DEV3    007712            2755#  3716*  4122  4826*  5151*
DEV4    007714            2756#  3717*  4136  4140  4144  4149  4153  4827*  5152*
DFPTBL  002130 G          2074#
DIAGMC= 000000            1945
DIVN91  034744            6092#  6097
DLE   = 000020            2244#  3305
DLL     033566            2660  5822#
DLLCM   014122            3018#  5824
DLLMOD= 000033            2313#  2812
DMPE  = 000053            2329#  2842
DMPQ  = 000054            2330#  2844
DMPS  = 000052            2328#  2840  4889  5158
DMSGAD  002172            2405#  3978
DMSGCT  002150            2392#  3977
DOW   = 000004            2201#  5268  6033
DPLX    046330            7916  7956#
DPV   = 000000            2232#
DPVP1   011472            2953#  5380*  5383*  6061
DSITEN= 000040            2363#  6197  6545  7110
DSR   = 001000            2351#  2694
DTR   = 000002            2361#  6037
DUMEX   023262            3890  3896#
DUMPSR  023126            3861#  4891
DUM1    023220            3871  3882#
DUM2    023242            3881  3889#
DUM3    023156            3870#  3894
DUM4    023132            3862#  3893
DVEM0   016747            3018#  6008  6016
DVEM1   017026            3018#  6625  6633
DVEM2   017110            3018#  6265  6273  7464  7472
DVEM3   017205            3018#  6389  6395
DVEM4   017245            3018#  6412  6420
DVEM5   017311            3018#  6112  6120  7446  7454
DVEM6   017375            3018#  6045  6053
DVI   = 000012            2241#  3292
DVINEX  035102            6035  6063  6065  6067  6093  6105  6125#
DVINIT  034304            5386  5995#  6018  6123
DVIN1   034366            6001  6022#
DVIN3   034422            6025  6029#
```

```
DVIN3A  034446          6034    6037#   6055
DVIN31  036444          6076    6101    6539#
DVIN32  036474          6549#   6566
DVIN34  036572          6561    6567#
DVIN35  036536          6558#   6563
DVIN37  034554          6027    6030    6061#
DVIN38  034462          6040#   6060
DVIN39  034534          6041    6056#
DVIN4   034654          6080#
DVIN41  034634          6075#   6083
DVIN8   034766          6081    6100#   6108
DVIN81  035030          6078    6098    6103    6112#
DVIN9   034674          6085#   6107
DVMEX   035156          6154    6156#
DVMODS  035124          3324    6151#   6607*
DVRCC   006472          2597#   5546*   5574    5928*   5943    5945*   5946*   6190*   6430*   7711*
DVRCT   006474          2598#   5415*   5488*   5583*   5587*   5607*   5608    5678*   5735*
DVREX   035266          6182    6198#
DVRXA   006470          2596#   5544*   5573    5926*   5942    5945    6188    6428    7709
DVRXI   035554 G        4619    4818    6339#
DVRXQ   035166          5551    5937    6180#
DVRX2   035244          6186    6195#
DVRX3   035260          6191    6197#
DVTCC   006456          2590#   5558*   5615    5876*   5890    6245    6251    6502    7843
DVTCT   006460          2591#   5438*   5461*   5486*   5617*   5622    5625
DVTREX  035552          6281    6287    6292#
DVTR1   035344          6242    6249#
DVTR2   035374          6246    6253#
DVTR3   035410          6238    6256#   6275
DVTR4   035516          6278    6283#
DVTR5   037044          6651    6657#
DVTR8   035416          6258#   6284    6290
DVTXA   006454          2589#   5556*   5614    5873*   5889    6244    6501
DVTXI   036254 G        4626    6477#
DVTXRX  035270          5569    5888    5939    6237#
DVTXR1  037004          6619    6636#
DVTXR2  036720          6616#   6621
DVTXR3  036604          6594#   6635
DVTXR9  037004          6590    6637#
ECHO  = 000037          2317#   2858
ECHOB = 000004          2222#   4144    5276    5286    5296    5576
EDABO   015452          2726    3018#
EDDCK   015244          2721    3018#
EDDDE   015357          2724    3018#   5718    5731
EDDER   015227          2719    3018#
EDDLE   015322          2723    3018#   5700
EDDVI   015274          2720    3018#
EDEOP   015412          2725    3018#
EDMOS   015426          2722    3018#
EDRXC   015201          2718    3018#
EDRXQ   015154          2717    3018#
EDTXC   015125          2716    3018#
EDTXQ   015101          2715    3018#
EFM11   014701          3018#   3055
EFM13   015030          3018#   3069
EFM2    014604          3018#   3041
```

```
EF.CON= 000036 G        2153#   4492
EF.NEW= 000035 G        2154#   4498
EF.PWR= 000034 G        2155#
EF.RES= 000037 G        2152#   4487
EF.STA= 000040 G        2151#   4482
EMSG0   002215          2393    2417#
EMSG1   002216          2394    2419#
EMSG2   002217          2395    2421#
EMSG3   002220          2396    2423#
EMSG4   002320          2397    2437#
EMSG5   002412          2398    2451#
EMSG6   002513          2399    2465#
EMSG8   002643          2401    2479#
ENADD   006512          2606#   3889    5162*
ENDEVT  023004          3639    3649    3653    3796#
ENDHWL  046330          7939    7945    7952#
ENDIT   026116          4496    4604    4606#
ENDPS   033500          5766    5796#
ENDQQ = 000017          2301#   2895
ENQ   = 000005          6786#   7147    7200    7789    7808    7820    7830
EOP   = 000024          2246#   3313
ERRCNT  006506          2604#   3323    3326    4682    5375*   5696*   5714*   5727*   5770*   5800    6012*   6049*   6116*
                        6269*   6391*   6416*   6629*   7450*   7468*
ERREXT  044502          7458    7477    7502    7511    7520    7529    7538    7547    7553    7558    7562    7566#
ERRPRC  043726          7142    7156    7162    7175    7211    7268    7309    7356    7364    7378    7438#
ERR1    017462 G        3021#   5719
ERR10   017552 G        3051#   5701
ERR13   017634 G        3085#   6017    6054    6121    6274    6396    6421    6634    7455    7473
ERR14   017666 G        3104#
ERR2    017524 G        3038#   5732
ERR4    017602 G        3065#   5775
ERX   = 000100          2255#   5416    5464    5490    5603    5621    5930    6280
ETX   = 000200          2256#   5439    5490    5604    5620    5878    5896    6286
EVL   = 000004 G        2171#
EVMCTS  016477          2704    3018#
EVMDCD  016507          2706    3018#
EVMDSR  016503          2705    3018#
EVMOCG  016357          3018#   3786
EVMOHD  016402          3018#   3806
EVMOST  016462          3018#   3826
EVMRI   016517          2708    3018#
EVMRTS  016513          2707    3018#
EVMSQD  016523          2709    3018#
EVMTM   016527          2710    3018#
EVTADD  007650          2733#   3678*   3683    3725*   3731    3741*   3747    3756*   3762
EVTBCT  007652          2734#   3679*   3682    3726*   3730    3742*   3746    3757*   3761
EVTEND  007552          2683#   3368    3646    4563
EVTF0   015547          3018#   3657
EVTF1   015643          3018#   3671
EVTF2   015672          3018#   3684
EVTF3   015744          3018#   3697    3778
EVTF3C  015756          3018#   3739    3705
EVTF3D  015773          3018#   3709
EVTF4   016015          3018#   3748
EVTF4A  016117          3018#   3763
EVTF4B  016215          3018#   3732
```

```
EVTF5A  016274            3018#   3028
EVTLOG  006650            2681    2682#   3372    3631    3644    4560
EVTLST  007612            2715#   3667
EVTMIN  007644            2731#   3665*   3670
EVTPTR  006646            2681#   3358    3373*   3630    3651    4561*
EVTSEC  007642            2730#   3664*   3669
EVTTCK  007646            2732#   3663*   3668
EVTTMP  007654            2735#   3692*   3696    3727*   3729    3743*   3745    3758*   3760    3775*   3777
EXIT  = 000057            2333#   2784    4675    4893    5078
EXITCL  026314            4676    4685#
E$END = 002100            1945#
E$LOAD= 000035            1945#   2032
FACSIM  023506            4048#   5247
FHDPLX  006572            2642#   4579*   6064    6604    6650    6910    6988    7238    7763
FIR     037250            3508*   3514*   3528*   3553    6860#
FIRST = 004000            2269#   6594    6602    7605    7851
FLAG    006600            2653#   5372*   5416*   5439*   5464*   5490*   5540    5552    5571    5603*   5604*   5610*   5612
                          5620*   5621*   5629*   5878*   5884*   5896*   5930*   6022*   6072*   6086*   6092    6126*   6127*
                          6181    6184*   6187*   6237    6239*   6243*   6277    6280    6283    6286    6289    6344    6382
                          6400*   6424    6426*   6427*   6433*   6485    6489*   6498    6500*   6504*   6539*   6550*   6562
                          6565    6594    6602*   6893    7062*   7770*   7788*   7807*   7819*   7829*   7841*
FMTMSK= 000002            6778#   7546
FORMER= 000021            6771#   7308    7541
F$AU  = 000015            1945#   4727    4738
F$AUTO= 000020            1945#   4655    4659
F$BGN = 000040            1945#   1947    3021    3038    3051    3065    3085    3104    3121    3203    4435    4455    4471
                          4636    4655    4669    4688    4705    4711    4727    4733    4754    4837    4897    6339    6477
                          7598    7889    7908    7993
F$CLEA= 000007            1945#   4669    4695
F$DU  = 000016            1945#   4705    4716
F$END = 000041            1945#   1947    3036    3049    3063    3077    3097    3117    3119    3230    4447    4636    4645
                          4661    4688    4697    4709    4718    4731    4740    4754    4837    4897    6441    6510    7725
                          7889    7891    7955    7993
F$HARD= 000004            1945#   7908    7939    7945    7953
F$HW  = 000013            1945#   2072    2100
F$INIT= 000006            1945#   4471    4643
F$JMP = 000050            1945#   3119    4636    4688    4709    4731    4837    4897
F$MOD = 000000            1945#   1947    7993
F$MSG = 000011            1945#   3021    3034    3038    3047    3051    3061    3065    3075    3085    3095    3104    3115
F$PROT= 000021            1945#   4455    4462
F$PWR = 000017            1945#
F$RPT = 000012            1945#   4435    4445
F$SEG = 000003            1945#
F$SOFT= 000005            1945#
F$SRV = 000010            1945#   3203    3228    6339    6439    6477    6508    7598    7723
F$SUB = 000002            1945#
F$SW  = 000014            1945#
F$TEST= 000001            1945#   4755    7889
GETCL   026762            4833    4844#   4869    4878    4881    4883    4885    4892    4901    4903    4916    4931    4945
                          4957    4973    4987    5361
GETPRM  025712            4502    4568#   4577
GETRCL  020636            3396#   3423    3433    3437
GLINC   035704            6356    6360#
GLMSG   016671            3018#   6361    7622
GLOBCC  037063            6702#   7193*
GOOD    006554            2627#   3026    5712*
```

```
GTRA2   026346          4762#
GTRA3   026410          4765    4775#
GTRA4   026416          4780#
GTRA5   026642          4776    4816#   5567    5809    5829
GTREX   032036          5372#
GTRX2   032140          5388#   5807    5898    5958
GTR9    031734          4840    4888    5349#
GTXRXB  026346          4761#
G$CNTO= 000200          1945#
G$DELM= 000372          1945#
G$DISP= 000003          1945#
G$EXCP= 000400          1945#
G$HILI= 000002          1945#
G$LOLI= 000001          1945#
G$NO  = 000000          1945#   3404    4850    5860
G$OFFS= 000400          1945#   3404    4542    4850    5860    7915    7925    7930    7935    7941    7947
G$OFSI= 000376          1945#   3404    4542    4850    5860    7915    7925    7930    7935    7941    7947
G$PRMA= 000001          1945#   7925    7930
G$PRMD= 000002          1945#   3404    4542    4850    5860    7947
G$PRML= 000000          1945#   7915    7935    7941
G$RADA= 000140          1945#   3404    4850    5860
G$RADB= 000000          1945#
G$RADD= 000040          1945#   4542    7947
G$RADL= 000120          1945#   7915    7935    7941
G$RADO= 000020          1945#   7925    7930
G$XFER= 000004          1945#   7939    7945
G$YES = 000010          1945#   4542    7915    7925    7930    7935    7941    7947
HDMADR  002652          2493#   7769*
HDMC    002654          2496#   6089    6196    6252    6542    6552    7105
HDMCC   002646          2490#   6075*   6087*   6100*   6114    6251*   7448    7843*   7844*
HDMID   002645          2488#   7789*   7808*   7820*   7830*   7842*   7854
HDMNUM  002651          2492#   7801*   7812*   7824*   7834*   7846*
HDMREP  002650          2491#   7800*   7813*   7823*   7833*   7845*
HDMSG   002644          2487#   6071*   6088    6249*   6250    6551
HDMTYP  002646          2489#   7790*   7791*   7798*   7809*   7810*   7811*   7821*   7822*   7831*   7832*
HDXMTP  037136          6756#   6909*   6914*   6934    7225    7406    7755    7792    7865
HEADBC= 000001          6765#   7170    7171    7496    7501
HELP  = 000000            1#    1945    1959    2050    2062    2076    2104    2108    2115    2976    2999    3013    3018
                       3130    4437    4441    4443    4463    4473    4639    4657    4671    4691    4707    4712    4729
                       4734    4751    4752    4756    7885    7892    7911    7959    7973    7978    7986
HELPDC= 000000            1#    1961    2085    2103    2231    2258    2344    2358    2937    2977    2992    3018    3079
                       3317    3627    3773    4581    4617    5821    5963    5968    6133    6161    6202    6296    6301
                       6442    7920    7956
HLP   = 000005          2291#   2778    2780    4882    5075
HLPEND  003036          2522#   5073
HLPF    012273          3018#   3488    5068
HLPTAB  003016          2514#   5065
HLP0    012215          3018#   4811
HLP1    012300          2514    3018#
HLP2    012313          2515    3018#
HLP3    012430          2516    3018#
HLP3A   012515          2517    3018#
HLP4    012542          2518    3018#
HLP4A   012621          2519    3018#
HLP5    012677          2520    3018#
HLP6    012767          2521    3018#
```

```
HOE    = 100000 G      2184#
HRDMSG  016717         3018#  6358   7619
HSMSE   002653         2494#
IBE    = 010000 G      2181#
IDU    = 000040 G      2174#
IER    = 020000 G      2182#
IMFLAG  037112         6746#  6893*  6943   6954   6990   7006   7025   7030   7062   7603   7605   7851*
IMPRST  037132         6754#  7745*  7768
ININT  = 000001        2251#
INIT1   025370         4475   4480#
INMASK  037073         6717#  7170*  7267*  7377*
INOVR  = 002000        2267#  6127   6344   7603
INTPRI  011470         2952#  4599*
INVEC   011464         2950#  4596*  4620   4819   6905
IRDSR   011500         2965#  6369*  6376   6384   6387   6413   7631*  7632   7648   7656
IRXCSR  011476         2964#  6058*  6059   6341*  6346   6348   6367   6370   6388   6407   6414   7600*  7607
                       7609   7628
ISR    = 000100 G      2175#
IXE    = 004000 G      2180#
I$AU   = 000041        1945#  4727#  4740#
I$AUTO= 000041         1945#  4655#  4661#
I$CLN  = 000041        1945#  4669#  4688   4697#
I$DU   = 000041        1945#  4705#  4718#
I$HRD  = 000041        7908#  7955#
I$INIT= 000041         1945#  4471#  4636   4645#
I$MOD  = 000041        1945#  1947#  7993#
I$MSG  = 000041        1945#  3021#  3036#  3038#  3049#  3051#  3063#  3065#  3077#  3085#  3097#  3104#  3117#
I$PROT= 000040         1945#  4455#
I$PTAB= 000041         1945#
I$PWR  = 000041        1945#
I$RPT  = 000041        1945#  4435#  4447#
I$SEG  = 000041        1945#  4754
I$SETU= 000041         1945#
I$SRV  = 000041        1945#  3203#  3230#  6339#  6441#  6477#  6510#  7598#  7725#
I$SUB  = 000041        1945#  4754
I$TST  = 000041        1945#  4754#  4837   4897   7889#  7891#
J$JMP  = 000167        1945#  3119   4709   4731
KEYWD1  003010         2510#  3435   3495*  3499*  3504*  4473*  4675   4880   4882   4884   4886   4889   4893
                       4900   4902   4904   5059*  5062*  5075*  5078*  5081*  5085*  5095   5109   5158*  5168*
                       5171*  5243   5248*
LAST    037246         3509*  3515*  3529*  3558   6859#
LBMASK  037075         6723#  7692*  7704*
LBUFER  037074         6722#  7691*  7701*  7703*
LCLKEN= 000100         2214#  4515
LGDVE   020160         3285#  6011   6048   6115   6268   6390   6415   6628   7449   7467
LIS    = 000006        2203#  5265   5897
LISCK   034056         2662   5919#
LISCKA  034104         5926#  5954   5956
LISMOD= 000032         2312#  2810
LISP    014045         3018#  5921
LMDLOP= 000046         2324#  2923
LNCNT   006500         2601#  3339   3341*  3350*  5378*
LOE    = 040000 G      2183#
LOGCMD  020300         3307#  5733
LOGCML  020262         3303#  5702
LOGCMP  020244         3299#  5692
```

```
LOGDVI  020176            3290#  5385
LOGEOP  020316            3311#  5801
LOGEX   020606            3328   3375#
LOGMSC  020334            3317#  6362   7623
LOGRXC  020150            3282#  5575   5944
LOGRXQ  020132            3277#  5548   5938
LOGS1   020352            3270   3275   3280   3284   3323#
LOGS2   020600            3369   3373#
LOGS3   020404            3288   3297   3302   3306   3310   3314   3320   3332#
LOGS4   020460            3340   3349#
LOGS5   020504            3334   3336   3357#  4683
LOGTXC  020114            3272#  5616   5891
LOGTXQ  020076            3267#  5560   5877
LOGUNT  006556            2632#  4566*  4568*  4569   4573
LOOPS   003120            2553#  4121
LOSTER  037102            6737#  7307*  7636*
LOT     = 000010 G        2172#
LP0     013714            2553   3018#  4120
LP00    013715            3018#  4117
LP1     013724            2554   3018#
LP2     013735            2555   3018#
LP3     013743            2556   3018#
LP4     013756            2557   3018#
LSMASK  037103            6738#  7308*  7637*
L$ACP   002110 G          2039#
L$APT   002036 G          1997#
L$AU    026332 G          2024   4727#
L$AUT   002070 G          2023#
L$AUTO  026234 G          2040   4655#
L$CCP   002106 G          2037#
L$CLEA  026236 G          2038   4669#
L$CO    002032 G          1993#
L$DEPO  002011 G          1975#
L$DESC  011536 G          2030   3004#
L$DESP  002076 G          2029#
L$DEVP  002060 G          2015#
L$DISP  002124 G          2000   2059#
L$DLY   002116 G          2045#
L$DTP   002040 G          1999#
L$DTVP  002034 G          1995#
L$DU    026324 G          2026   4705#
L$DUT   002072 G          2025#
L$DVTY  011526 G          2016   2994#
L$EF    002052 G          2010#
L$ENVI  002044 G          2003#
L$ETP   002102 G          2033#
L$EXP1  002046 G          2005#
L$EXP4  002064 G          2019#
L$EXP5  002066 G          2021#
L$HARD  046250 G          1982   7908   7909#
L$HIME  002120 G          2047#
L$HPCP  002016 G          1981#
L$HPTP  002022 G          1985#
L$HW    002130 G          1986   2072   2073#
L$ICP   002104 G          2035#
L$INIT  025350 G          2036   4471#
```

```
L$LADP  002026 G        1989#
L$LAST  046636 G        1990    7991#
L$LOAD  002100 G        2031#
L$LUN   002074 G        2027#
L$MREV  002050 G        2007#
L$NAME  002000 G        1964#
L$PRIO  002042 G        2001#
L$PROT  025342 G        2042    4455#
L$PRT   002112 G        2041#
L$REPP  002062 G        2017#
L$REV   002010 G        1973#
L$RPT   025334 G        2018    4435#
L$SPC   002056 G        2013#
L$SPCP  002020 G        1983#
L$SPTP  002024 G        1987#
L$STA   002030 G        1991#
L$TEST  002114 G        2043#
L$TIML  002014 G        1979#
L$UNIT  002012 G        1977#   4569
L10000  002150          2072    2100#
L10001  017522          3034#
L10002  017550          3047#
L10003  017600          3061#
L10004  017632          3075#
L10005  017664          3095#
L10006  017722          3115#   3120
L10007  020074          3228#
L10010  025340          4445#
L10012  026232          4637    4643#
L10013  026234          4659#
L10014  026322          4689    4695#
L10015  026330          4710    4716#
L10016  026336          4732    4738#
L10017  046244          4838    4898    7889#
L10020  036252          6439#
L10021  036442          6508#
L10022  045346          7723#
L10023  046330          7908    7954#
L5060   014063          3018#   4543
MAINT =  000144         6785#
MCREP   033454          5764    5770#
MES     037252          3554*   3568    3582    3611    6861#
MESDAT  037254          3555*   3567    3579    3581    3593    3596    3599    3610    6862#
MESLON=  000020         6770#   7532    7693
MGLCNT  011520          2973#   3067    5376*   5763    5776*   6360*   7621*
MHRCNT  011522          2974#   3068    5377*   5765    5777*   6357*   7618*
MLTYP   006570          2639#   3294    4806*   4825    4839*   5150    5259*   5278*   5324*   5326*   5328*   5330*   5332*
                        5337*   5349    6024    6029    6588
MOBITE  007574          2700#   3823
MOBITS  007556          2693#   3811
MOCHK =  000010         2223#   4153    5309    6342    7106    7601
MODE    006602          2656#   5395
MODES   003102          2545#   4114
MODLOC=  000003         2208#
MODREM=  000004         2209#
MODS    007554          2687#   3330    6151*   6152*   6155*   6156*   6354    6363*   6608    7615    7624*
```

```
MODTYP  006566          2636# 3293  4805* 4824  5149  5254* 5257* 5262* 5265* 5268* 5271* 5274* 5287
                        5334  5393  5585  5601  5618  5897* 5957* 6033  6938  7234  7408  7748  7867
MOMSGS  007574          2704# 3812
MOP   = 000043          2210# 2321#
MOSC  = 000056          2332# 2867
MO0     013623          2545  3018#
MO1     013633          2546  3018#
MO2     013644          2547  3018#
MO3     013654          2548  3018#
MO4     013663          2549  3018#
MO5     013700          2550  3018#
MO6     013705          2551  3018#
MPPTP   037126          4601* 6648  6752# 6912  6923
MSC   = 000016          2243# 3319
MSCMS   014756          3018# 5774
MSGCC   011504          2967# 6089* 6245* 6252* 6495* 6502* 6552* 7855*
MSGLIM= 000017          2190# 2575  2576  2577  4057  4922  4964
MSGPTR  011502          2966# 6088* 6244* 6250* 6492  6494* 6501* 6551* 7854*
MSGTRN  014365          3018# 4911  4926  4952  4968
MSGTRU  014416          3018# 3928
MSGTYP  006516          2611# 3972  4789* 4801* 5120* 5199* 5223* 5225* 5227* 5229* 5231* 5233* 5236*
                        5391*
MSG0    002214          2393  2406  2416#
MSG0C   002150          2393#
MSG1    002215          2394  2407  2418#
MSG1C   002152          2394#
MSG2    002216          2395  2408  2420#
MSG2C   002154          2395#
MSG3    002217          2396  2409  2422#
MSG3C   002156          2396#
MSG4    002220          2397  2410  2424#
MSG4C   002160          2397#
MSG5    002320          2398  2411  2438#
MSG5C   002162          2398# 4790  4802  5121  5234
MSG6    002412          2399  2412  2452#
MSG6C   002164          2399# 5237
MSG8    002642          2401  2414  2478#
MSG8C   002170          2401#
MTLMSK= 000002          6779# 7537  7692
MYDATA= 001000          6807# 7103  7185  7321  7338  7645  7670
N       037054          6689# 7229* 7327* 7396*
NAK   = 000002          6792# 7205  7809
NAKRX = 000400          6806# 7210  7481
NEW     025704          4501  4566# 4570
NO    = 000036          2316# 2850  5281  5284  5315
NOCLK   014226          3018# 4769
NOD0    007716          2777#
NOD1    007722          2778#
NOD10   010000          2785#
NOD100  010626          2857#
NOD101  010632          2858#
NOD102  010646          2859#
NOD103  010652          2861#
NOD104  010656          2862#
NOD105  010674          2863#
NOD106  010700          2864#
```

```
NOD107  010712          2865#
NOD11   010002          2786#
NOD110  010716          2867#
NOD111  010732          2868#
NOD112  010736          2870#
NOD113  010752          2871#
NOD114  010756          2874#
NOD115  010762          2877#
NOD116  010776          2878#
NOD117  011002          2879#
NOD12   010014          2787#
NOD120  011020          2880#
NOD121  011024          2881#
NOD122  011040          2882#
NOD123  011044          2883#
NOD124  011060          2884#
NOD125  011064          2885#
NOD126  011100          2886#
NOD127  011104          2887#
NOD13   010020          2788#
NOD130  011120          2888#
NOD131  011124          2889#
NOD132  011140          2890#
NOD133  011144          2891#
NOD134  011164          2892#
NOD135  011170          2894#
NOD136  011174          2895#
NOD137  011200          2896#
NOD14   010034          2789#
NOD140  011204          2897#
NOD141  011210          2898#
NOD142  011214          2899#
NOD143  011220          2900#
NOD144  011222          2903#
NOD145  011226          2904#
NOD146  011232          2905#
NOD147  011246          2906#
NOD15   010040          2790#
NOD150  011252          2907#
NOD151  011266          2908#
NOD152  011272          2911#
NOD153  011276          2912#
NOD154  011302          2913#
NOD155  011306          2916#
NOD156  011312          2919#
NOD157  011334          2920#
NOD16   010054          2791#
NOD160  011340          2921#
NOD161  011354          2922#
NOD162  011360          2923#
NOD163  011402          2924#
NOD164  011406          2925#
NOD165  011430          2926#
NOD166  011434          2929#
NOD167  011440          2930#
NOD17   010060          2792#
```

```
NOD170  011444            2931#
NOD171  011450            2936#
NOD172  021030            3444#
NOD173  021034            3445#
NOD174  021040            3446#
NOD175  021042            3447#
NOD176  021056            3448#
NOD177  021060            3449#
NOD2    007726            2779#
NOD20   010064            2793#
NOD200  021074            3450#
NOD201  021076            3451#
NOD202  021110            3452#
NOD203  021112            3453#
NOD204  021132            3454#
NOD205  021136            3455#
NOD206  021142            3456#
NOD207  021156            3457#
NOD21   010076            2794#
NOD210  021160            3458#
NOD211  021174            3459#
NOD212  021176            3460#
NOD213  021214            3461#
NOD214  021220            3462#
NOD215  021224            3463#
NOD216  021226            3464#
NOD217  021230            3465#
NOD22   010102            2795#
NOD23   010114            2796#
NOD24   010120            2797#
NOD25   010122            2801#
NOD26   010126            2802#
NOD27   010142            2803#
NOD3    007730            2780#
NOD30   010146            2804#
NOD31   010164            2805#
NOD32   010170            2806#
NOD33   010206            2807#
NOD34   010212            2808#
NOD35   010230            2809#
NOD36   010234            2810#
NOD37   010252            2811#
NOD4    007744            2781#
NOD40   010256            2812#
NOD41   010302            2813#
NOD42   010306            2814#
NOD43   010312            2815#
NOD44   010330            2816#
NOD45   010334            2817#
NOD46   010346            2818#
NOD47   010352            2822#
NOD5    007746            2782#
NOD50   010356            2823#
NOD51   010400            2824#
NOD52   010402            2825#
NOD53   010426            2826#
```

```
NOD54   010430          2831#
NOD55   010434          2832#
NOD56   010452          2833#
NOD57   010456          2834#
NOD6    007762          2783#
NOD60   010476          2835#
NOD61   010502          2838#
NOD62   010506          2839#
NOD63   010512          2840#
NOD64   010516          2841#
NOD65   010522          2842#
NOD66   010526          2843#
NOD67   010532          2844#
NOD7    007764          2784#
NOD70   010536          2845#
NOD71   010542          2848#
NOD72   010546          2849#
NOD73   010552          2850#
NOD74   010564          2851#
NOD75   010570          2852#
NOD76   010604          2853#
NOD77   010610          2856#
NONE  = 000000          2205#
NOTNUF= 000050          2326#  2788  2791  2792  2802  2841  2843  2849  2904
NULEVT  015507          3018#  3634
NULL  = 000000          2286#
NUM   = 000014          2298#  2912
N10$    007722          2777#
N100$   010352          2791   2794  2821#
N102$   010356          2822#
N104$   010402          2823   2824#
N110$   010430          2796   2830#
N111$   010434          2831#
N112$   010456          2832   2833#
N114$   010546          2848#
N115$   010542          2805   2807  2809  2811  2813  2816  2818  2847#  2853  2857  2859  2863  2868
                        2920   2922  2924  2926  2931
N116$   010564          2850#
N117$   010610          2851   2852  2855#
N118$   010632          2856   2857#
N119$   010652          2858   2860#
N120$   010756          2833   2835  2873#
N121$   011222          2878   2880  2882  2884  2886  2888  2890  2896  2902#  2913
N122$   011252          2905   2906#
N123$   011226          2903#
N125$   011450          2841   2843  2845  2849  2892  2904  2935#
N126$   011272          2906   2908  2910#
N130$   010700          2862   2863#
N131$   010736          2867   2869#
N132$   010716          2861   2866#
N140$   011306          2871   2915#
N141$   011312          2918#
N142$   011340          2919   2920#
N143$   011360          2921   2922#
N144$   011406          2923   2924#
N150$   011434          2865   2928#
```

```
N20$    010060          2790    2791#
N25$    010102          2793    2794#
N30$    010120          2792    2795    2796#   2801    2802    2803    2814    2817    2822    2825    2831    2834    2840
                        2842    2844    2864    2870    2874    2894    2907    2911    2912    2916    2925    2929    2930
N40$    010040          2788    2789#
N42$    007730          2778    2779#
N43$    007746          2780    2781#
N44$    007764          2782    2783#
N45$    010002          2784    2785#
N46$    010020          2786    2787#
N50$    010502          2789    2837#
N51$    010506          2838#
N52$    010512          2839#
N60$    010762          2876#
N61$    011002          2877    2878#
N62$    011024          2879    2880#
N63$    011044          2881    2882#
N64$    011064          2883    2884#
N65$    011104          2885    2886#
N66$    011124          2887    2888#
N67$    011144          2889    2890#
N68$    011170          2891    2893#
N70$    011174          2894#   2899
N71$    011204          2895    2896#
N72$    011210          2897#
N73$    011220          2898    2899#
N80$    010122          2787    2800#
N81$    010126          2801#
N82$    010170          2804    2805#
N83$    010212          2806    2807#
N84$    010234          2808    2809#
N85$    010256          2810    2811#
N86$    010306          2812    2813#
N87$    010334          2815    2816#
OFSET   006532          2618#   3027    5705*   5722*
OPBFPT  002514          2471#   5948
OPBUF   002520          2413    2472#   5209    5852    5859    5867    5873    5874    5892    5894    5926    5927    5953
                        5955
OPCNT   002166          2400#   5207*   5871*   5875    5876
OPEND   002642          2473#   5854
OPRMM   014056          3018#   5861
OPRMSG= 000015          2299#   2894
OPVAR   006502          2602#   4680    5373*   5798    6122*   7456*
OTINT = 000002          2252#
OUTMAS  037071          6711#   7501*   7510*
OUTVEC  011466          2951#   4597*   4598*   4627
O$APTS= 000000          1945#   1991
O$AU  = 000001          1945#   1958#   2023
O$BGNR= 000001          1945#   1958#   2017
O$BGNS= 000000          1945#   1983
O$DU  = 000001          1945#   1958#   2025
O$ERRT= 000000          1945#   2033
O$GNSW= 000000          1945#   1987
O$POIN= 000001          1945#   1958#   2049
O$SETU= 000000          1945#   1977    7989
PAD   = 001000          2265#   6086    6239    6485    6504    6550    7770
```

```
PARAM   006574          2643#  3296   3335   3544   4808*  4823*  4827   5152   5276*  5286*  5296*  5314*  5317*
                        5351   5436*  5549   5562   5565   5576   5581   5630   5674   5851*  5882   5919*  5934
                        6066   6342   6898   6928   7106   7207   7219   7221*  7249   7275   7282*  7301*  7384
                        7391*  7440   7601   7746   7796
PAS    = 000002         2199#  5257   5287   5601   5618
PASC   = 000042         2320#  2930
PASMOD= 000030          2310#  2806
PCADD   006560          2633#
PCK     014005          3018#  4139
PCLKCT= 001600          2216#  4526
PCLKEN= 000111          2215#  4528
PCPM    014647          3018#
PCSAR   011454          2944#  4587*  4588*  6061*
PEC     014015          3018#  4143
PLCK    032300          2658   5460#
PLCK2   032300          5461#
PLCK3   032314          5463#
PMS     014024          3018#  4152
PNCK    014003          3018#  4142
PNEC    014013          3018#  4146
PNMS    014022          3018#  4155
PNPR    014032          3018#  4151
PNST    013772          3018#  4138
PNT    = 001000 G       2178#
PPR     014034          3018#  4148
PRFLAG  037134          6755#  7102*  7149*  7654   7662*  7677   7682*  7683*  7715*
PRI    = 002000 G       2179#
PRIBB   021666          3577#  6845   6846   6847   6848   6849   6850
PRIBS   021724          3590#  6851   6852   6853   6854   6855   6856
PRINEX  045344          7630   7653   7665   7712   7721#
PRIN1   044656          7620   7623#
PRIN2   044604          7612   7614#
PRIN21  044670          7602   7604   7606   7608   7628#
PRIN4   044752          7633   7645#
PRIN5   045060          7655   7667#
PRIN6   045034          7657   7662#
PRIN7   045316          7678   7707   7715#
PRIN8   045332          7639   7659   7695   7718#
PRIW    021640          3566#  6842   6843   6844   6857
PRI00 = 000000 G        2167#  4633
PRI01 = 000040 G        2166#
PRI02 = 000100 G        2165#
PRI03 = 000140 G        2164#
PRI04 = 000200 G        2163#  4618   4625   4817   6903
PRI05 = 000240 G        2162#
PRI06 = 000300 G        2161#
PRI07 = 000340 G        2002   2160#  4673
PRNT   = 000055         2331#  2782   4884   5081
PROEND  037110          6745#  7083
PROFDX  041522          6989   7025#
PROHDX  041330          6935   6987#
PROINT  041730          6922   7079#
PRORUN= 000100          2226#  4823   6898   6928   7207   7219   7221   7249   7275   7301   7384   7391   7440
                        7746   7796
PROTEX  041670          6956   6962   6982   7008   7021   7056#
PROTO = 000041          2319#  2862
```

```
PROTOB=  000040          2225#  3544   4149   5312   5549   5562   5882   5934   6066
PROTOC   040702          5564*  5885*  5941*  6893#
PRRXI    044504 G        6904   7598#
PRSTAT   037046          3555   6673#  6921*  6925*  6945*  6968*  6997*  7015*  7048*  7081   7100*  7103*  7152
                         7160   7167   7169*  7185*  7186   7191   7210*  7212*  7254*  7262*  7277*  7299*  7306*
                         7321*  7338*  7339   7345*  7350   7360   7369   7373*  7401*  7481   7552   7635*  7645
                         7658*  7667*  7670*  7681*  7694*  7706*  7716*  7745   7768*  7773   7775   7777   7779
                         7781   7802*  7814*  7825*  7836*
PRTEMP   037140          6757#
PRUN     037124          6751#  6919*  6930*  7404
PSCNT    006504          2603#  4681   5374*  5796*  5799
PST      013774          3018#  4135
PTPMLP   046466          7942   7956#
PTREND   006434          2578#
PTRTAB   006150          2575#  4060   4065   4783   4920   5106   5113   5365
PTR13    006244          2576#  4787   4962   5090   5100   5366
PTR23    006340          2577#  5367
PSACT    003136          2563#  3412*  4236   4858*
PSBUFA   003132          2561#  3410*  4191   4856*
PSCNT    003140          2564#  4333*  4340   4402*  4409*  4414
PSEXIT   024362          4194   4229#  4253   4258
PSGDBD   003147          2568#  3396*  3415   3538*  4251*  4256*  4264   4315*  4358*  4844*  4861   5191*  5245*
                         5295*  5336*
PSNNUF   003146          2567#  3397*  3425   3481*  3521*  4845*  4870   5056*  5086*  5094*  5108*  5163*  5251*
                         5258*  5277*  5344*
PSNUM    003142          2565#  3522   3524   3526   3528   3529   4328*  4331*  4333   4334*  4336*  4340*  4342*
                         4348*  5156   5162   5183   5193   5195   5321
PSRADX   003144          2566#  4288*  4291*  4298*  4305   4338   4346
PSTREE   003134          2562#  3411*  4192   4857*
PSTRV    024244          3414   4190#  4860
PSTR5    024254          4193#  4202   4222   4227
QCOPY  = 000013          2297#  2907   5178
QRX    = 000004          2253#  5416   5464   5490   5540   5571   5603   5610   5621   5930   6181   6184   6283
                         6433   6539   6565   6954   7006   7030   7102   7715
QTX    = 000010          2254#  5439   5490   5552   5604   5612   5620   5629   5878   5896   6086   6092   6237
                         6239   6277   6289   6489   6550   6562   6943   6990   7025
QUALFG   003012          2512#  3413*  4859*  5175*  5178*  5181   5281*  5284   5297*  5315   5318*
QUALVL   003014          2513#  4943*  4985*  5172*  5195*  5196*
R        037060          6697#  7252   7334*  7335   7337*  7800   7813   7845
RBMASK   037077          6728#  7519*  7537*
RBUFER   037076          6727#  7516*  7518*  7534*  7536*
RDATRY=  000200          2365#  6367   6370   7628
RDSR     011456          2945#  4589*  4590*  6369   7031
REANAK   037064          6704#  7171*  7263*  7638*  7693*  7705*  7810
REC    = 000000          2197#
RECMOD=  000031          2311#  2808
REMTMO   037062          6701#  7255*  7257*
REP    = 000003          6793#  7245
REPLOG   022056          3503   3623#
REPMSK=  000004          6776#  7267
REPORT   020610          3384#  4438   5082
REPSEN=  000003          6767#  7263
RERR   = 100000          2377#  6384
RESET  = 000001          2368#  5998
RESFLG   006564          2635#  4480*  4504*  4775
RESTRT   025644          4490   4516   4529   4548   4557#
```

```
RHDADR  002664            2504#  7181  7668
RHDMCC  002660            2501#  6080  6082  6104  6106  6113  6429  6430  7447  7673  7684*  7687  7710
                          7711
RHDMID  002657            2499#  7104  7138  7147  7200
RHDMSG  002656            2498#  6195  6541  7101*
RHDNUM  002663            2503#  7252  7335
RHDREP  002662            2502#  7223  7231  7325  7394
RHDTYP  002660            2500#  7205  7217  7245  7273  7297  7495*  7496  7505  7514  7523  7532  7541
RHLPEN  003052            2530#  3493
RHLPTB  003036            2524#  3485
RHLP0   013066            3018#  3390
RHLP1   013124            2524   3018#
RHLP2   013146            2525   3018#
RHLP3   013201            2526   3018#
RHLP4   013232            2527   3018#
RHLP5   013264            2528   3018#
RHLP6   013327            2529   3018#
RI    = 040000            2354#  2697
RINTEN= 000100            2364#  5808  6197  6435  6545  7108  7110  7719  7766
RMDLOP= 000047            2325#  2925
RMMASK  037101            6733#  7528*  7546*
RMSGCC  011514            2971#  6189*  6196*  6380*  6401*  6429*  6542*  7105*  7652*  7663*  7710*
RMSGPT  011512            2970#  6188*  6195*  6375  6377*  6402*  6428*  6541*  7104*  7647  7649*  7664*  7709*
RMSTER  037100            6732#  7525*  7527*  7543*  7545*
RNODE   011524            2975#  4600*  5381  6062  6185  6241
RNODM   046442            7936  7956#
RNOTNF= 000006            2341#  3453  3455  3460
RPASS   006576            2652#  3295  4807*  4826  4834  5087*  5151  5321*  5803  5805*
RPERR = 000004            2339#  3456
RPEXT = 000002            2337#  3435  3449  3499
RPFUL = 000005            2340#  3458
RPHLP = 000001            2336#  3445  3447  3495
RPLOG = 000003            2338#  3451  3504
RPSWO = 000007            2342#  3462
RPT     022124            3641#  3690  3710  3721  3738  3753  3769  3793
RPTAA   022174            3656#
RPTABO  022510            2750   3724#
RPTDCK  022640            2745   3756#
RPTDDE  022564            2748   3741#
RPTDER  022350            2743   3692#
RPTDLE  022640            2747   3755#
RPTDSP  007656            2739#  3676
RPTDVI  022444            2744   3712#
RPTEOP  022510            2749   3725#
RPTIV   013376            3018#  3533
RPTMSB  023014            3689   3784  3792  3805#
RPTMSC  022714            2746   3775#
RPTMSE  023000            3793#
RPTNV   013421            3018#  3547
RPTTXQ  022276            2739   2740  2741  2742  3678#
RPT0    022170            3632   3648  3652  3655#
RPT1    022156            3645   3651#
RSTARY= 002000            2366#  6367  6407  7628
RTS   = 000004            2353#  2696  6614  6656
RUN   = 000004            2290#  2786  4886  5085
RXACT = 004000            2367#
```

```
RXBUF   004150          2573#   5389
RXC   = 000006          2239#   3283    3333
RXCSR   011452          2943#   4584*   5808*   6000    6009    6037*   6046    6058    6151    6197*   6266    6341    6435*
                        6545*   6614*   6618    6626    6656*   7108*   7110*   7465    7600    7719*   7766*
RXD   = 000100          6805#   7186    7339    7350    7716
RXENA = 000020          2362#   5808    6197    6435    6545    7108    7110    7719    7766
RXINEX  036250          6368    6381    6403    6408    6431    6437#
RXIN1   035716          6359    6362#
RXIN2   035644          6351    6353#
RXIN21  035730          6343    6345    6347    6367#
RXIN3   036110          6371    6407#
RXIN4   035756          6375#
RXIN5   036164          6383    6424#
RXIN6   036064          6386    6400#
RXIN7   036234          6422    6425    6433#
RXIN8   036242          6398    6435#
RXM   = 000040          2261#   6072    6184    6187    6424    6426    7102    7149    7677    7682
RXMTOT  006476          2599#   5369*   5415    5488    5678
RXON    041774          6967*   6981*   6994*   7012*   7020*   7045*   7054*   7098#   7132*   7879*
RXONEX  042076          7099    7112#
RXONLY  032206          2656    5413#
RXON2   032206          5414#
RXOVER= 004000          2380#   7632
RXOVMS= 000001          6777#   7528    7637
RXOVRU= 000011          6769#   7523    7638
RXPRC   037114          6747#   6895*   6952    6973    7004    7028    7032*   7035    7236    7283*   7319    7403*   7699
RXPREX  043724          7143    7242    7291    7302    7332    7388    7405    7407    7409    7411    7414#
RXPROT  042100          6927    6947*   6959*   6970*   6999*   7017*   7034*   7050*   7131#   7157    7164    7178    7194
                        7209    7214    7227    7240    7251    7259    7270    7279    7311    7347    7357    7366    7380
                        7412    7759*
RXPTR   006440          2582#   4785*   5367*   5390    5414    5463    5489    5584    5676
RXQ   = 000004          2238#   3279    5884
RXTHER  037066          6707#   7199*   7555    7557*   7561*
RXTXTE  037104          6742#   7489*   7560*
RXWAIT  042104          6995*   7013*   7046*   7135#
R10$    021034          3444#
R11$    021042          3445    3446#
R12$    021060          3447    3448#
R125$   021230          3464#
R13$    021076          3449    3450#
R14$    021112          3451    3452#
R20$    021136          3454#
R21$    021160          3456    3457#
R22$    021176          3458    3459#
R30$    021226          3453    3455    3460    3461    3462    3463#
SACK  = 000004          6802#   7015    7254    7299    7345    7401    7773    7802
SCM     016603          3018#   3300
SCMD    016636          3018#   3308
SCML    016625          3018#   3304
SDATA = 000020          6804#   6945    6968    6997    7048    7212    7781
SDVE    016572          3018#   3286
SDVI    016614          3018#   3291
SELECT  037130          6753#   6917*   6992    7010    7043    7176*   7675*   7757    7772*   7799*
SELTHE  037065          6705#
SEOP    016647          3018#   3312
SETET = 000060          2334#   2891    4880    5248
```

```
SETEXP= 000010            2294#   2832    4904    5168    5243
SETTRN= 000011            2295#   2834    5171
SHF0    014501            3018#   4128
SHF1    014537            3018#   4164
SHMSG   013507            3018#   5141
SHOW  = 000002            2288#   2793    4902    5062    5095    5109
SHTAB   003072            2539#   5129    5135
SHTEND  003101            2542#   5132
SHTYP0  013543            2532    3018#
SHTYP1  013552            2532    3018#
SHTYP2  013557            2532    3018#
SHTYP3  013564            2532    3018#
SHTYP4  013571            2532    3018#
SHTYP5  013577            2532    3018#
SHTYP6  013604            2532    3018#
SHTYP7  013612            2532    3018#
SHTYTB  003052            2532#   5140
SHWOP   023712            3719    4112#   4828    5153
SIZE  = 000012            2296#   2905    5175    5181
SMSC    016660            3018#   3318
SNAK  = 000010            6803#   7160    7169    7262    7306    7360    7373    7552    7635    7694    7706    7775    7814
SOH   = 000201            6784#   7842
SPARE0  037106            6743#
SPARE1  037107            6744#
SQD   = 000040            2355#   2698
SRXQ    016561            3018#   3278
SSTACK= 002000            6808#   7277    7779    7836
SSTART= 004000            6809#   6925    7777    7825
STACK = 000007            6795#   7297    7831
STADD   006510            2605#   3861    5156*
STAIND  037206            3556    6842#
STALST  037146            3554    6819#
STAPRI  021540            3510    3516    3530    3543#
START   025444            4485    4504#
STATB = 000001            2220#   3335    4136    5303
STATUS= 000016            2300#   2856
STA0A   037256            6819    6868#
STA1A   037305            6820    6868#
STA10A  040003            6827    6868#
STA11A  040065            6828    6868#
STA12A  040162            6829    6868#
STA13A  040256            6830    6868#
STA14A  040354            6831    6868#
STA15A  040452            6832    6868#
STA16A  040540            6833    6868#
STA17A  040625            6834    6868#
STA2A   037343            6821    6868#
STA3A   037377            6822    6868#
STA4A   037471            6823    6868#
STA5A   037555            6824    6868#
STA6A   037635            6825    6868#
STA7A   037721            6826    6868#
STRT  = 000006            6794#   7273    7821
STXC    016550            3018#   3273
STXQ    016537            3018#   3268
SVCGBL= 000000            1945#   1964    1973    1975    1977    1979    1981    1983    1985    1987    1989    1991    1993
```

SVCINS= 000001

1995 1997 1999 2001 2003 2005 2007 2010 2013 2015 2017 2019 2021
2023 2025 2027 2029 2031 2033 2035 2037 2039 2041 2043 2045 2047
2059 2073 2074 2994 3004 3021 3038 3051 3065 3085 3104 3203 4435
4455 4471 4655 4669 4705 4727 6339 6477 7598 7909 7991# 7992
1945# 1965 1966 1967 1968 1969 1970 1971 1972 1974 1976 1978 1980
1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006
2008 2009 2011 2012 2014 2016 2018 2020 2022 2024 2026 2028 2030
2032 2034 2036 2038 2040 2042 2044 2046 2048 2058 2060 2072 2995
2997 3005 3010 3023 3024 3025 3026 3027 3028 3029 3030 3031 3032
3035 3040 3041 3042 3043 3044 3045 3048 3053 3054 3055 3056 3057
3058 3059 3062 3067 3068 3069 3070 3071 3072 3073 3076 3087 3088
3089 3090 3091 3092 3093 3096 3106 3107 3108 3109 3110 3111 3112
3113 3116 3119 3120 3229 3344 3345 3346 3347 3348 3352 3353 3354
3355 3356 3390 3391 3392 3393 3394 3401 3402 3403 3404 3405 3406
3407 3408 3418 3419 3420 3421 3422 3428 3429 3430 3431 3432 3487
3488 3489 3490 3491 3492 3533 3534 3535 3536 3537 3547 3548 3549
3550 3551 3567 3568 3569 3570 3571 3572 3578 3579 3580 3581 3582
3583 3584 3585 3586 3603 3604 3605 3606 3607 3608 3609 3610 3611
3612 3613 3614 3615 3634 3635 3636 3637 3638 3657 3658 3659 3660
3661 3667 3668 3669 3670 3671 3672 3673 3674 3675 3682 3683 3684
3685 3686 3687 3688 3696 3697 3698 3699 3700 3701 3703 3704 3705
3706 3707 3708 3709 3729 3730 3731 3732 3733 3734 3735 3736 3745
3746 3747 3748 3749 3750 3751 3752 3760 3761 3762 3763 3764 3765
3766 3767 3777 3778 3779 3780 3781 3782 3786 3787 3788 3789 3790
3806 3807 3808 3809 3810 3826 3827 3828 3829 3830 3864 3865 3866
3867 3868 3869 3874 3875 3876 3877 3878 3879 3880 3883 3884 3885
3886 3887 3888 3928 3929 3930 3931 3932 4124 4125 4126 4127 4128
4129 4130 4131 4132 4159 4160 4161 4162 4163 4164 4165 4166 4167
4168 4310 4311 4312 4313 4314 4353 4354 4355 4356 4357 4446 4478
4482 4483 4485 4487 4488 4490 4492 4493 4495 4498 4499 4501 4509
4510 4511 4513 4519 4520 4521 4523 4532 4534 4539 4540 4541 4542
4543 4544 4545 4546 4551 4552 4553 4554 4555 4573 4574 4575 4577
4608 4609 4610 4611 4612 4613 4618 4619 4620 4621 4622 4623 4625
4626 4627 4628 4629 4630 4633 4634 4636 4637 4644 4660 4673 4674
4686 4688 4689 4696 4709 4710 4717 4731 4732 4739 4769 4770 4771
4772 4773 4811 4812 4813 4814 4815 4817 4818 4819 4820 4821 4822
4831 4833 4837 4838 4847 4848 4849 4850 4851 4852 4853 4854 4864
4865 4866 4867 4868 4873 4874 4875 4876 4877 4897 4898 4910 4911
4912 4913 4914 4915 4925 4926 4927 4928 4929 4930 4951 4952 4953
4954 4955 4956 4967 4968 4969 4970 4971 4972 5067 5068 5069 5070
5071 5072 5139 5140 5141 5142 5143 5144 5145 5186 5187 5188 5189
5190 5216 5217 5218 5219 5220 5290 5291 5292 5293 5294 5339 5340
5341 5342 5343 5356 5357 5358 5359 5360 5698 5699 5700 5701 5716
5717 5718 5719 5729 5730 5731 5732 5772 5773 5774 5775 5824 5825
5826 5827 5828 5857 5858 5859 5860 5861 5862 5863 5864 5921 5922
5923 5924 5925 5948 5949 5950 5951 5952 6003 6014 6015 6016 6017
6051 6052 6053 6054 6057 6095 6118 6119 6120 6121 6259 6271 6272
6273 6274 6393 6394 6395 6396 6418 6419 6420 6421 6440 6509 6559
6599 6617 6631 6632 6633 6634 6903 6904 6905 6906 6907 6908 7137
7286 7287 7288 7289 7290 7452 7453 7454 7455 7470 7471 7472 7473
7724 7872 7890 7908 7915 7916 7917 7925 7926 7927 7928 7930 7931
7932 7933 7935 7936 7937 7939 7941 7942 7943 7945 7947 7948 7949
7950 7951 7953 7988 7989 7990

SVCSUB= 000001

1945#

SVCTAG= 000001

1945# 2100 3034 3047 3061 3075 3095 3115 3228 3409 4445 4547 4643
4659 4695 4716 4738 4855 5865 6439 6508 7723 7889 7954

```
SVCTST= 000001  1945#  4754
SYN   = 000226  2378#
SYNCC   011506  2968#  6090*  6253*  6479  6482*  6553*  7856*
SYNCW   011510  2969#  5379*  5384*  6481
S$LSYM= 010000  1945#  2101#  3035#  3048#  3062#  3076#  3096#  3116#  3229#  3402  3409  3410#  4446#
                4540  4547  4548#  4644#  4660#  4696#  4717#  4739#  4848  4855  4856#  5858  5865
                5866#  6440#  6509#  7724#  7890#  7955#
S1      025432  4495  4497#
S2      025506  4513  4518#
S3      025556  4523  4531#
S4      025624  4534  4550#
T       037056  6694#  7086*  7223  7229  7230  7232*  7325  7327  7328  7329  7330*  7394  7396
                7397  7398  7399*  7846
TABEX   014307  3018#  4925  4967
TAL   = 000005  2202#  5274  5957
TALCK   033612  2661  5850#  5893  5895
TALMOD= 000035  2315#  2817
TBMT  = 000004  2370#
TCURAD  006466  2594#  4791*  4794  4936  4941*  5116*
TDSR    011462  2947#  4593*  4594*  6481*  6484*  6487*  6493*  6497*
TEMP    006534  2619#  3269*  3274*  3279*  3283*  3287*  3292*  3301*  3305*  3309*  3313*  3319*  3333
                3359  3360*  3361*  3362  3872*  3875  3934*  3935*  3936  3974*  3975*  3981  4114*
                4127  4135*  4138*  4163  4679*  5137*  5139  5200*  5201*  5205  5208  5590*  5591*
                5595  6596*  6597*  6600  7080*  7085
TEMP1   006536  2620#  3268*  3273*  3278*  3286*  3291*  3300*  3304*  3308*  3312*  3318*  3352  3590*
                3595*  3608  4121*  4125  4139*  4142*  4162
TEMP2   006540  2621#  3293*  3294*  3365  3591*  3598*  3606  3977*  3979*  3983  4122*  4124  4143*
                4146*  4161  4681*  5543*  5555*  5573*  5579  5593*  5594  5614*  5682*  5799*  5874*
                5889*  5927*  5942*  6008*  6045*  6112*  6265*  6358*  6361*  6389*  6412*  6625*  6652*
                6653*  7446*  7464*  7619*  7622*
TEMP3   006542  2622#  3053  3088  3108  3295*  3366  3592*  3601*  3604  4117*  4120*  4126  4682*
                5545*  5557*  5574*  5580  5591  5592  5593  5615*  5684*  5694  5800*  5875*  5890*
                5929*  5943*  6009*  6046*  6113*  6266*  6353*  6355  6387*  6413*  6626*  7444*  7447*
                7465*  7614*  7616
TEMP4   006544  2623#  3040  3087  3107  3296*  3330*  3367  4148*  4151*  4159  4680*  5691*  5704*
                5709*  5710  5725  5798*  6010*  6047*  6114*  6267*  6354*  6355  6388*  6414*  6627*
                7445*  7448*  7466*  7615*  7616
TEMP5   006546  2624#  4152*  4155*  4160
TEOM  = 001000  2375#  6484  6497
TERR  = 100000  2376#
TIMEOU  037144  6759#  6897*  7461*  7462  7474*
TIMERS  006644  2678#  3221  3225*  6073*  6077  6096  6102  6256*  6260  6560  6916*  6933*  7140
                7154  7188  7341  7354  7438  7457*  7476*
TIMER1  006640  2676#  3215  3217*  4763*  4764  6039*  6040  6603*  6610  6615*  6620
TIMER2  006642  2677#  3218  3220*
TIMMIN  006632  2672#  3212*  3364  4557*
TIMSEC  006634  2673#  3209*  3210  3213*  3363  4558*
TIMTCK  006636  2674#  3206*  3208*  3223  3361  4559*
TINTEN= 000100  2373#  6091  6254  6488  6554  7859
TM    = 000040  2356#  2699  6350  6352  7611  7613
TMESRX  037052  6687#  6961*  6975*  7019*  7037*  7056  7058*
TMESTX  037050  6686#  6950*  6976*  7002*  7053*  7059  7061*
TOINOT  035472  6261  6277#
TOTCC   006530  2617#  3921*  3922  3933*  3935  3937*  4780*  4906*  4907  4940  4947*  4948  4984
TRA   = 000001  2198#  5271
TRAMOD= 000034  2314#  2815
```

```
TRIBN   037061          4603* 4605* 6698# 7080  7085* 7181  7668  7769
TRIBNQ  046524          7948  7956#
TRVACT  024364          4223  4234# 4250  4255  4260  4263  4283  4349  4372  4393  4417
TRVALN  025156          4212  4376#
TRVALP  025112          4211  4362#
TRVBIF  024470          4208  4263#
TRVBR   024460          4207  4260#
TRVBRC  024404          4221  4241# 4261  4266  4285  4359  4374  4395  4421
TRVDEC  024564          4214  4288#
TRVERR  024422          4205  4250#
TRVEXI  024442          4206  4255#
TRVNMA  024604          4289  4292#
TRVNOB  024414          4246# 4267  4284  4350  4373  4394
TRVNUM  024576          4210  4291#
TRVOCT  024576          4213  4290#
TRVSPA  024512          4209  4269#
TRVSTR  025244          4215  4399#
TSOM  = 000400          2374# 6484
TTL   = 000001          2206# 4839  6024
TTLL  = 000010          2371# 6023  6026
TTLLOP= 000044          2322# 2919
TTOTCC  006464          2593# 4084  4781* 4906  4917  4940* 5112*
TURNON  037142          6758# 6918* 7098  7111* 7718* 7767*
TXACT = 000002          2369#
TXBUF   003150          2572# 4052  4791  5115
TXC   = 000002          2237# 3274
TXCSR   011460          2946# 4591* 4592* 5998* 6010  6023* 6026* 6047  6091* 6153  6254* 6267  6350
                        6488* 6554* 6627  7466  7611  7859*
TXENA = 000020          2372# 6091  6254  6554  7859
TXINEX  036440          6483  6491  6496  6503  6506#
TXIN1   036300          6480  6484#
TXIN2   036346          6486  6492#
TXIN3   036432          6499  6504#
TXM   = 000020          2259# 6072  6239  6243  6498  6500  7788  7807  7819  7829  7841
TXMTOT  006462          2592# 4072  4797* 4919* 4921  4942* 5107  5111* 5353  5438  5461  5486
TXONLY  032240          2657  5436#
TXON2   032246          5437#
TXPRC   037116          6748# 6896* 6948  6971  7000  7003* 7027* 7040  7051  7233* 7284* 7322  7331*
                        7400* 7410  7794
TXPREX  046236          7869  7879#
TXPROT  045350          6926  6946* 6969* 6998* 7016* 7049* 7163  7177  7213  7258  7269  7278  7300
                        7310  7346* 7365  7379  7402  7745#
TXPTR   006442          2583# 4065* 4067* 4068  4077* 4079  4784* 4795  4920* 4934* 4935  4939* 5113*
                        5114  5365* 5437  5462  5487
TXQ   = 000000          2236# 3269
TXREAD  037122          6490* 6750# 6894* 6979  7750  7771* 7873
TXTHER  037067          6708# 7228* 7485  7487* 7490*
TSARGC= 000001          1965# 1966# 1967# 1968# 1969# 1970# 3023# 3032  3040# 3045  3053# 3059  3067#
                        3073  3087# 3093  3106# 3113  3344# 3348  3352# 3356  3390# 3394  3418# 3422
                        3428# 3432  3487# 3492  3533# 3537  3547# 3551  3567# 3572  3578# 3586  3603#
                        3615  3634# 3638  3657# 3661  3667# 3675  3682# 3688  3696# 3701  3703# 3709
                        3729# 3736  3745# 3752  3760# 3767  3777# 3782  3786# 3790  3806# 3810  3826#
                        3830  3864# 3869  3874# 3880  3883# 3888  3928# 3932  4124# 4132  4159# 4168
                        4310# 4314  4353# 4357  4551# 4555  4769# 4773  4811# 4815  4864# 4868  4873#
                        4877  4910# 4915  4925# 4930  4951# 4956  4967# 4972  5067# 5072  5139# 5145
                        5186# 5190  5216# 5220  5290# 5294  5339# 5343  5356# 5360  5824# 5828  5921#
```

```
                    5925    5948#   5952    7286#   7290
T$CODE= 005052      3404#   4542#   4850#   5860#   7915#   7925#   7930#   7935#   7939#   7941#   7945#   7947#
T$ERRN= 000007      1945#   5699#   5717#   5730#   5773#   6015#   6052#   6119#   6272#   6394#   6419#   6632#   7453#
                    7471#
T$EXCP= 000000      3404#   3409    4542#   4547    4850#   4855    5860#   5865    7925#   7929    7930#   7934    7947#
                    7952
T$FLAG= 000040      3119#   3121    4636#   4688#   4709#   4711    4731#   4733    4837#   4897#
T$GMAN= 000000      1945#   3401#   3404    3410#   4539#   4548#   4847#   4850    4856#   5857#   5860    5866#
T$HILI= 000377      3404#   3408    4542#   4546    4850#   4854    5860#   5864    7925#   7928    7930#   7933    7947#
                    7951
T$LAST= 000001      1945#   7989#
T$LOLI= 000001      3404#   3407    4542#   4545    4850#   4853    5860#   5863    7925#   7927    7930#   7932    7947#
                    7950
T$LSYM= 010000      1945#   2101    3035    3048    3062    3076    3096    3116    3229    4446    4644    4660    4696
                    4717    4739    6440    6509    7724    7890    7955
T$LTNO= 000001      7992#
T$NEST= 177777      1945#   1947#   2072#   2100#   3021#   3034#   3038#   3047#   3051#   3061#   3065#   3075#   3085#
                    3095#   3104#   3115#   3203#   3228#   4435#   4445#   4455#   4462#   4471#   4643#   4655#   4659#
                    4669#   4695#   4705#   4716#   4727#   4738#   4755#   6339#   6439#   6477#   6508#   7598#   7723#
                    7889#   7908#   7939    7945    7953#   7993#
T$NS0 = 000000      1947#   7993
T$NS1 = 000004      2072#   2100    3021#   3034    3038#   3047    3051#   3061    3065#   3075    3085#   3095    3104#
                    3115    3203#   3228    4435#   4445    4455#   4462    4471#   4643    4655#   4659    4669#   4695
                    4705#   4716    4727#   4738    4755#   7889    7908#   7939    7945    7953
T$NS2 = 000010      6339#   6439    6477#   6508    7598#   7723
T$PTNU= 000000      1945#
T$SAVL= 177777      1945#
T$SEGL= 177777      1945#
T$SUBN= 000000      1945#   4754#
T$TAGL= 177777      1945#
T$TAGN= 010024      1945#   2072#   3021#   3038#   3051#   3065#   3085#   3104#   3203#   4435#   4455#   4471#   4655#
                    4669#   4705#   4727#   4755#   6339#   6477#   7598#   7908#
T$TEMP= 000000      2060#   2061#   2100#   3034#   3047#   3061#   3075#   3095#   3115#   3119#   3120    3228#   3404#
                    4445#   4462#   4542#   4636#   4637    4643#   4659#   4688#   4689    4695#   4709#   4710    4716#
                    4731#   4732    4738#   4837#   4838    4850#   4897#   4898    5860#   6439#   6508#   7723#   7889#
                    7915#   7925#   7930#   7935#   7941#   7947#   7953#   7993#
T$TEST= 000001      1945#   4754#   7992
T$TSTM= 177777      1945#   3031    3035    3044    3048    3058    3062    3072    3076    3092    3096    3112    3116
                    3347    3355    3393    3401    3421    3431    3491    3536    3550    3571    3585    3614    3637
                    3660    3674    3687    3700    3708    3735    3751    3766    3781    3789    3809    3829    3868
                    3879    3887    3931    4131    4167    4313    4356    4446    4478    4483    4488    4493    4499
                    4510    4520    4532    4539    4554    4574    4612    4622    4629    4634    4636    4644    4660
                    4674    4686    4688    4696    4717    4739    4772    4814    4821    4831    4837    4847    4867
                    4876    4897    4914    4929    4955    4971    5071    5144    5189    5219    5293    5342    5359
                    5698    5716    5729    5772    5827    5857    5924    5951    6003    6014    6051    6057    6095
                    6118    6259    6271    6393    6418    6559    6599    6617    6631    6907    7137    7289    7452
                    7470    7872    7890
T$TSTS= 000001      1945#   4755#
T$$AU = 010016      4727#   4731    4738
T$$AUT= 010013      4655#   4659
T$$CLE= 010014      4669#   4688    4695
T$$DU = 010015      4705#   4709    4716
T$$HAR= 010023      7908#   7954
T$$HW = 010000      2072#   2100
T$$INI= 010012      4471#   4636    4643
T$$MSG= 010006      3021#   3034    3038#   3047    3051#   3061    3065#   3075    3085#   3095    3104#   3115    3119
```

```
T$$PRO= 010011            4455#
T$$RPT= 010010            4435#   4445
T$$SRV= 010022            3203#   3228    6339#   6439    6477#   6508    7598#   7723
T$$TES= 010017            4755#   4837    4897    7889
T1      026340 G          2060    4754#
UAM   = 000200 G          2176#
UPTABL  032606            5576#
UPTA1   032674            5582    5589#
UPTA3   032672            5586    5588#
UPTA4   032632            5577    5581#
UPTEX   032744            5588    5600#
VECTOR  046407            7931    7956#
X       037057            6695#   7230*   7328*   7397*
XS    = 000220            1948#   2777#   2778#   2779#   2780#   2781#   2782#   2783#   2784#   2785#   2786#   2787#   2788#
                          2789#   2790#   2791#   2792#   2793#   2794#   2795#   2796#   2797#   2801#   2802#   2803#   2804#
                          2805#   2806#   2807#   2808#   2809#   2810#   2811#   2812#   2813#   2814#   2815#   2816#   2817#
                          2818#   2822#   2823#   2824#   2825#   2826#   2831#   2832#   2833#   2834#   2835#   2838#   2839#
                          2840#   2841#   2842#   2843#   2844#   2845#   2848#   2849#   2850#   2851#   2852#   2853#   2856#
                          2857#   2858#   2859#   2861#   2862#   2863#   2864#   2865#   2867#   2868#   2870#   2871#   2874#
                          2877#   2878#   2879#   2880#   2881#   2882#   2883#   2884#   2885#   2886#   2887#   2888#   2889#
                          2890#   2891#   2892#   2894#   2895#   2896#   2897#   2898#   2899#   2900#   2903#   2904#   2905#
                          2906#   2907#   2908#   2911#   2912#   2913#   2916#   2919#   2920#   2921#   2922#   2923#   2924#
                          2925#   2926#   2929#   2930#   2931#   2936#   3444#   3445#   3446#   3447#   3448#   3449#   3450#
                          3451#   3452#   3453#   3454#   3455#   3456#   3457#   3458#   3459#   3460#   3461#   3462#   3463#
                          3464#   3465#
XSALWA= 000000            1945#
XSFALS= 000040            1945#   7945
XSOFFS= 000400            1945#   7939    7945
XSTRUE= 000020            1945#   7939
$PATCH  046552            7983#
.     = 046636            1945#   2466#   2472#   2480#   2495#   2505#   2509#   2543#   2572#   2573#   2574#   2575#   2576#
                          2577#   2580#   2682#   2683#   2780#   2784#   2788#   2795#   2802#   2804#   2810#   2812#   2823#
                          2825#   2832#   2834#   2850#   2852#   2856#   2858#   2870#   2877#   2879#   2881#   2883#   2885#
                          2891#   2905#   2907#   2923#   2997#   3018#   3120    3447#   3449#   3453#   3458#   3460#   4637
                          4689    4710    4732    4838    4898    7293#   7939    7945    7984#
```

```
BCOMPL      1#  1945#  4484  4489  4500  4832
BERROR      1#  1945#
BGNAU       1#  1945#  4726
BGNAUT      1#  1945#  4654
BGNCLN      1#  1945#  4668
BGNDU       1#  1945#  4704
BGNHRD      1#  1945#  7907
BGNHW       1#  1945#  2071
BGNINI      1#  1945#  4470
BGNMOD      1#  1945#  1946
BGNMSG      1#  1945#  3020  3037  3050  3064  3084  3103
BGNPRO      1#  1945#  4454
BGNPTA      1#  1945#
BGNRPT      1#  1945#  4434
BGNSEG      1#  1945#
BGNSET      1#  1945#
BGNSFT      1#  1945#
BGNSRV      1#  1945#  3202  6338  6476  7597
BGNSUB      1#  1945#
BGNSW       1#  1945#
BGNTST      1#  1945#  4753
BNCOMP      1#  1945#  4494  4512  4522  4533  4576
BNERRO      1#  1945#
BREAK       1#  1945#  6002  6056  6094  6258  6558  6598  6616  7136  7871
BRESET      1#  1945#  4685
CKLOOP      1#  1945#
CLI      1949#  2776   2777  2778  2779  2780  2781  2782  2783  2784  2785  2786  2787  2788  2789
         2790   2791   2792  2793  2794  2795  2796  2800  2801  2802  2803  2804  2805  2806  2807
         2808   2809   2810  2811  2812  2813  2814  2815  2816  2817  2821  2822  2823  2824  2825
         2830   2831   2832  2833  2834  2837  2838  2839  2840  2841  2842  2843  2844  2847  2848
         2849   2850   2851  2852  2855  2856  2857  2858  2860  2861  2862  2863  2864  2866  2867
         2869   2870   2873  2876  2877  2878  2879  2880  2881  2882  2883  2884  2885  2886  2887
         2888   2889   2890  2891  2893  2894  2895  2896  2897  2898  2899  2902  2903  2904  2905
         2906   2907   2910  2911  2912  2915  2918  2919  2920  2921  2922  2923  2924  2925  2928
         2929   2930   2935  3443  3444  3445  3446  3447  3448  3449  3450  3451  3452  3453  3454
         3455   3456   3457  3458  3459  3460  3461  3462  3463  3464
CLOCK       1#  1945#  4508  4518
CLOSE       1#  1945#
CLRVEC      1#  1945#
COMMEN      1#  1945#
DELAY       1#  1945#
DESCRI      1#  1945#  3003
DEVTYP      1#  1945#  2993
DISPAT      1#  1945#  2057
DISPLA      1#  1945#
DOCLN       1#  1945#  4477
DODU        1#  1945#
DORPT       1#  1945#
ENDAU       1#  1945#  4737
ENDAUT      1#  1945#  4658
ENDCLN      1#  1945#  4694
ENDCOM      1#  1945#
ENDDU       1#  1945#  4715
ENDHRD      1#  1945#  7952
ENDHW       1#  1945#  2099
ENDINI      1#  1945#  4642
```

```
ENDMOD      1#  1945#  7992
ENDMSG      1#  1945#  3033  3046  3060  3074  3094  3114
ENDPRO      1#  1945#  4461
ENDPTA      1#  1945#
ENDRPT      1#  1945#  4444
ENDSEG      1#  1945#
ENDSET      1#  1945#
ENDSFT      1#  1945#
ENDSRV      1#  1945#  3227  6438  6507  7722
ENDSUB      1#  1945#
ENDSW       1#  1945#
ENDTST      1#  1945#  7888
EQUALS      1#  1945#  2116
ERRDF       1#  1945#
ERRHRD      1#  1945#
ERROR       1#  1945#
ERRSF       1#  1945#
ERRSOF      1#  1945#  5697  5715  5728  5771  6013  6050  6117  6270  6392  6417  6630  7451  7469
ERRTBL      1#  1945#
ESCAPE      1#  1945#
EXIT        1#  1945#  3118  4635  4687  4708  4730  4836  4896
FEQUAL      1#  1945#
GETBYT      1#  1945#
GETPRI      1#  1945#
GETWOR      1#  1945#
GMANIA      1#  1945#
GMANID      1#  1945#  3400  4538  4846  5856
GMANIL      1#  1945#
GPHARD      1#  1945#  4572
GPRMA       1#  1945#  7924  7929
GPRMD       1#  1945#  3401#  3404  4539#  4542  4847#  4850  5857#  5860  7946
GPRML       1#  1945#  7914  7934  7940
HEADER      1#  1945#  1963
INLOOP      1#  1945#
IOSETU      1#  1945#
IOSTAR      1#  1945#
KT11        1#  1945#
LASTAD      1#  1945#  7987
MANUAL      1#  1945#  4830
MEMORY      1#  1945#
MSBYTE      1#  1945#  1964#  1970  1971  1972
MSCHEC      1#  1945#  3119#  4636#  4688#  4709#  4731#  4837#  4897#
MSCNTO      1#  1945#  3404#  4542#  4850#  5860#  7915#  7925#  7930#  7935#  7941#  7947#
MSCOUN      1#  1945#  3023#  3040#  3053#  3067#  3087#  3106#  3344#  3352#  3390#  3418#  3428#  3487#  3533#
            3547#  3567#  3578#  3603#  3634#  3657#  3667#  3682#  3696#  3703#  3729#  3745#  3760#  3777#  3786#
            3806#  3826#  3864#  3874#  3883#  3928#  4124#  4159#  4310#  4353#  4551#  4769#  4811#  4864#  4873#
            4910#  4925#  4951#  4967#  5067#  5139#  5186#  5216#  5290#  5339#  5356#  5824#  5921#  5948#  7286#
MSDATA      1#  1945#  1964#  1973  1975  1977  1979  1981  1983  1985  1987  1989  1991  1993  1995
            1997  1999  2001  2003#  2005  2007  2010  2013  2015  2017  2019  2021  2023  2025  2027
            2029  2031  2033  2035  2037  2039  2041  2043  2045  2047  2994#  3004#
MSDECR      1#  1945#  2100#  3034#  3047#  3061#  3075#  3095#  3115#  3228#  4445#  4462#  4643#  4659#  4695#
            4716#  4738#  6439#  6508#  7723#  7889#  7953#  7993#
MSDEFA      1#  1945#  3404#  4542#  4850#  5860#  7915#  7925#  7930#  7935#  7941#  7947#
MSENDE      1#  1945#  2100#  3034#  3047#  3061#  3075#  3095#  3115#  3228#  4445#  4643#  4659#  4695#  4716#
            4738#  6439#  6508#  7723#  7889#  7953#  7993#
MSERRI      1#  1945#  5698#  5716#  5729#  5772#  6014#  6051#  6118#  6271#  6393#  6418#  6631#  7452#  7470#
```

M$ESCA	1#	1945#													
M$ESCS	1#	1945#													
M$EXCP	1#	1945#	3404#	4542#	4850#	5860#	7925#	7930#	7947#						
M$EXIT	1#	1945#	3119#	4636#	4637	4688#	4689	4709#	4731#	4837#	4838	4897#	4898		
M$EXSE	1#	1945#	3119#	4636#	4688#	4709#	4731#	4837#	4897#						
M$EXTJ	1#	1945#	3119#	3120	4636#	4688#	4709#	4710	4731#	4732	4837#	4897#			
M$GEN	1#	1945#	1964#	1973#	1975#	1977#	1979#	1981#	1983#	1985#	1987#	1989#	1991#	1993#	1995#
	1997#	1999#	2001#	2003#	2005#	2007#	2010#	2013#	2015#	2017#	2019#	2021#	2023#	2025#	2027#
	2029#	2031#	2033#	2035#	2037#	2039#	2041#	2043#	2045#	2047#	2059#	2073#	2074#	2100#	2994#
	3004#	3021#	3034#	3038#	3047#	3051#	3061#	3065#	3075#	3085#	3095#	3104#	3115#	3203#	3228#
	3409#	4435#	4445#	4455#	4471#	4547#	4643#	4655#	4659#	4669#	4695#	4705#	4716#	4727#	4738#
	4754#	4855#	5865#	6339#	6439#	6477#	6508#	7598#	7723#	7889#	7909#	7954#	7991#		
M$GENB	1#	1945#	3401#	3402	4539#	4540	4847#	4848	5857#	5858					
M$GETS	1#	1945#	2100#	3034#	3047#	3061#	3075#	3095#	3115#	3228#	4445#	4462#	4643#	4659#	4695#
	4716#	4738#	6439#	6508#	7723#	7889#	7939#	7945#	7953#	7993#					
M$GETT	1#	1945#	3119#	4636#	4688#	4709#	4731#	4837#	4897#	7939#	7945#				
M$GNGB	1#	1945#	1947#	1964#	1973#	1975#	1977#	1979#	1981#	1983#	1985#	1987#	1989#	1991#	1993#
	1995#	1997#	1999#	2001#	2003#	2005#	2007#	2010#	2013#	2015#	2017#	2019#	2021#	2023#	2025#
	2027#	2029#	2031#	2033#	2035#	2037#	2039#	2041#	2043#	2045#	2047#	2058#	2059	2072#	2073
	2074	2994#	3004#	3021#	3038#	3051#	3065#	3085#	3104#	3203#	4435#	4455#	4471#	4655#	4669#
	4705#	4727#	6339#	6477#	7598#	7908#	7909	7988#	7991						
M$GNIN	1#	1945#	1964#	1965	1966	1967	1968	1969	1970#	1971#	1972#	1973#	1974	1975#	1976
	1977#	1978	1979#	1980	1981#	1982	1983#	1984	1985#	1986	1987#	1988	1989#	1990	1991#
	1992	1993#	1994	1995#	1996	1997#	1998	1999#	2000	2001#	2002	2003#	2004	2005#	2006
	2007#	2008	2009	2010#	2011	2012#	2013#	2014	2015#	2016	2017#	2018	2019#	2020	2021#
	2022	2023#	2024	2025#	2026	2027#	2028	2029#	2030	2031#	2032	2033#	2034	2035#	2036
	2037#	2038	2039#	2040	2041#	2042	2043#	2044	2045#	2046	2047#	2048	2058#	2060#	2072#
	2994#	2995	2997	3004#	3005	3010	3023#	3024	3025#	3026	3027#	3028#	3029#	3030	3031#
	3032	3035#	3040#	3041#	3042#	3043	3044#	3045	3048#	3053#	3054#	3055#	3056#	3057	3058#
	3059	3062#	3067#	3068#	3069#	3070#	3071	3072#	3073	3076#	3087#	3088#	3089#	3090#	3091
	3092#	3093	3096#	3106#	3107#	3108#	3109#	3110#	3111	3112#	3113	3116#	3119#	3120#	3228#
	3229	3344#	3345#	3346	3347#	3348	3352#	3353#	3354	3355#	3356	3390#	3391#	3392	3393#
	3394	3401#	3402#	3403#	3404#	3405	3406	3407	3408	3418#	3419#	3420	3421#	3422	3428#
	3429#	3430	3431#	3432	3487#	3488#	3489#	3490	3491#	3492	3533#	3534#	3535	3536#	3537
	3547#	3548#	3549	3550#	3551	3567#	3568#	3569#	3570	3571#	3572	3578#	3579	3580#	3581
	3582#	3583#	3584	3585#	3586	3603#	3604	3605#	3606	3607#	3608	3609#	3610	3611#	3612#
	3613	3614#	3615	3634#	3635#	3636	3637#	3638	3657#	3658#	3659	3660#	3661	3667#	3668#
	3669#	3670#	3671#	3672#	3673	3674#	3675	3682#	3683#	3684#	3685#	3686	3687#	3688	3696#
	3697#	3698#	3699	3700#	3701	3703#	3704#	3705#	3706#	3707	3708#	3709	3729#	3730#	3731#
	3732#	3733#	3734	3735#	3736	3745#	3746#	3747#	3748#	3749#	3750	3751#	3752	3760#	3761#
	3762#	3763#	3764#	3765	3766#	3767	3777#	3778#	3779#	3780	3781#	3782	3786#	3787#	3788
	3789#	3790	3806#	3807#	3808	3809#	3810	3826#	3827#	3828	3829#	3830	3864#	3865#	3866#
	3867	3868#	3869	3874#	3875	3876#	3877#	3878	3879#	3880	3883#	3884#	3885#	3886	3887#
	3888	3928#	3929#	3930	3931#	3932	4124#	4125#	4126#	4127#	4128#	4129#	4130	4131#	4132
	4159#	4160#	4161#	4162#	4163#	4164#	4165#	4166	4167#	4168	4310#	4311#	4312	4313#	4314
	4353#	4354#	4355	4356#	4357	4446#	4478#	4482#	4483#	4485#	4487#	4488#	4490#	4492#	4493#
	4495#	4498#	4499#	4501#	4509#	4510#	4511#	4513#	4519#	4520#	4521#	4523#	4532#	4534#	4539#
	4540#	4541#	4542#	4543	4544	4545	4546	4551#	4552#	4553	4554#	4555	4573#	4574#	4575#
	4577#	4608#	4609#	4610#	4611#	4612#	4613	4618#	4619#	4620#	4621#	4622#	4623	4625#	4626#
	4627#	4628#	4629#	4630	4633#	4634#	4636#	4637#	4644#	4660#	4673#	4674#	4686#	4688#	4689#
	4696#	4709#	4710#	4717#	4731#	4732#	4739#	4769#	4770#	4771	4772#	4773	4811#	4812#	4813
	4814#	4815	4817#	4818#	4819#	4820#	4821#	4822	4831#	4833#	4837#	4838#	4847#	4848#	4849#
	4850#	4851	4852	4853	4854	4864#	4865#	4866	4867#	4868	4873#	4874#	4875	4876#	4877
	4897#	4898#	4910#	4911#	4912#	4913	4914#	4915	4925#	4926#	4927#	4928	4929#	4930	4951#
	4952#	4953#	4954	4955#	4956	4967#	4968#	4969#	4970	4971#	4972	5067#	5068#	5069#	5070
	5071#	5072	5139#	5140#	5141#	5142#	5143	5144#	5145	5186#	5187#	5188	5189#	5190	5216#

```
                5217#  5218   5219#  5220   5290#  5291#  5292   5293#  5294   5339#  5340#  5341   5342#  5343   5356#
                5357#  5358   5359#  5360   5698#  5699#  5700#  5701#  5716#  5717#  5718#  5719#  5729#  5730#  5731#
                5732#  5772#  5773#  5774#  5775#  5824#  5825#  5826   5827#  5828   5857#  5858#  5859#  5860#  5861
                5862   5863   5864   5921#  5922#  5923   5924#  5925   5948#  5949#  5950   5951#  5952   6003#  6014#
                6015#  6016#  6017#  6051#  6052#  6053#  6054#  6057#  6095#  6118#  6119#  6120#  6121#  6259#  6271#
                6272#  6273#  6274#  6393#  6394#  6395#  6396#  6418#  6419#  6420#  6421#  6439#  6440   6508#  6509
                6559#  6599#  6617#  6631#  6632#  6633#  6634#  6903#  6904#  6905#  6906#  6907#  6908   7137#  7286#
                7287#  7288   7289#  7290   7452#  7453#  7454#  7455#  7470#  7471#  7472#  7473#  7723#  7724   7872#
                7890#  7908#  7915#  7916   7917   7925#  7926   7927   7928   7930#  7931   7932   7933   7935#  7936
                7937   7939#  7941#  7942   7943   7945#  7947#  7948   7949   7950   7951   7953#  7988#  7989#  7990#
MSGNLS            1#  1945#  3401#  3409   4539#  4547   4847#  4855   5857#  5865
MSGNSU            1#  1945#
MSGNTA            1#  1945#  2100#  3034#  3047#  3061#  3075#  3095#  3115#  3228#  4445#  4643#  4659#  4695#  4716#
               4738#  6439#  6508#  7723#  7889#  7953#  7954
MSGNTE            1#  1945#  4754#
MSHAPT            1#  1945#  1964#
MSHNAP            1#  1945#  1964#  2003
MSINCR            1#  1945#  1947#  2072#  3021#  3031#  3035#  3038#  3044#  3048#  3051#  3058#  3062#  3065#  3072#
               3076#  3085#  3092#  3096#  3104#  3112#  3116#  3203#  3347#  3355#  3393#  3401#  3410   3421#  3431#
               3491#  3536#  3550#  3571#  3585#  3614#  3637#  3660#  3674#  3687#  3700#  3708#  3735#  3751#  3766#
               3781#  3789#  3809#  3829#  3868#  3879#  3887#  3931#  4131#  4167#  4313#  4356#  4435#  4446#  4455#
               4471#  4478#  4483#  4488#  4493#  4499#  4510#  4520#  4532#  4539#  4548   4554#  4574#  4612#  4622#
               4629#  4634#  4636#  4644#  4655#  4660#  4669#  4674#  4686#  4688#  4696#  4705#  4717#  4727#  4739#
               4754#  4755#  4772#  4814#  4821#  4831#  4837#  4847#  4856   4867#  4876#  4897#  4914#  4929#  4955#
               4971#  5071#  5144#  5189#  5219#  5293#  5342#  5359#  5698#  5716#  5729#  5772#  5827#  5857#  5866
               5924#  5951#  6003#  6014#  6051#  6057#  6095#  6118#  6259#  6271#  6339#  6393#  6418#  6477#  6559#
               6599#  6617#  6631#  6907#  7137#  7289#  7452#  7470#  7598#  7872#  7890#  7908#
MSIOSE            1#  1945#
MSLDRO            1#  1945#  4482#  4487#  4492#  4498#  4509#  4519#  4573#  4633#  4673#
MSMASK            1#  1945#
MSMCHI            1#  1945#
MSMCLO            1#  1945#
MSMSK1            1#  1945#
MSPOP             1#  1945#  2100#  3034#  3047#  3061#  3075#  3095#  3115#  3228#  4445#  4462#  4643#  4659#  4695#
               4716#  4738#  6439#  6508#  7723#  7889#  7953#  7993#
MSPRIN            1#  1945#  3023#  3040#  3053#  3067#  3087#  3106#  3344#  3352#  3390#  3418#  3428#  3487#  3533#
               3547#  3567#  3578#  3603#  3634#  3657#  3667#  3682#  3696#  3703#  3729#  3745#  3760#  3777#  3786#
               3806#  3826#  3864#  3874#  3883#  3928#  4124#  4159#  4310#  4353#  4551#  4769#  4811#  4864#  4873#
               4910#  4925#  4951#  4967#  5067#  5139#  5186#  5216#  5290#  5339#  5356#  5824#  5921#  5948#  7286#
MSPUSH            1#  1945#  1947#  2072#  3021#  3038#  3051#  3065#  3085#  3104#  3203#  4435#  4455#  4471#  4655#
               4669#  4705#  4727#  4754#  4755   6339#  6477#  7598#  7908#
MSPUT             1#  1945#  3023#  3040#  3053#  3067#  3087#  3106#  3344#  3352#  3390#  3418#  3428#  3487#  3533#
               3547#  3567#  3578#  3603#  3634#  3657#  3667#  3682#  3696#  3703#  3729#  3745#  3760#  3777#  3786#
               3806#  3826#  3864#  3874#  3883#  3928#  4124#  4159#  4310#  4353#  4551#  4608#  4618#  4625#  4769#
               4811#  4817#  4864#  4873#  4910#  4925#  4951#  4967#  5067#  5139#  5186#  5216#  5290#  5339#  5356#
               5824#  5921#  5948#  6903#  7286#
MSPUT1            1#  1945#  3023#  3025   3027   3028   3029   3040#  3041   3042   3053#  3054   3055   3056   3067#
               3068   3069   3070   3087#  3088   3089   3090   3106#  3107   3108   3109   3110   3344#  3345   3352#
               3353   3390#  3391   3418#  3419   3428#  3429   3487#  3488   3489   3533#  3534   3547#  3548   3567#
               3568   3569   3578#  3580   3582   3583   3603#  3605   3607   3609   3611   3612   3634#  3635   3657#
               3658   3667#  3668   3669   3670   3671   3672   3682#  3683   3684   3685   3696#  3697   3698   3703#
               3704   3705   3706   3729#  3730   3731   3732   3733   3745#  3746   3747   3748   3749   3760#  3761
               3762   3763   3764   3777#  3778   3779   3786#  3787   3806#  3807   3826#  3827   3864#  3865   3866
               3874#  3876   3877   3883#  3884   3885   3928#  3929   4124#  4125   4126   4127   4128   4129   4159#
               4160   4161   4162   4163   4164   4165   4310#  4311   4353#  4354   4551#  4552   4608#  4609   4610
               4611   4618#  4619   4620   4621   4625#  4626   4627   4628   4769#  4770   4811#  4812   4817#  4818
```

```
                4819    4820    4864#   4865    4873#   4874    4910#   4911    4912    4925#   4926    4927    4951#   4952    4953
                4967#   4968    4969    5067#   5068    5069    5139#   5140    5141    5142    5186#   5187    5216#   5217    5290#
                5291    5339#   5340    5356#   5357    5824#   5825    5921#   5922    5948#   5949    6903#   6904    6905    6906
                7286#   7287
M$RADI             1#   1945#   3404#   4542#   4850#   5860#   7915#   7925#   7930#   7935#   7941#   7947#
M$RBRO             1#   1945#
M$RNRO             1#   1945#   4509#   4511    4519#   4521    4573#   4575
M$SETS             1#   1945#   1947#   2072#   3021#   3038#   3051#   3065#   3085#   3104#   3203#   4435#   4455#   4471#   4655#
                4669#   4705#   4727#   4755#   6339#   6477#   7598#   7908#
M$STAR             1#   1945#
M$SVC              1#   1945#   3023#   3031    3034#   3035    3040#   3044    3047#   3048    3053#   3058    3061#   3062    3067#
                3072    3075#   3076    3087#   3092    3095#   3096    3106#   3112    3115#   3116    3119#   3344#   3347    3352#
                3355    3390#   3393    3401#   3418#   3421    3428#   3431    3487#   3491    3533#   3536    3547#   3550    3567#
                3571    3578#   3585    3603#   3614    3634#   3637    3657#   3660    3667#   3674    3682#   3687    3696#   3700
                3703#   3708    3729#   3735    3745#   3751    3760#   3766    3777#   3781    3786#   3789    3806#   3809    3826#
                3829    3864#   3868    3874#   3879    3883#   3887    3928#   3931    4124#   4131    4159#   4167    4310#   4313
                4353#   4356    4445#   4446    4478#   4482#   4483    4487#   4488    4492#   4493    4498#   4499    4509#   4510
                4519#   4520    4532#   4539#   4551#   4554    4573#   4574    4608#   4612    4618#   4622    4625#   4629    4633#
                4634    4636#   4643#   4644    4659#   4660    4673#   4674    4686#   4688#   4695#   4696    4709#   4716#   4717
                4731#   4738#   4739    4769#   4772    4811#   4814    4817#   4821    4831#   4837#   4847#   4864#   4867    4873#
                4876    4897#   4910#   4914    4925#   4929    4951#   4955    4967#   4971    5067#   5071    5139#   5144    5186#
                5189    5216#   5219    5290#   5293    5339#   5342    5356#   5359    5698    5716    5729    5772    5824#   5827
                5857#   5921#   5924    5948#   5951    6003#   6014    6051    6057#   6095#   6118    6259#   6271    6393    6418
                6559#   6599#   6617#   6631    6903#   6907    7137#   7286#   7289    7452    7470    7872#   7889#   7890
M$TLAB             1#   1945#   3031#   3035#   3044#   3048#   3058#   3062#   3072#   3076#   3092#   3096#   3112#   3116#   3347#
                3355#   3393#   3401#   3421#   3431#   3491#   3536#   3550#   3571#   3585#   3614#   3637#   3660#   3674#   3687#
                3700#   3708#   3735#   3751#   3766#   3781#   3789#   3809#   3829#   3868#   3879#   3887#   3931#   4131#   4167#
                4313#   4356#   4446#   4478#   4483#   4488#   4493#   4499#   4510#   4520#   4532#   4539#   4554#   4574#   4612#
                4622#   4629#   4634#   4636#   4644#   4660#   4674#   4686#   4688#   4696#   4717#   4739#   4772#   4814#   4821#
                4831#   4837#   4847#   4867#   4876#   4897#   4914#   4929#   4955#   4971#   5071#   5144#   5189#   5219#   5293#
                5342#   5359#   5698#   5716#   5729#   5772#   5827#   5857#   5924#   5951#   6003#   6014#   6051#   6057#   6095#
                6118#   6259#   6271#   6393#   6418#   6559#   6599#   6617#   6631#   6907#   7137#   7289#   7452#   7470#   7872#
                7890#
M$TSTL             1#   1945#   3031#   3035#   3044#   3048#   3058#   3062#   3072#   3076#   3092#   3096#   3112#   3116#   3347#
                3355#   3393#   3401#   3421#   3431#   3491#   3536#   3550#   3571#   3585#   3614#   3637#   3660#   3674#   3687#
                3700#   3708#   3735#   3751#   3766#   3781#   3789#   3809#   3829#   3868#   3879#   3887#   3931#   4131#   4167#
                4313#   4356#   4446#   4478#   4483#   4488#   4493#   4499#   4510#   4520#   4532#   4539#   4554#   4574#   4612#
                4622#   4629#   4634#   4636#   4644#   4660#   4674#   4686#   4688#   4696#   4717#   4739#   4772#   4814#   4821#
                4831#   4837#   4847#   4867#   4876#   4897#   4914#   4929#   4955#   4971#   5071#   5144#   5189#   5219#   5293#
                5342#   5359#   5698#   5716#   5729#   5772#   5827#   5857#   5924#   5951#   6003#   6014#   6051#   6057#   6095#
                6118#   6259#   6271#   6393#   6418#   6559#   6599#   6617#   6631#   6907#   7137#   7289#   7452#   7470#   7872#
                7890#
M$WORD             1#   1945#   2003#   2012    2058#   2060    3119#   3401#   3403    3404#   4539#   4541    4542#   4636#   4688#
                4709#   4731#   4837#   4847#   4849    4850#   4897#   5698#   5699    5700    5701    5716#   5717    5718    5719
                5729#   5730    5731    5732    5772#   5773    5774    5775    5857#   5859    5860#   6014#   6015    6016    6017
                6051#   6052    6053    6054    6118#   6119    6120    6121    6271#   6272    6273    6274    6393#   6394    6395
                6396    6418#   6419    6420    6421    6631#   6632    6633    6634    7452#   7453    7454    7455    7470#   7471
                7472    7473    7915#   7925#   7930#   7935#   7939#   7941#   7945#   7947#   7989    7990
M$XFER             1#   1945#   7939#   7945#
NODCL           1951#   2777    2778    2779    2780    2781    2782    2783    2784    2785    2786    2787    2788    2789    2790
                2791    2792    2793    2794    2795    2796    2797    2801    2802    2803    2804    2805    2806    2807    2808
                2809    2810    2811    2812    2813    2814    2815    2816    2817    2818    2822    2823    2824    2825    2826
                2831    2832    2833    2834    2835    2838    2839    2840    2841    2842    2843    2844    2845    2848    2849
                2850    2851    2852    2853    2856    2857    2858    2859    2861    2862    2863    2864    2865    2867    2868
                2870    2871    2874    2877    2878    2879    2880    2881    2882    2883    2884    2885    2886    2887    2888
                2889    2890    2891    2892    2894    2895    2896    2897    2898    2899    2900    2903    2904    2905    2906
```

```
        2907    2908    2911    2912    2913    2916    2919    2920    2921    2922    2923    2924    2925    2926    2929
        2930    2931    2936    3444    3445    3446    3447    3448    3449    3450    3451    3452    3453    3454    3455
        3456    3457    3458    3459    3460    3461    3462    3463    3464    3465
OPEN       1#   1945#
POINTE     1#   1945#   1957
PRINTB     1#   1945#   3022    3039    3052    3066    3086    3105
PRINTF     1#   1945#   3343    3351    3389    3417    3427    3486    3546    3863    3873    3882    3927    4309    4352
        4550    4768    4810    4863    4872    4909    4924    4950    4966    5066    5138    5185    5215    5289    5338
        5355    5823    5920    5947    7285
PRINTS     1#   1945#   3532    3566    3577    3602    3633    3656    3666    3681    3695    3702    3728    3744    3759
        3776    3785    3805    3825    4123    4158
PRINTX     1#   1945#
READBU     1#   1945#   4531
READEF     1#   1945#   4481    4486    4491    4497
RFLAGS     1#   1945#
SETPRI     1#   1945#   4632    4672
SETVEC     1#   1945#   4607    4617    4624    4816    6902
SLASH      1#   1945#
STARS      1#   1945#
SVC        1#   1945#
XFER       1#   1945#   3119#   4636#   4688#   4709#   4731#   4837#   4897#
XFERF      1#   1945#   7944
XFERT      1#   1945#   7938


. ABS.  046636     000

 ERRORS DETECTED:  0

 CVCLHC,CVCLHC.LST/CRF/SOL=SVC34R.MLB,CVCLHC.P11
 RUN-TIME: 27 34 4 SECONDS
 RUN-TIME RATIO: 101/66=1.5
 CORE USED:  22K  (43 PAGES)
```