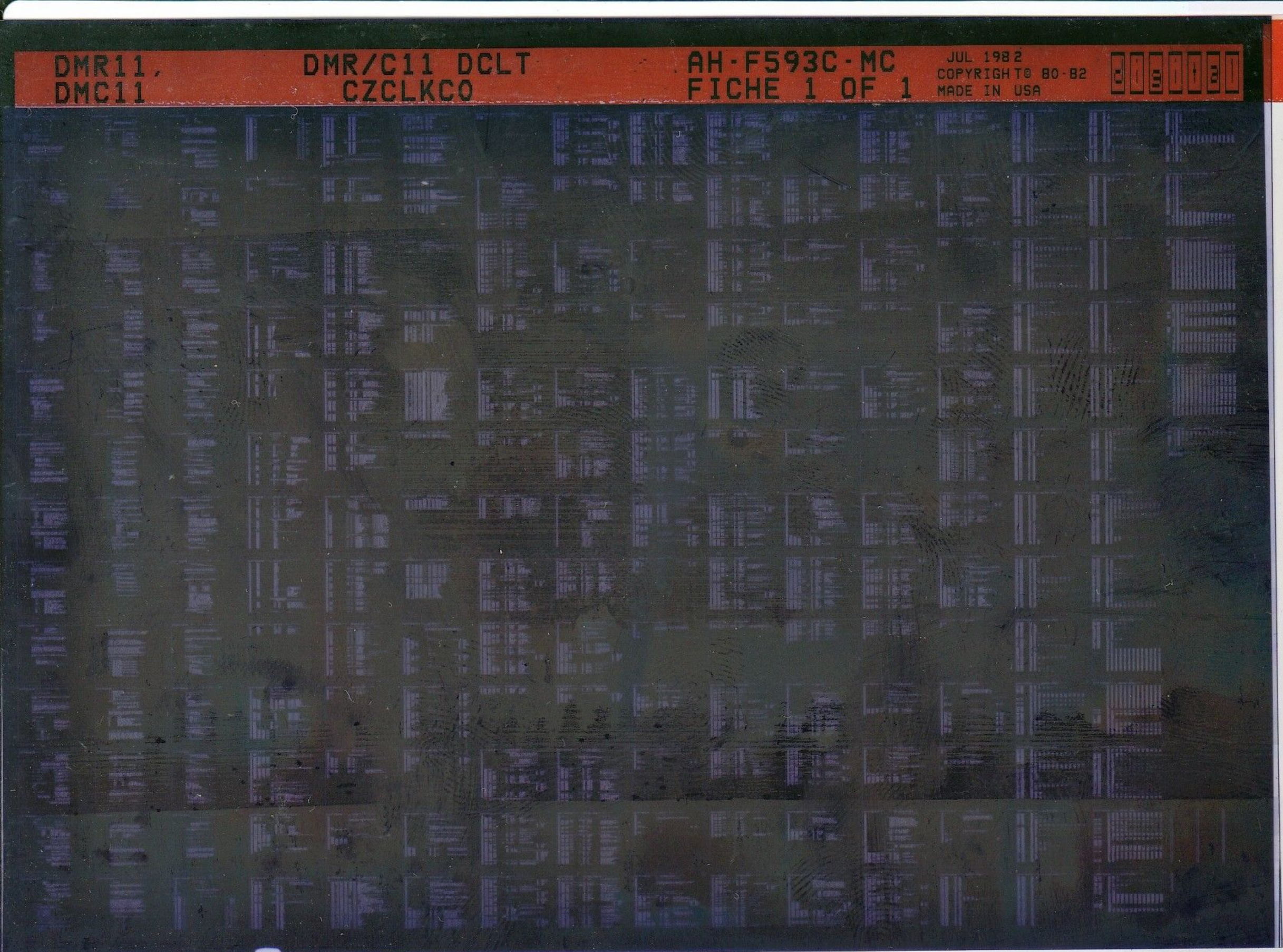
DMR11,
DMC11

DMR/C11 DCLT
CZCLKC0

AH-F593C-MC
FICHE 1 OF 1



```
    1                           .TITLE  CZCLKCO DMR,DMC-11 DATA COMM. LINK TEST
    2


                                .REM &
```

IDENTIFICATION

PRODUCT CODE: AC-F591C-MC

PRODUCT NAME: CZCLKCO DMR,DMC-11 DATA COMM. LINK TEST

PRODUCT DATE: MARCH 82

MAINTAINER: MERRIMACK DIAGNOSTIC ENGINEERING

AUTHOR: BRUCE LUHRS - BRUCE RIBOLINI

REVISION HISTORY:

REV	DATE	AUTHOR	REASON
A	23-APR-80	BRUCE LUHRS BRUCE RIBOLINI	ORIGINAL ISSUE, DCLT FOR THE DMC OR DMR-11
B	15-JUL-81	ERNIE COOPER	ADD ''MODEM/NO MODEM'' COMMAND ADD ''SET EXPECT=TRANSMIT'' COMMAND ADD ''EXIT'' COMMAND ADD ''RPT >'' COMMAND ADD PASSWORD AND ID ON DOWNLINE LOAD ADD TX / EXPECT MESSAGE TOTAL CHECK UPDATE DOCUMENTATION
C	JUNE 82	ERNIE COOPER	ADD ^C ABORT FEATURE TO EVENT LOG ADD FIX FOR AID REPORT #DD 334 (DCLT HANGS WITHOUT CLOCK) ADD CODE TO CLEAR BITS 14 AND 15 IN SEL6 IF DMC.

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT
1.2 SYSTEM REQUIREMENTS
1.3 RELATED DOCUMENTS AND STANDARDS
1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
1.5 ASSUMPTIONS - RESTRICTIONS

2.0 OPERATING INSTRUCTIONS

2.1 COMMANDS
2.2 SWITCHES
2.3 FLAGS
2.4 HARDWARE QUESTIONS
2.5 DATA COMM. LINK TEST COMMANDS

2.5.1 MESSAGE COMMANDS
2.5.2 STATISTICAL COMMANDS
2.5.3 RUN COMMANDS
2.5.4 DEFAULTS
2.5.5 PRINT COMMANDS
2.5.6 MISC COMMANDS

2.6 QUICK STARTUP PROCEDURE

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES
3.2 SPECIFIC ERROR MESSAGES

3.2.1 COMMAND LINE INTERPRETER ERRORS
3.2.2 DCLT ERRORS
3.2.3 DEVICE ERRORS

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PRINTING EVENT LOG
4.2 OPERATOR STATUS MESSAGES
4.3 PRINTING DMR,DMC-11 BASE TABLE

4.3.1 PRINTING ERROR COUNTER LOCATIONS
4.3.2 PRINTING ENTIRE BASE TABLE
4.3.3 PRINTING SINGLE LOCATION

5.0 DEVICE INFORMATION TABLES

6.0 MODE AND MESSAGE DESCRIPTIONS

6.1 MODE DESCRIPTIONS

6.1.1 TRANSMIT MODE
6.1.2 RECEIVE MODE
6.1.3 PASSIVE MODE
6.1.4 ACTIVE MODE
6.1.5 DOWN-LINE LOAD MODE
6.1.6 TALK MODE
6.1.7 LISTEN MODE
6.1.8 MAINTENANCE MODE

6.2 MESSAGE DESCRIPTIONS

7.0 OTHER INFORMATION

7.1 INTERFACING TO AN "ITEP" NODE
7.2 TROUBLESHOOTING HINTS

7.2.1 INTERNAL LOOP AT EACH NODE
7.2.2 TRANSMIT ON ONE NODE-RECEIVE ON THE OTHER
7.2.3 ONE NODE ACTIVE-THE OTHER NODE PASSIVE
7.2.4 BOTH NODES ACTIVE
7.2.5 TALK AND LISTEN MODES FOR COMMUNICATIONS

7.3 EXAMPLES OF COMMANDS

7.3.1 MESSAGES COMMANDS
7.3.2 STATISTICAL COMMANDS
7.3.3 RUN COMMANDS
7.3.4 PRINT COMMANDS
7.3.5 EXIT COMMAND

7.4 THINGS TO WATCH OUT FOR

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 DMR/DMC-11 TO DMR/DMC-11 AND OTHER (POINT TO POINT) DDCMP SUPPORTED COMMUNICATION LINKS. THIS DCLT PROGRAM WILL PROVIDE THE COVERAGE NECESSARY TO DETECT FAILURES TO THE COMPUTER EQUIPMENT, THE COMMUNICATION LINK, OR THE MODEM.

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

1.2 SYSTEM REQUIREMENTS

IN ORDER TO RUN THE DMR/DMC-11 DCLT PROGRAM, THE FOLLOWING MINIMUM HARDWARE IS REQUIRED:

- A PDP-11 CPU
- MINIMUM OF 24K WORDS OF MEMORY
- A WORKING, LINE OR REAL-TIME CLOCK
- A CONSOLE TERMINAL
- ANY XXDP+ SUPPORTED LOAD MEDIA
- ONE OF THESE DMR-11 OR DMC-11 CONFIGURATIONS:
  - DMC11-AL - LOCAL MICROPROCESSOR
  - DMC11-AR - REMOTE MICROPROCESSOR
  - DMC11-DA - E.I.A. LINE UNIT
  - DMC11-FA - CCITT V.35 LINE UNIT
  - DMC11-MA - 1M BPS LINE UNIT
  - DMC11-MD - 56K BPS LINE UNIT

  - DMR11-AA - E.I.A. (RS 232/423)
  - DMR11-AB - CCITT V.35
  - DMR11-AC - LOCAL
  - DMR11-AE - E.I.A. (RS 422)

IF DOWN-LINE-LOADING A DMC-11 SATELLITE, THE SATELLITE END REQUIRES:
  - M9301-YJ/M9312 - BOOTSTRAP MODULE

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 DMR OR DMC-11'S AT EACH END OF THE LINK HAVE ALREADY BEEN TESTED.

IF NO LINE OR REAL-TIME CLOCK IS 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 DMR OR DMC-11, BUT TO TEST THE COMMUNICATION LINK TO WHICH THEY ARE CONNECTED.

SOME OF THE DIAGNOSTICS THAT COULD BE RUN IF THE DMC-11 OR DMR-11 LOOKS BAD:

DMR:
- CZDMIXX DMR-11 FCTNL DIAG
- CZDMPXX M8207 STATIC DIAG #1
- CZDMQXX M8207 STATIC DIAG #2
- CZDMRXX M8203 STATIC DIAG #1
- CZDMSXX M8203 STATIC DIAG #2

DMC:
- CZDMCXX BSC W/R MICRO-PROC TST
- CZDMEXX DDCMP MDLN UNIT TST
- CZDMGXX DMC-11 CROM + JMUP TEST
- MD-11-DZDMHXX DMC-11 FREE RUNNING TEST

XX = LATEST REVISION

1.5 ASSUMPTIONS - RESTRICTIONS

IT IS ASSUMED THAT THE COMMUNICATIONS DEVICE (DMC OR DMR-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.

BECAUSE THE DMC-11 AND DMR-11 SUPPORT DDCMP OPERATION IN THE FIRMWARE, THE PDP-11 D.C.L.T. PROGRAM IS UNABLE TO CONTROL OR KNOW EXACTLY WHAT IS BEING TRANSMITTED AT ANY GIVEN TIME. ALL DATA MESSAGES ARE ENCLOSED IN A DDCMP ENVELOPE AND THERE MAY ALSO BE CONTROL MESSAGES (AKS, NAKS,.....) BEING TRANSMITTED. BECAUSE OF THIS PLEASE BEWARE IF IF YOU ARE SCOPING DATA.

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

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

UNIT 0
FULL DUPLEX OPERATION :  (L) Y ?
DMR,DMC-11 CSR ADDRESS : (O)  160170 ?
INTERRUPT VECTOR ADDRESS:  (O)  300  ?
INTERRUPT PRIORITY : (O)   5  ?
DEVICE OPTION TYPE : (0=DMC, 5=DMR-DMC MODE ,7=DMR) (O)  0 ?
```

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 (000'S) AND THEN PUTS DEFAULT ITEP MSG IN SO NOT REALLY EMPTY
CLEAR TRANSMITLIST	ZEROES 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	EXPECTLIST=TRANSMITLIST	MAKES A COPY OF THE TRANSMIT LIST IN THE EXPECT LIST.
SET	TRANSMITMSG=TYPE/QUAL	DEFINE A MESSAGE TO BE PUT ON THE TRANSMIT LIST (SEE DESCRIPT FOR SET EXP)
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. FROM HERE YOU CAN EXAMINE THE EVENT LOG OR BASE TABLE.
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	

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	(FORCES /NOECHO ,NO LOOPING, /NOCHECK, (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. NOTE: THIS SWITCH CAUSES NO ACTION IN THIS DCLT PROGRAM BUT IT IS INCLUDED BECAUSE IT IS USED IN OTHER DCLT PROGRAMS.
/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
                         =CABLE
                         =LOCALMODEM (DMR IN DMR MODE AND RS449 MODEMS ONLY.
                                CAUSES A "WRITE MODEM" TO BE DONE TO SET UP
                                LOCAL-LOOPBACK (MAINT1) . ALSO CALLED
                                ANALOG-LOOPBACK.

                         =REMOTEMODEM (DMR IN DMR MODE AND RS449 MODEMS ONLY.
                                CAUSES A "WRITE MODEM" TO BE DONE TO SET UP
                                REMOTE-LOOPBACK (MAINT2) . ALSO CALLED
                                DIGITAL-LOOPBACK.

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

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


        2.5.4   DEFAULTS
                ---------

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

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

        FOR THE RUN COMMAND THE DEFAULTS ARE:

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

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


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

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

OTHER NOTES:

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

2.5.5 PRINT

THE PRINT COMMAND TAKES YOU A LEVEL BELOW DCLT> CALLED REPORT.
THE COMMANDS AVAILABLE IN RPT> ARE ...

COMMAND	DESCRIPTION
HELP OR ?	PRINT HELP INFORMATION FOR RPT>
LOG	PRINTS THE DCLT EVENT LOG.
BASE/FULL	PRINTS ENTIRE BASE TABLE.
BASE/ERROR	PRINTS ONLY ERROR COUNTERS IN BASE TABLE.
BASE/OFFSET=NNN	PRINTS SINGLE LOCATION IN BASE TABLE AS SPECIFIED BY OFFSET.
EXIT	RETURNS YOU TO THE LEVEL THAT YOU ENTERED FROM. (DCLT> OR DR>)

2.5.6 MISC COMMANDS

COMMAND	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 MOD=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 TRANSMIT AND EXPECT LISTS MUST BE EQUAL. IF THEY ARE NOT THIS ERROR WILL BE DISPLAYED. USE 'SE E=T' COMMAND.

3.2.2 DCLT ERROR MESSAGES:

BAD CLOCK - PROGRAM WILL HANG ON ''TIMEOUT''!!

THIS MEANS THAT EITHER NO CLOCK WAS ON THE SYSTEM OR THE ONE THAT WAS FOUND 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 MESSAGES

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.

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

TIME OUT WAITING FOR RDI TO CLEAR

```
SEL0     SEL2
XXXXXX   XXXXXX
```

THIS MEANS THAT A SOFTWARE TIMER EXPIRED BEFORE
THE DEVICE CLEARED RDI IN RESPONSE TO THE DROPPING
OF RQI.
NOTE: PROGRAM RESETS TIMER AND WAITS AGAIN
SO AN EFFECTIVE LOOP ON ERROR IS SETUP.

TIME OUT WAITING FOR RDI TO SET

```
SEL0     SEL2
XXXXXX   XXXXXX
```

THIS MEANS THAT A SOFTWARE TIMER EXPIRED BEFORE
THE DEVICE CAUSED AN INTERRUPT IN RESPONSE TO THE
PROGRAM SETING RQI.
NOTE: PROGRAM RESETS TIMER AND WAITS AGAIN
SO AN EFFECTIVE LOOP ON ERROR IS SETUP.

TIME OUT WAITING FOR RUN TO SET

```
SEL0     SEL2
XXXXXX   XXXXXX
```

THIS MEANS THAT A SOFTWARE TIMER EXPIRED BEFORE
THE DEVICE SET THE RUN BIT IN RESPONSE TO THE
PROGRAM SETING MASTER CLEAR.
NOTE: PROGRAM RESETS TIMER AND ISSUES ANOTHER
MASTER CLEAR AND WAITS AGAIN SO AN EFFECITVE
LOOP ON ERROR IS SETUP.
THIS ERROR COULD INDICATE WRONG ADDRESS FOR
DMR/DMC-11 WAS GIVEN IN HARDWARE P TABLE.

TIME OUT WAITING FOR OUTPUT INTERRUPT

```
SEL0     SEL2
XXXXXX   XXXXXX
```

THIS MEANS THAT A SOFTWARE TIMER EXPIRED BEFORE
THE DEVICE SET OUTPUT INTERRUPT IN RESPONSE TO
PROGRAM REQUESTING DEVICE TO TRANSMIT OR RECEIVE.
NOTE: PROGRAM RESETS TIMER AND WAITS AGAIN SO AN
EFFECTIVE LOOP ON ERROR IS SET UP.
THIS ERROR WILL OCCUR WHEN ONE NODE IS STARTED

IN RX OR TX MODE AND THE OTHER IS STILL BEING
SET UP. IGNORE THIS ERROR IF PROGRAM CONTINUES
WITHOUT FURTHER ERRORS.

INPUT INTERRUPT WHEN EXPECTING OUTPUT
SEL0 SEL2
XXXXXX XXXXXX

THIS WILL HAPPEN IF THE DEVICE IS BAD. IT MEANS
THAT AFTER THE PROGRAM HAS ISSUED ALL INPUT REQUESTS
TO THE DEVICE, THE DEVICE ISSUES AN INPUT INTERRUPT

ILLEGAL OUTPUT INTERRUPT
SEL2 SEL6
XXXXXX XXXXXX

THIS HAPPENS WHEN THE DEVICE ISSUES AN OUTPUT INTERRUPT
WITHOUT SETTING ''RDO''. IF THIS HAPPENS THE DEVICE IS BAD.

CONTROL OUT INSTEAD OF BA-CC OUT
SEL2 SEL6
XXXXXX XXXXXX MMMMMMM

WHERE 'MMMMM'' IS ONE OF THE FOLLOWING MESSAGES
THAT RESULT FROM INTERPRETING THE REGISTER CONTENTS
FOR YOU:

PROCEDURE ERROR/HALT
NON EXIST MEM
DDCMP START REC
DISCONNECT
LOST DATA
DDCMP MAINT REC
OVERRUN
TIME OUT
DATA CHECK
RUN SET ILLEAGLLY (DMR IN DMR-MODE ONLY)
CD GLITCHED (DMR IN DMR-MODE ONLY)
RX IDLE (DMR IN DMR-MODE ONLY)
CTS FALILED (DMR IN DMR-MODE ONLY)

THIS ERROR OCCURS WHEN THE DEVICE SETS CONTROL OUT
TO INDICATE ERROR CONTIDION. THE PROGRAM EXPECTS A
BACC OUT.

TX BUFF COMPLETED AND SHOULD BE RX
SEL4 SEL6
XXXXXX XXXXXX

THIS ERROR OCCURS WHEN THE THE DEVICE HAS
A BACC OUT WITH TX COMPLETED AND THE PROGRAM
WAS EXPECTING A RX COMPLETED.

RX BUFF COMPLETED AND SHOULD BE TX
SEL4 SEL6
XXXXXX XXXXXX

THIS ERROR OCCURS WHEN THE THE DEVICE HAS
A BACC OUT WITH RX COMPLETED AND THE PROGRAM
WAS EXPECTING A TX COMPLETED.

WHERE "XXXXX" IS THE OCTAL CONTENTS OF THAT
DEVICE REGISTER.

DOWN LINE LOAD ABORTED

THIS ERROR CAN ONLY OCCUR IN A NODE THAT
IS A DLL "HOST" WHEN IT HAPPENS IT ALSO
PRINTS ONE OF THE FOLLWING QUALIFERS:

TX NOT COMPLETE
HOST DEVICE DID NOT GIVE BACC OUT TX
THIS SHOULD NOT HAPPEN BECAUSE DEVICE
DOES NOT NEED AN ACK FOR MAINT MESGS.

RX NOT COMPLETE
HOST DEVICE DID NOT GIVE BACC OUT RX
THIS CAN HAPPEN IF SATELLITE DOES NOT
SEND THE SEC BOOT REQUEST MESSAGE.

SEC REQ WORD1
HOST RECEIVED A MESSAGE FROM SATELLITE
BUT MESSAGE WAS NOT 1ST WORD OF SEC BOOT REQ.

SEC REQ WORD2
HOST RECEIVED A MESSAGE FROM SATELLITE
BUT MESSAGE WAS NOT 2ND WORD OF SEC BOOT REQ.

CALLED FROM PC. XXXXXX

THIS MESSAGE OCCURS WITH OTHER ERROR MESAGES
TO INDICATE PC OF CALLING ROUTINE.

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
GIVE 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,
NO. OF "NOBUFF'S"(NO. OF CONTROL-OUTS WITH THE
NO-BUFFER SET SINCE THE LAST "DCLT RUN" COMMAND.)

NOTE: IF THE NODES ON THE LINK ARE SIMILAR WITH
RESPECT TO CONSOLE SPEED AND SETUP, THE
NUMBER OF "NOBUFFS" SHOULD BE NEAR ZERO.

4.2 OPERATOR STATUS MESSAGES

THE "/STATUS, /NOSTATUS" QUALIFIERS 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

4.3 PRINTING OF DMR/DMC-11 BASE TABLE

AT THE "DCLT>" OR "DR>" LEVEL, GIVE THE PRINT COMMAND. THIS WILL TAKE YOU TO THE "RPT>" LEVEL. YOU NOW HAVE THE OPTION OF PRINTING ONLY ERROR LOCATIONS, ENTIRE BASE TABLE OR A SINGLE LOCATION. YOU ONLY HAVE TO INPUT ENOUGH OF THE COMMAND TO MAKE IT UNIQUE. THE ENTIRE BASE TABLE IN LOCAL PDP11 MEMORY IS UPDATED BY THE DMC OR DMR, WHENEVER A FATAL ERROR OCCURS. THE ERROR COUNTER LOCATIONS OF THE BASE TABLE ARE UPDATED EVERY SECOND BY THE DMC OR DMR IN DMC MODE. IF THE DMR IS IN DMR MODE, THE ENTIRE BASE TABLE WILL BE UPDATED AT "END OF DCLT PASS".

4.3.1 PRINTING ERROR LOCATIONS

TO PRINT ERROR COUNTER LOCATIONS, INPUT "BASE/ERROR". FOR A DMC LOCATIONS BASE+3..BASE+12 WILL BE DISPLAYED. IF A DMR, LOCATIONS BASE+3..BASE+41 WILL BE DISPLAYED. THE BASE ADDRESS IN THIS PROGRAM IS ALWAYS 17370.

EXAMPLE - DEVICE IS DMC

```
RPT> (A) ? B/E

LOCATION     CONTENTS        DESCRIPTION
17373         004        NAKS-MSG NO BUFFERS CUMUL
  .            .                 .
  .            .                 .
17402         007        REPS RECD CUMUL
```

4.3.2 PRINTING ENTIRE BASE TABLE

TO PRINT THE ENTIRE BASE TABLE, INPUT "BASE/FULL". IF A DMC 256 BYTES WILL BE DISPLAYED. IF A DMR, 128 BYTES WILL BE DISPLAYED. IN ORDER TO SAVE PROGRAM SPACE IN MEMORY, NOT ALL LOCATIONS WILL HAVE A DESCRIPTIVE MESSAGE. WHEN IN DOUBT SEE THE DMC OR DMR TECHNICAL MANUALS FOR A FULL DESCRIPTION.

EXAMPLE - DEVICE IS DMR IN DMC MODE;

```
RPT> (A) ? BASE/FULL

LOCATION     CONTENTS          DESCRIPTION
17370          000        BASE TABLE UPDATE INDEX POINTER
17371          000        BASE TABLE UPDATE LIMIT
17372          000        BEGINNING OF BASE TABLE DATA
17373          000        NAKS RCVD..BUFFER TEMP UNAVAILABLE
  .             .           .....
  .             .           .....
17567           .         SEE DMR MANUAL FOR DESCRIPTION
```

4.3.3 PRINTING SINGLE LOCATION

TO EXAMINE A SINGLE LOCATION, INPUT "BASE/OFFSET=NNN". FOR A DMC NNN IS A OCTAL NUMBER BETWEEN 0-377. FOR A DMR, NNN IS A OCTAL NUMBER BETWEEN 0-177. IF THE OFFSET VALUE IS NOT WITHIN THIS RANGE AN ERROR MESSAGE WILL BE PRINTED.

EXAMPLE - DEVICE IS DMR

```
RPT> (A) ? B/O=27

LOCATION        CONTENTS              DESCRIPTION
17417             006          STREAMING TIME OUT COUNT

RPT> (A) ?
```

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] INTERRUPT PRIORITY (5)
.WORD   0               ;[10] SPARE
.WORD   0               ;[12] OPTION TYPE(0=DMC,5=DMR-DMC MODE,7=DMR)
```

6.0 MODE AND MESSAGE DESCRIPTIONS

6.1 MODE DESCRIPTIONS

BECAUSE THE DMC-11 AND DMR-11 SUPPORT DDCMP OPERATION IN THE
FIRMWARE, THE PDP11 DCLT PROGRAM IS UNABLE TO CONTROL OR KNOW
EXACTLY WHAT IS BEING TRANSMITTED OR RECEIVED AT ANY GIVEN TIME.
ALL DATA MESSAGES ARE ENCLOSED IN A DDCMP ENVELOPE AND THEREFORE
CONTROL MESSAGES (ACKS,NAKS...) ARE ALSO BEING TRANSMITTED AND
RECEIVED.

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

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

THE "HOST" OR ORIGINATING STATION REQUESTS THE "SATELLITE" OR
BOOT STATION TO ENTER MOP MODE. THE SATELLITE THEN SENDS A
"SECONDARY BOOT REQUEST MESSAGE". THE "HOST" THEN CHECKS THE
RECEIVED MESSAGE TO SEE THAT IT IS A "SECONDARY BOOT REQUEST".
THEN THE HOST SENDS A "MEMORY LOAD WITH TRANSFER ADDRESS"
THAT CONTAINS IMAGE DATA TO BE LOADED BY THE SATELLITE'S
M9301-YJ/M9312 STARTING AT LOC. 0. THIS IMAGE DATA WILL CONTAIN A
CODE THAT PRINTS A MESSAGE SAYING DOWN-LINE-LOAD WAS SUCESSFUL.
THE BOOTING PROCESS OVERWRITES PART OF THE "VECTOR" AREA SO THE DCLT
PROGRAM MUST BE RELOADED IN THE "SATELLITE" SYSTEM.

THE SATELLITE WILL ENTER MOP MODE ONLY IF THE PASSWORD WORD
SUPPLIED BY THE USER MATCHES THAT SET IN ITS PASSWORD SWITCH PACK.
INCLUDED IN THE "SECONDARY BOOT MESSAGE", IS THE DEVICE TYPE CODE
THAT IS DECIPHERED AND INCLUDED IN AN IDENTIFICATION MESSAGE.

EXAMPLE DOWNLINE LOAD;

```
DCLT>R M=D
SATTELITE PASSWORD = NNN    ;NNN = OCTAL # BETWEEN 0-376
SECONDARY BOOT REQ FROM XXX DEVICE TYPE = YY
```

YY	XXX
0	DP
2	DU
4	DL
6	DQ
8	DA
10	DUP
12	DMC
14	DN
16	DLV
18	DMP
20	DTE

22	DV
24	DZ
28	KDP
30	KDZ
32	KL
34	DMV

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	THE "LU LOOP" BIT IS SET SO THAT THE UNIT'S SERIAL LINE OUT IS LOOPED BACK TO THE SERIAL LINE IN AT THE TTL LEVEL BEFORE LEVEL CONVERSION.
CABLE	NOT USED BY DMR OR DMC-11 CODE.
LOCALMODEM	FOR DMR-11 IN DMR MODE AND RS449 MODEMS ONLY. CAUSES A "WRITE MODEM" TO BE DONE TO SET UP LOCAL-LOOPBACK (MAINT1) . ALSO CALLED ANALOG-LOOPBACK.
REMOTEMODEM	FOR DMR-11 IN DMR MODE AND RS449 MODEMS ONLY. CAUSES A "WRITE MODEM" TO BE DONE TO SET UP REMOTE-LOOPBACK (MAINT2) . ALSO CALLED

DIGITAL-LOOPBACK.

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):

STATION A "HOST" NODE	"/LOOP" ALLOWED?	STATION B "REMOTE" NODE	DUPLEX
TALK	NO	LISTEN*, RECEIVE	HALF OR FULL
LISTEN	NO	TALK*, TRANSMIT	HALF OR FULL
TRANSMIT	NO	RECEIVE*, LISTEN	HALF OR FULL
RECEIVE	NO	TRANSMIT*, TALK	HALF OR FULL
PASSIVE	NO	ACTIVE*	HALF OR FULL
ACTIVE	YES	ACTIVE*	FULL
		PASSIVE*	HALF OR FULL
DOWNLINELOAD	NO	PASSIVE	HALF FORCED

*= MOST LIKELY TO BE IN THAT MODE

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/[\]^_%)
"A-Z,0-9,SPACES,TABS"	THESE ARE THAT THE CHARACTERS THAT CAN BE TYPED BETWEEN QUOTATION MARKS ("..") TO SPECIFIY A UNIQUE MESSAGE. (CALLED AN OPERATOR SPECIFIED MESSAGE.)

7.0 OTHER INFORMATION

7.1 INTERFACING TO AN "ITEP" NODE

WHEN DCLT IS USED TO INTERFACE TO AN ITEP NODE.
THE TABLE BELOW APPLIES:

ITEP NODE	DCLT NODE
ONE-WAY-OUT	RECEIVE OR LISTEN
ONE-WAY-IN	TRANSMIT OR TALK
INTERNAL LOOP	ACTIVE
EXTERNAL LOOP	ACTIVE OR PASSIVE

NOTE: WHEN INTERFACING TO ITEP IF THE RX BUFFER ON THE
ITEP SIDE IS ONLY 10 BYTES LARGER THAN THE TX BUFFER YOU
HAVE SELECTED, SO BE SURE TO SET THE TX BUFFER ON THE DCLT
NODE ACCORDINGLY.

WHEN ITEP IS IN A MODE THAT IT IS EXPECTING TO BE TRANSMITTED
TO, A SOFT ERROR "BASE TABLE ERR COUNTS NON-ZERO" WILL OCCUR.
THIS IS DUE TO THE SPEED DIFFERENCES IN THE SOFTWARE.

WHEN DCLT IS IN LISTEN MODE THE RX BUFFER IS ONLY
82 BYTES LONG THEREFORE DO NOT SEND THE DCLT NODE
ITEP MSG. 3 FROM THE ITEP NODE OR A "LOST DATA" ERROR WILL
OCCUR

BE SURE ITEP NODE HAS INCORPERATED PATCH FROM DEPO# MD-11-DZDMO-A1

ITEP NODE SHOULD ALWAYS BE RUN WITH SW 4 = TO 0

7.2 TROUBLESHOOTING HINTS

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

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

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

7.2.1 INTERNAL LOOP AT EACH NODE

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

A POSSIBLE COMMMAND SEQUENCE IS:

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

WHAT THE ABOVE COMMAND SEQUENCE MEANS:

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

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

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

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

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

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

WHAT THIS SEQUENCE MEANS:

THIS SEQUENCE IS SIMILAR TO THE ONE ABOVE . THE "SE E=T"
MAKES A COPY OF THE TRANSMIT LIST IN THE EXPECT LIST.
THE EXPECT LIST NOW CONTAINS 3 MESSAGES. THE MESSAGES WILL
HAVE ALL ONES FOR DATA AND BE 20 BYTES EACH IN LENGTH.

THE RUN COMMAND IS THE SAME WITH THE ADDITION OF TWO
SWITCHES "/CH/PAS=3". THE "CH" SWITCH TELLS THE PROGRAM
TO CHECK THE RECEIVED DATA AGAINST THE "EXPECTED LIST".
THE "PAS=3" SWITCH TELLS THE PROGRAM TO RUN 3 PASSES BEFORE
RETURNING TO THE DCLT> PROMPT.

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

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

IF A CABLE TURNAROUND CONNECTOR IS AVAILABLE, PUT IT ON THE END OF
THE CABLE JUST BEFORE THE MODEM AND RUN IN ACTIVE MODE WITH NO LOOP.
POSSIBLE COMMAND SEQUENCE IS:

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

WHAT THIS SEQUENCE MEANS:

THIS SEQUENCE HAS THE "/LO=I" REMOVED. THIS INFORMS
THE DEVICE TO ACT AS IF IT WAS RECEIVING FROM ANOTHER
NODE.

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/PASS=00000
                 /STATUS/CHECK/NOECHO/NOMODEM
DCLT> (A)  ?
```

7.2.2 TRANSMIT ON ONE NODE RECEIVE ON THE OTHER

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

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

```
                R M=TR/PAS=3

WHAT THIS SEQUENCE MEANS:

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

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

FOR NODE A:

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

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


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

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

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

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

NODE A: IS THE SAME AS ABOVE.

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

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

7.2.3 ONE NODE ACTIVE THE OTHER NODE PASSIVE

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

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

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

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

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

FOR NODE A:

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

FOR NODE B:

```
INI RXQ TXQ TXC RXQ TXQ TXC RXQ
TXQ TXC EOP RXQ TXQ TXC RXQ TXQ
TXC EOP RXQ TXQ TXC RXQ TXQ TXC
```

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

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

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

7.2.4 BOTH NODES ACTIVE

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

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

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

WHAT YOU SHOULD SEE AFTER ENTERING THE RUN COMMAND IF THINGS ARE RUNNING CORRECTLY :
ON BOTH NODES A AND B:

```
MODE=ACTIVE/PASS=00000
            /NOSTATUS/CHECK/NOECHO/NOMODEM
```

```
DCLT> (A)  ?
```

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

7.2.5 TALK AND LISTEN MODES FOR COMMUNICATING

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

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

7.3 EXAMPLES OF COMMANDS

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

7.3.1 EXAMPLES OF MESSAGES COMMANDS

THE CLEAR COMMANDS .

```
C E
C T
```

THIS WILL INITIALIZE THE TRANSMIT AND EXPECT LIST
TO 1 MESSAGE OF 58 BYTES. THE DATA OF THE MESSAGE WILL
BE THE ITEP MESSAGE.

IF THESE COMMANDS ARE FOLLOWED BY A SHOW COMMAND

```
SH E
```

SUCH AS THE SHOW EXPECT LIST. WHAT YOU WOULD SEE IS

```
MSG: TYPE=ITEP/SIZE=58
     MODE=ACTIVE/PASS=00001
               /NOSTATUS/CHECK/NOECHO/NOMODEM
DCLT> (A) ?
```

NOW IF YOU DID A SET EXPECT LIST COMMAND SUCH AS:

```
SE E=A/S=35/C=3
```

AND FOLLOWED IT WITH A SHOW EXPECT LIST COMMAND

```
SH E
```

WHAT YOU WOULD SEE IS

```
MSG: TYPE=ALPHA/SIZE=35
MSG: TYPE=ALPHA/SIZE=35
MSG: TYPE=ALPHA/SIZE=35
MSG: TYPE=ALPHA/SIZE=35
     MODE=ACTIVE/PASS=00001
               /NOSTATUS/CHECK/NOECHO/NOMODEM
DCLT> (A) ?
```

```
7.3.2 EXAMPLES STATISTICAL COMMANDS

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

DCLT> (A)  ?

THE SAME WILL HAPPEN IF YOU USE THE ?

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

41260   104423  000167 177772 021122 012112 006312 006312 006312
41300   006312
IF YOU HAD USED THE /B SWITCH
        DUM 41260-41300/B
WHAT YOU WOULD SEE IS
        41260   023 211 167 000 372 377 122 024
        41270   112 024 312 014 312 014 312 014
        41300   312


7.3.3 EXAMPLES RUN COMMMANDS

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

IF YOU WERE TO EXECUTE THE RUN COMMAND
        R M=TR/NOST/CH/PAS=4
WHAT WOULD HAPPEN IS AFTER 4 PASSES THE PROGRAM WOULD RETURN
TO THE DCLT PROMPT AND PRINT
        MODE=TRANSMIT/PASS=00000
                    /NOSTATUS/CHECK/NOECHO/NOMODEM
  DCLT> (A)  ?
IF YOU WERE TO EXECUTE THE RUN COMMAND
        C E
        C T
        R M=A/LO=I/ST/CH/PAS=3
WHAT YOU WOULD SEE (IF USING DEFAULT TRANSMIT AND EXPECT
```

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

 IF YOU USE THE EXIT COMMAND
        EXIT
 WHAT YOU WOULD SEE IS
        CZCLK EOP
             0 CUMLATIVE ERRORS

        DR>
```

7.3.4 EXAMPLES PRINT COMMANDS

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

```
        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
BASE/ERROR - PRINT ONLY ERRORS
BASE/FULL - PRINT ENTIRE TABLE
BASE/OFFSET=NNN - PRINT SINGLE LOCATION

RPT> (A)  ? LOG

THE LOG COMMAND PRODUCES THE FOLLOWING

>>> DCLT EVENT LOG ENTRY >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
    0:01:17         ^C ABORT
   PASS = 12       ERRORS = 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   1
>>> 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   1
                              .....
                              .....
                              .....

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

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.

```
                                .SBTTL  PROGRAM HEADER

002000                                  BGNMOD



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

002000                                  POINTER BGNRPT,BGNAU,BGNDU


002000                                  HEADER  CZCLK,C,0,1800.,0,#PRI07
002000                                                                  L$NAME::
002000  103                                                                     .ASCII /C/
002001  132                                                                     .ASCII /Z/
002002  103                                                                     .ASCII /C/
002003  114                                                                     .ASCII /L/
002004  113                                                                     .ASCII /K/
002005  000                                                                     .BYTE   0
002006  000                                                                     .BYTE   0
002007  000                                                                     .BYTE   0
002010                                                                  L$REV::
002010  103                                                                     .ASCII  /C/
002011                                                                  L$DEPO::
002011  060                                                                     .ASCII  /0/
002012                                                                  L$UNIT::
002012  000000                                                                  .WORD   0
002014                                                                  L$TIML::
002014  003410                                                                  .WORD   1800.
002016                                                                  L$HPCP::
002016  046140                                                                  .WORD   L$HARD
002020                                                                  L$SPCP::
002020  000000                                                                  .WORD   0
002022                                                                  L$HPTP::
002022  002130                                                                  .WORD   L$HW
002024                                                                  L$SPTP::
002024  000000                                                                  .WORD   0
002026                                                                  L$LADP::
002026  046522                                                                  .WORD   L$LAST
002030                                                                  L$STA::
002030  000000                                                                  .WORD   0
002032                                                                  L$CO::
002032  000000                                                                  .WORD   0
002034                                                                  L$DTYP::
002034  000000                                                                  .WORD   0
002036                                                                  L$APT::
002036  000000                                                                  .WORD   0
002040                                                                  L$DTP::
002040  002124                                                                  .WORD   L$DISPATCH
```

```
2050  002042
2051  002042  000340
2052  002044
2053  002044  000000
2054  002046
2055  002046  000000
2056  002050
2057  002050     003
2058  002051     003
2059  002052
2060  002052  000000
2061  002054  000000
2062  002056
2063  002056  000000
2064  002060
2065  002060  012414
2066  002062
2067  002062  033732
2068  002064
2069  002064  000000
2070  002066
2071  002066  000000
2072  002070
2073  002070  034746
2074  002072
2075  002072  034740
2076  002074
2077  002074  000000
2078  002076
2079  002076  012430
2080  002100
2081  002100  104035
2082  002102
2083  002102  000000
2084  002104
2085  002104  033746
2086  002106
2087  002106  034650
2088  002110
2089  002110  034646
2090  002112
2091  002112  033740
2092  002114
2093  002114  000000
2094  002116
2095  002116  000000
2096  002120
2097  002120  000000
2098
2099
```

```
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
```

```
                                .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
002124                                                          L$DISPATCH::
002124  034754                                                          .WORD   T1
```

```
                                .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] INTERRUPT PRIORITY (5)
002140  000000                          .WORD   0               ;[10] DEVICE PARAMETERS WORD
                                                                ;   (ENABLE CRC, STRIP SYNC, COMPATIBLE MODE...)
002142  000000                          .WORD   0               ;[12] DEVICE OPTION TYPE(0=DMC,5=DMR-DMC MODE,
                                                                ; 7=DMR.
002144  000004                          .WORD   4               ;[14] BAUD RATE (0=2.4K, 1=4.8K, 2=9.6K, 3= 19.2K,
                                                                ; 4=56K, 5=250K, 6=500K, 7=1 MEGA-BAUD)
002146  000000                          .WORD   0               ;[16] LINE INTERFACE (422, V.35, INT, EIA...)

002150                                  ENDHW
002150                                                                  L10000:
```

```
                                .SBTTL  GLOBAL EQUATES SECTION


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

002150                                  EQUALS
                                ;
                                ; BIT DIFINITIONS
                                ;
        100000                  BIT15== 100000
        040000                  BIT14== 40000
        020000                  BIT13== 20000
        010000                  BIT12== 10000
        004000                  BIT11== 4000
        002000                  BIT10== 2000
        001000                  BIT09== 1000
        000400                  BIT08== 400
        000200                  BIT07== 200
        000100                  BIT06== 100
        000040                  BIT05== 40
        000020                  BIT04== 20
        000010                  BIT03== 10
        000004                  BIT02== 4
        000002                  BIT01== 2
        000001                  BIT00== 1
                                ;
        001000                  BIT9==  BIT09
        000400                  BIT8==  BIT08
        000200                  BIT7==  BIT07
        000100                  BIT6==  BIT06
        000040                  BIT5==  BIT05
        000020                  BIT4==  BIT04
        000010                  BIT3==  BIT03
        000004                  BIT2==  BIT02
        000002                  BIT1==  BIT01
        000001                  BIT0==  BIT00
                                ;
                                ; EVENT FLAG DEFINITIONS
                                ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                ;
        000040                  EF.START==      32.             ; START COMMAND WAS ISSUED
        000037                  EF.RESTART==    31.             ; RESTART COMMAND WAS ISSUED
        000036                  EF.CONTINUE==   30.             ; CONTINUE COMMAND WAS ISSUED
        000035                  EF.NEW==        29.             ; A NEW PASS HAS BEEN STARTED
        000034                  EF.PWR==        28.             ; A POWER-FAIL/POWER-UP OCCURRED
                                ;
```

```
2210                                    ;
2211                                    ; PRIORITY LEVEL DEFINITIONS
2212                                    ;
2213            000340                  PRI07== 340
2214            000300                  PRI06== 300
2215            000240                  PRI05== 240
2216            000200                  PRI04== 200
2217            000140                  PRI03== 140
2218            000100                  PRI02== 100
2219            000040                  PRI01== 40
2220            000000                  PRI00== 0
2221                                    ;
2222                                    ;OPERATOR FLAG BITS
2223                                    ;
2224            000004                  EVL==       4
2225            000010                  LOT==      10
2226            000020                  ADR==      20
2227            000040                  IDU==      40
2228            000100                  ISR==     100
2229            000200                  UAM==     200
2230            000400                  BOE==     400
2231            001000                  PNT==    1000
2232            002000                  PRI==    2000
2233            004000                  IXE==    4000
2234            010000                  IBE==   10000
2235            020000                  IER==   20000
2236            040000                  LOE==   40000
2237            100000                  HOE==  100000
2238
```

```
                        ;***** INDEPENDENT EQUATES
                        
001000                          BUFLIM=512.             ;MAX BUFFER SIZE IN BYTES
                                                        ;  APPLIES TO TX,RX AND CMP BUFFS
000017                          MSGLIM=15.              ;MAX NO. OF MESSAGES PER BUFFER
                                                        ;  (FOR EACH INCREMENT (+1) TO MSGLIM,
                                                        ;   ADD 6 WORDS TO THE POINTER TABLE
                                                        ;   (PTRTAB:) SINCE THIS MEANS 2 MORE
                                                        ;   "POINTER" WORDS PER BUFFER.
                        
                        ;MODE OF OPERATION EQUATES
000000                          REC=0                   ;RECEIVE MODE
000001                          TRA=1                   ;TRANSMIT MODE
000002                          PAS=2                   ;PASSIVE MODE
000003                          ACT=3                   ;ACTIVE MODE
000004                          DOW=4                   ;DOWN-LINE-LOAD MODE
000005                          TAL=5                   ;TALK MODE
000006                          LIS=6                   ;LISTEN MODE
                        ;MAINT LOOP TYPE EQUATES
000000                          NONE=   0               ;NO LOOP
000001                          TTL=    1               ;INTERNAL TTL
000002                          CABLE=  2               ;CABLE LOOP
000003                          MODLOC= 3               ;MODMEM LOCAL
000004                          MODREM= 4               ;MODEM REMOTE
000005                          MOP=    5               ;MOP
                        
                        ;CLOCK ENABLE VALUES TO BE LOADED IN CLK'S CSR
000100                          LCLKEN= 100             ;L-CLOCK CSR VALUE TO ENABLE THE CLOCK
000111                          PCLKEN= 111             ;P-CLOCK CSR VALUE TO ENABLE THE CLOCK
001600                          PCLKCT= 1600            ;P-CLOCK COUNT SET REGISTER FOR COUNTER
                        
                        ;PARAM WORD EQUATES
                        
000001                          STATB=  BIT0            ;OPERATOR AWAKE ASKED FOR
000002                          DATCKB= BIT1            ;DATA CHECK BIT
000004                          ECHOB=  BIT2            ;ECHO BIT
000010                          MOCHK=  BIT3            ;MODEM CHECK/NO CHECK   ADDED BY EC
000020                          CRCB=   BIT4            ;CRC CALCUALTE ASKED FOR
000040                          PROTOB= BIT5            ;PROTOCOL PROCESSING ASKED FOR
                        
                        ;OPTION TYPE EQUATES
                        
000000                          DMC=    0               ;DMC
000004                          DMRC6=  4               ;8206 DMR IN DMC MODE
000005                          DMRC7=  5               ;8207 DMR IN DMC MODE
000006                          DMR6=   6               ;8206 DMR IN DMR MODE
000007                          DMR7=   7               ;8207 DMR IN DMR MODE
                        
                        ;EVENT LOG MESSAGE TYPES (USED TO LOCATE EVENT DESCRIPTION IN EVENT TABLE
                        ;  AND DISPATCHING TO SEPERATE SECTIONS OF THE EVENT REPORTING SECTION)
000000                          TXQ=    0               ;TRANSMIT MESSAGE QUEUED
000002                          TXC=    2               ;TRANSMIT COMPLETE
000004                          RXQ=    4               ;RECEIVE BUFFER QUEUED
000006                          RXC=    6               ;RECEIVE COMPLETE
000010                          DER=    10              ;DEVICE INFORMATION
```

```
2295    000012                  DVI=    12              ;DEVICE ABOUT TO INIT
2296    000014                  DCK=    14              ;DATA COMPARISON RESULTS
2297
2298    000020                  DLE=    20              ;DATA COMPARISON LENGH ERROR
2299    000022                  DDE=    22              ;DATA COMPARISON DATA ERROR
2300    000024                  EOP=    24              ;END OF PASS
2301    000026                  ABO=    26              ;^C ABORT
2302
2303                    ;;;;EQUATES FOR FLAG WORD;;;;;;
2304
2305    000001                  ININT= 1                ;INPUT INT. REC.
2306    000002                  OTINT= 2                ;OUTPUT INT REC
2307    000004                  QRX=   4                ;RX QUED /COMPL
2308    000010                  QTX=    10              ;TX QUED/COMPL
2309    000020                  CTX=    20              ;TX COMPL AND IN TXSEL4 AND TSEL6
2310    000040                  CRX=    40              ;RX COMPL AND IN TSEL4 AND TSEL6
2311    000100                  ERX=    100             ;EXPECT TO GET A RX COMPLED
2312    000200                  ETX=    200             ;EXPECT TO GET A TX COMPLETED
2313    000400                  DLLGA=  400             ;DOWN LINE LOAD GO AHEAD BIT
2314    001000                  DMRRUN= 1000            ;DMR RUN MODE EXPECTED
2315    002000                  BTUP=   2000            ;BASE TABLE UPDATE REQUESTED
2316
2317                    ; SPECIAL CLI CODES FOR "CHAR" ARGUMENT IN CLI CALLS
2318                    ;  (COMMAND LINE INTERPRETER DEFINITIONS)
2319    000000                  CLIERR= 0
2320    000001                  CLIEXI= 1
2321    000002                  CLIBR=  2
2322    000003                  CLIBIF= 3
2323    000004                  CLISPA= 4
2324    000005                  CLINUM= 5
2325    000006                  CLIALP= 6
2326    000007                  CLIALN= 7
2327    000010                  CLIOCT= 8.
2328    000011                  CLIDEC= 9.
2329    000012                  CLISTR= 10.
2330
2331                    ; DEFS FOR COMMAND LINE INTERPRETATION ACTION VALUES
2332    000000                  NULL=0
2333    000001                  CLEAR=1
2334    000002                  SHOW=2
2335    000003                  CHECK=3
2336    000004                  RUN=4
2337    000005                  HLP=5
2338    000006                  CSHEXP=6
2339    000007                  CSHTRN=7
2340    000010                  SETEXP=10
2341    000011                  SETTRN=11
2342    000012                  SIZE=12
2343    000013                  QCOPY=13
2344    000014                  NUM=14
2345    000015                  OPRMSG=15
2346    000016                  STATUS=16
2347    000017                  ENDQO=17
2348    000020                  CMSG0=20
2349    000021                  CMSG1=21
2350    000022                  CMSG2=22
```

```
2351    000023                  CMSG3=23
2352    000024                  CMSG4=24
2353    000025                  CMSG5=25
2354    000026                  CMSG6=26
2355    000027                  ATVMOD=27
2356    000030                  PASMOD=30
2357    000031                  RECMOD=31
2358    000032                  LISMOD=32
2359    000033                  DLLMOD=33
2360    000034                  TRAMOD=34
2361    000035                  TALMOD=35
2362    000036                  NO=36
2363    000037                  ECHO=37
2364    000040                  CRC=40
2365    000041                  PROTO=41
2366    000042                  PASC=42
2367    000043                  MOP=43
2368    000044                  TTLLOP=44
2369    000045                  CBLLOP=45
2370    000046                  LMDLOP=46
2371    000047                  RMDLOP=47
2372    000050                  NOTNUF=50
2373    000051                  BADCHR=51
2374    000052                  DMPS=52
2375    000053                  DMPE=53
2376    000054                  DMPQ=54
2377    000055                  PRNT=55
2378    000056                  MOSC=56                 ;MODEM/NOMODEM REV B BY EC
2379    000057                  EXIT=57                 ;EXIT COMMAND REV B BY EC
2380    000060                  SETET=60                ;S E=T COMMAND REV B BY EC
2381                    ;FOLLOWING EQUATES USED IN REPORT CLI    REV B BY EC
2382    000001                  RPHLP=1                 ;HELP COMMAND
2383    000002                  RPEXT=2                 ;EXIT COMMAND
2384    000003                  RPLOG=3                 ;PRINT EVENT LOG COMMAND
2385    000004                  RPSWE=4                 ;BASE/ERROR COMMAND
2386    000005                  RPSWF=5                 ;BASE/FULL COMMAND
2387    000006                  RPSWO=6                 ;BASE/OFFSET
2388    000007                  RNOTNF=7                ;MORE COMMAND NEEDED
2389
2390                    ;***** DEVICE DEPENDENT EQUATES
2391                    ; MODEM SIGNAL BIT DEFINITONS
2392                    ;       IF SIGNAL AVAILABLE IN DEVICE, EQUATE NAME TO BIT POSITION,
2393                    ;       ELSE EQUATE IT TO = 0
2394
2395    000004                  CTS=    BIT2            ;CLEAR TO SEND (CIRCUIT CB)
2396    000010                  DSR=    BIT3            ;DATA SET READY (CIRCUIT CC)
2397    000001                  DCD=    BIT0            ;DATA CARRIER DETECT (CIRCUIT CF)
2398    000040                  RTS=    BIT5            ;REQUEST TO SEND (CIRCUIT CA)
2399    000200                  RI=     BIT7            ;RING INDICATOR (CIRCUIT CE)
2400    040000                  SQD=    BIT14           ;SIGNAL QUALITY DETECT (CIRCUIT CG)
2401    001000                  TM=     BIT9            ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)
2402
2403
2404
2405                    ; DEVICE SIGNALS
2406    000040                  RQI=    BIT5            ;REQUEST IN
```

```
2407            000200                  RDI=    BIT7            ;READY IN
2408            000200                  RDO=    BIT7
2409            000001                  BACC=   BIT0            ;BUFFER ADDR. CHAR COUNT
2410            040000                  MCLR=   BIT14           ;MASTER CLEAR
2411            004000                  LULOOP= BIT11           ;LINE UNIT LOOP(TTL)
2412            000400                  MAINTB= BIT8            ;MAINT MODE BIT
2413            002000                  HALFDB= BIT10           ;HALF DUPLEX BIT
2414            000004                  RXBIT=  BIT2            ;RX BIT
2415            000100                  IEO=    BIT6            ;ENABLE OUTPUT INTERRUPT BIT
2416
```

```
                                .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
002172  000005                  DLLM1C: .WORD   DLLM1E-DLLM1    ;DLL MSG 1 COUNT
002174  000204                  DLLM2C: .WORD   DLLM2E-DLLM2    ;DLL MSG 2 COUNT

                                ;MESSAGE ADDRESS TABLE

002176                          DMSGAD:
002176  002220                          MSG0                    ;ADDRESS OF MESSAGE #0
002200  002221                          MSG1                    ;ADDRESS OF MESSAGE #1
002202  002222                          MSG2                    ;ADDRESS OF MESSAGE #2
002204  002223                          MSG3                    ;ADDRESS OF MESSAGE #3
002206  002224                          MSG4                    ;ADDRESS OF MESSAGE #4
002210  002324                          MSG5                    ;ADDRESS OF MESSAGE #5
002212  002416                          MSG6                    ;ADDRESS OF MESSAGE #6
002214  002524                          OPBUF                   ;ADDRESS OF OPERATOR SPEC'D MSG.
002216  002646                          MSG8                    ;ADDRESS OF RECEIVE BUFFER FILL PATTERN

002220  000                     MSG0:   .BYTE   000             ;MESSAGE OF ALL 0'S
002221                          EMSG0:
002221  377                     MSG1:   .BYTE   377             ;MESSAGE OF ALL 1'S
002222                          EMSG1:
002222  252                     MSG2:   .BYTE   252             ;MESSAGE OF ALTERNATING 1'S
002223                          EMSG2:
002223  125                     MSG3:   .BYTE   125             ;MESSAGE OF ALTERNATING 0'S
002224                          EMSG3:
002224                          MSG4:                           ;''CCITT'' 512-BIT (VS. 511 BITS) TEST PATTERN
002224  177603  157427  031011          .WORD   177603,157427,031011,047321,163715,105221,143325,142304
002232  047321  163715  105221
002240  143325  142304
002244  040041  014116  052606          .WORD   040041,014116,052606,172334,105025,123754,111337,111523
002252  172334  105025  123754
002260  111337  111523
002264  030030  145064  137642          .WORD   030030,145064,137642,143531,063617,135075,066730,026575
002272  143531  063617  135075
002300  066730  026575
002304  052012  053627  070071          .WORD   052012,053627,070071,151172,165044,031605,166632,016741
002312  151172  165044  031605
```

```
 2473  002320  166632  016741
 2474  002324                                EMSG4:
 2475  002324                                MSG5:                   ;''INTERPROCESSOR TEST PROGRAM'S (ITEP)'' MESSAGE
 2476                                                                ; #1, (DP1:)
 2477  002324  077577  040444  052040              .ASCII  <177><177>/$A THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG./
 2478  002332  042510  050440  044525
 2479  002340  045503  041040  047522
 2480  002346  047127  043040  054117
 2481  002354  045040  046525  042520
 2482  002362  020104  053117  051105
 2483  002370  052040  042510  046040
 2484  002376  055101  020131  047504
 2485  002404  027107
 2486  002406  005015  077401  077577              .ASCIZ  <15><12><001><177><177><177><177>
 2487  002414  000177
 2488  002416                                EMSG5:
 2489  002416                                MSG6:                   ;ALPHA-NUMERICS (OR FUTURE COMM TURNAROUND MSG)
 2490  002416  022043  021041  023040              .ASCII  /#$!'' &'()*+,-.0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ/
 2491  002424  024047  025051  026053
 2492  002432  027055  030460  031462
 2493  002440  032464  033466  034470
 2494  002446  035472  036474  037476
 2495  002454  040500  041502  042504
 2496  002462  043506  044510  045512
 2497  002470  046514  047516  050520
 2498  002476  051522  052524  053526
 2499  002504  054530     132
 2500  002507     057  056133  057135              .ASCIZ  ?/[\]^_%?
 2501  002514  022537     000
 2502  002517                                EMSG6:
 2503        002520                                .EVEN
 2504
 2505                                ; ************************************************************************
 2506                                ;THESE THREE STORAGE AREAS MUST NOT BE SEPERATED !!!!
 2507
 2508  002520  047045  040445       OPBFPT: .ASCII  /%N%A/
 2509  002524  000122               OPBUF:  .BLKB   82.             ;BUFFER FOR OPERATOR SPEC'D MESSAGES
 2510  002646                        OPEND:
 2511
 2512                                ; THE ABOVE THREE LINES MUST BE KEPT TOGETHER
 2513                                ; ************************************************************************
 2514
 2515  002646      033               MSG8:   .BYTE   33              ;RECEIVE BUFFER FILL PATTERN
 2516  002647                        EMSG8:
 2517
 2518                                ; DOWN-LINE-LOAD MESSAGE DEFINITIONS
 2519                                        ;;;;ENTER MOP MODE MESSAGE FORMAT
 2520                                        ;;;;THE NODE WILL ENTER MAINTENANCE MODE ONLY IF THE PASSWORD MATCHES.
 2521  002647      006               DLLM1:  .BYTE   6       ;BINARY CODE FOR MAINTENANCE MODE
 2522  002650      000               PASS1:  .BYTE   0       ;PASSWORD BYTE #1   LEGAL VALUE 0 - 255
 2523  002651      000               PASS2:  .BYTE   0       ;;VALUE IN BYTE 1 IS DUPLICATED HERE
 2524  002652      000               PASS3:  .BYTE   0       ;;;AND HERE
 2525  002653      000               PASS4:  .BYTE   0       ;;;;AND HERE.
 2526  002654                        DLLM1E:                 ;END ENTER MOP MODE MESSAGE FORMAT
 2527                                        ;;;;MEMORY LOAD WITH TRANSFER ADDRESS MESSAGE FORMAT
 2528  002654      000               DLLM2:  .BYTE   0       ;CODE
```

```
2529  002655  000                             .BYTE   0       ;LOAD NUMBER
2530  002656  006                             .BYTE   6       ;LOAD ADDRESS LSB
2531  002657  000                             .BYTE   0
2532  002660  000                             .BYTE   0
2533  002661  000                             .BYTE   0       ;LOAD ADDRESS
2534                                  ;
2535                                  ;IMAGE DATA
2536                                  ;
2537  002662  005037  000006                 CLR     @#6
2538  002666  012706  001000                 MOV     #1000,SP
2539  002672  012701  177560                 MOV     #177560,R1      ;SET UP TTY
2540  002676  010700                         MOV     PC,R0           ;MAKE ADDR.PIC
2541  002700  062700  000034                 ADD     #<MSG-.>,R0     ;ADDRESS MSG.
2542  002704  105761  000004         1$:     TSTB    4(R1)           ;TTY READY?
2543  002710  100375                         BPL     1$              ;WAIT TIL YES
2544  002712  112061  000006                 MOVB    (R0)+,6(R1)     ;TYPE A CHAR
2545  002716  001372                         BNE     1$              ;KEEP GOING
2546  002720  012737  000026  000024         MOV     #26,@#24        ;SET UP POWER FAIL
2547  002726  005037  000026                 CLR     @#26            ;MAKE SURE T BIT CLAER
2548  002732  000777                         BR      .               ;JUMP ON YOURSELF
2549  002734  006412  047502  052117 MSG:    .ASCII  <12><15>/BOOT MESSAGE WAS RECEIVED SUCCESSFULLY -END OF TEST!!/
2550  002742  046440  051505  040523
2551  002750  042507  053440  051501
2552  002756  051040  041505  044505
2553  002764  042526  020104  052523
2554  002772  041503  051505  043123
2555  003000  046125  054514  026440
2556  003006  047105  020104  043117
2557  003014  052040  051505  020524
2558  003022     041
2559  003023     012  027015  027056             .ASCIZ  <12><15>/....RELOAD PROGRAM..../
2560  003030  051056  046105  040517
2561  003036  020104  051120  043517
2562  003044  040522  027115  027056
2563  003052  000056
2564  003054     006                          .BYTE   6               ;NEXT FOUR BYTES CONTAINS TRANSFER ADDRESS
2565  003055     000                          .BYTE   0               ;;OF PROGRAM JUST DOWNLINE LOADED.
2566  003056     000                          .BYTE   0               ;;;THIS PROGRAM STARTS AT ADDRESS 6.
2567  003057     000                          .BYTE   0
2568  003060                          DLLM2E:                         ;END MEMORY LOAD MESSAGE FORMAT
2569
2570                                          .EVEN
2571
```

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

003060  000122  CMDBUF: .BLKB   82.              ;BUFFER FOR OPERATOR COMMANDS
003202  000000  KEYWD1: .WORD   0                 ;THIS LOC WILL =1 IF CLEAR TYPED, 2 FOR SHOW,
                                                 ; A 4 IF RUN WAS TYPED, 5 IF HELP WAS TYPED
003204  000000  QUALFG: .WORD   0                 ;THIS LOC HOLDS QUALIFIER VALUE (SIZE OR COPY)
003206  000000  QUALVL: .WORD   0
003210  013236  HLPTAB: .WORD   HLP1
003212  013251          .WORD   HLP2
003214  013366          .WORD   HLP3
003216  013453          .WORD   HLP3A
003220  013500          .WORD   HLP4
003222  013557          .WORD   HLP4A
003224  013635          .WORD   HLP5
003226  013725          .WORD   HLP6
003230          HLPEND:
                        ;INDEX TABLE FOR REPORT 'RPT>' HELP MESSAGES  REV B BY EC
003230  014063  RHLPTB: .WORD   RHLP1
003232  014106          .WORD   RHLP2
003234  014141          .WORD   RHLP3
003236  014172          .WORD   RHLP4
003240  014224          .WORD   RHLP5
003242  014263          .WORD   RHLP6
003244  014322          .WORD   RHLP7
003246  000000  RHLPEN: .WORD   0                 ;END OF REPORT HELP TABLE
                        ;INDEX TABLE FOR DMR BASE TABLE DATA DESCRIPTION MESSAGES REV B BY EC
003250  020342  DMRIND: .WORD   DMR000
003252  020402          .WORD   DMR001
003254  020432          .WORD   DMR002
003256  020467          .WORD   DMR003
003260  020532          .WORD   DMR004
003262  020566          .WORD   DMR005
003264  020620          .WORD   DMR006
003266  020663          .WORD   DMR007
003270  020717          .WORD   DMR010
003272  020751          .WORD   DMR011
003274  021003          .WORD   DMR012
003276  021035          .WORD   DMR013
003300  021065          .WORD   DMR014
003302  021114          .WORD   DMR015
003304  021146          .WORD   DMR016
003306  021176          .WORD   DMR017
003310  021226          .WORD   DMR020
003312  021255          .WORD   DMR021
003314  021307          .WORD   DMR022
003316  021333          .WORD   DMR023
003320  021365          .WORD   DMR024
003322  021410          .WORD   DMR025
003324  021463          .WORD   DMR026
003326  021513          .WORD   DMR027
003330  021544          .WORD   DMR030
003332  021627          .WORD   DMR031
003334  021664          .WORD   DMR032
003336  021721          .WORD   DMR033
003340  021756          .WORD   DMR034
003342  022042          .WORD   DMR035
```

```
 2628  003344  022107                          .WORD   DMR036
 2629  003346  022154                          .WORD   DMR037
 2630  003350  022212                          .WORD   DMR040
 2631  003352  022245                          .WORD   DMR041
 2632  003354  022303                          .WORD   DMR042
 2633  003356  022351                          .WORD   DMR043
 2634  003360  022405                          .WORD   DMR044
 2635  003362  022441                          .WORD   DMR045
 2636  003364  022472                          .WORD   DMR046
 2637  003366  022540                          .WORD   DMR047
 2638  003370  022602                          .WORD   DMR050
 2639  003372  022630                          .WORD   DMR051
 2640  003374  022664                          .WORD   DMR052
 2641  003376  022711                          .WORD   DMR053
 2642  003400  022744                          .WORD   DMR054
 2643  003402  022766                          .WORD   DMR055
 2644  003404  023041                          .WORD   DMR056
 2645  003406  023114                          .WORD   DMR057
 2646  003410  023136                          .WORD   DMR060
 2647  003412  023170                          .WORD   DMR061
 2648  003414  023222                          .WORD   DMR062
 2649  003416  023272                          .WORD   DMR063
 2650  003420  023342                          .WORD   DMR064
 2651  003422  023376                          .WORD   DMR065
 2652  003424  023432                          .WORD   DMR066
 2653  003426  023475                          .WORD   DMR067
 2654  003430  023540              DMREND:     .WORD   DMR177          ;NO DMR MESSAGES MUST FOLLOW DMREND
 2655
 2656                                          ;INDEX  TABLE FOR DMC BASE TABLE DATA DESCRIPTION MESSAGES REV B BY EC
 2657  003432  020322              DMCIND:     .WORD   DMUNKN
 2658  003434  020322                          .WORD   DMUNKN
 2659  003436  023604                          .WORD   DMC002
 2660  003440  023625                          .WORD   DMC003
 2661  003442  023662                          .WORD   DMC004
 2662  003444  023723                          .WORD   DMC005
 2663  003446  023756                          .WORD   DMC006
 2664  003450  024013                          .WORD   DMC007
 2665  003452  024050                          .WORD   DMC010
 2666  003454  024103                          .WORD   DMC011
 2667  003456  024125                          .WORD   DMC012
 2668  003460  024147                          .WORD   DMC013
 2669  003462  024206              DMCEND:     .WORD   DMC377          ;NO DMC MESSAGES MUST FOLLOW DMCEND
 2670
 2671  003464  014470  014477  014504  SHTYTB: .WORD   SHTYP0,SHTYP1,SHTYP2,SHTYP3,SHTYP4,SHTYP5,SHTYP6,SHTYP7
 2672  003472  014511  014516  014524
 2673  003500  014531  014537
 2674
 2675                                          ; THE LIST OF BYTES BELOW ARE THE FIRST BYTES OF THE PREDEFINED MESSAGES
 2676                                          ;  USED TO "SHOW" THE TRANSMIT AND COMPARE BUFFER CONTENTS.
 2677
 2678  003504     000     377     252  SHTAB:  .BYTE   0,377,252,125,203,177,043
 2679  003507     125     203     177
 2680  003512     043
 2681  003513                          SHTEND:
 2682          003514                          .EVEN
 2683
```

```
 2684  003514  014550                MODES:  .WORD   M00     ;ADDRESSES OF MODE TYPES IN ASCII
 2685  003516  014560                        .WORD   M01
 2686  003520  014571                        .WORD   M02
 2687  003522  014601                        .WORD   M03
 2688  003524  014610                        .WORD   M04
 2689  003526  014625                        .WORD   M05
 2690  003530  014632                        .WORD   M06
 2691
 2692  003532  014641                LOOPS:  .WORD   LP0     ;ADDRESSES OF LOOP TYPES IN ASCII
 2693  003534  014651                        .WORD   LP1
 2694  003536  014662                        .WORD   LP2
 2695  003540  014670                        .WORD   LP3
 2696  003542  014703                        .WORD   LP4
 2697
 2698                                ;COMMAND LINE TRAVERSE LOCATIONS (USED BY "P$TRV")
 2699
 2700  003544  000000                P$BUFA: .WORD   0               ;LOC. TO HOLD ADDR. OF CMD LINE BUFFER
 2701  003546  000000                P$TREE: .WORD   0               ;LOC. TO HOLD ADDR. OF PARSING TREE
 2702  003550  000000                P$ACT:  .WORD   0               ;LOC. TO HOLD ADDR. OF ACTION ROUTINE
 2703  003552  000000                P$CNT:  .WORD   0               ;LOC. TO BE A COUNTER LOCATION
 2704  003554  000000                P$NUM:  .WORD   0               ;LOC. TO HOLD NUMERIC VALUE FROM PARSE
 2705  003556  000000                P$RADX: .WORD   0               ;LOC. TO HOLD RADIX USED(LO) AND +/-(HI BYTE)
 2706  003560     000                P$NNUF: .BYTE   0               ;RETURN =0 IF ENOUGH OF COMMAND FOUND
 2707  003561     000                P$GDBD: .BYTE   0               ;RETURN CODE 0 IF NO ERROR FOUND
 2708
```

```
                                .SBTTL          MESSAGE BUFFERS AND POINTER TABLES

003562  001000                  TXBUF:  .BLKB   BUFLIM  ;TRANSMITTER BUFFERS
004562  001000                  RXBUF:  .BLKB   BUFLIM  ;RECEIVER BUFFERS
005562  001000                  CMPBUF: .BLKB   BUFLIM  ;COMPARISON BUFFERS
006562  000264                  PTRTAB: .BLKW   180.    ;TABLE FOR MESSAGE ADDRS. & BYTE COUNTS
007332                          PTREND:                 ; END OF MSG. PTR. TABLE

007332  000000                  RXPTR:  .WORD   0       ;RECEIVER MESSAGE POINTER
007334  000000                  TXPTR:  .WORD   0       ;TRANMITTER BUFFER POINTER
007336  000000                  CMPPTR: .WORD   0       ;COMPARISON BUFFER POINTER
007340  000000                  CMPTOT: .WORD   0       ;CMP MSG TOTAL
007342  000000                  CTOTCC: .WORD   0       ;COMPARE BUFFER CHAR. COUNT
007344  000000                  CCURAD: .WORD   0       ;CURRENT ADDR OF CMP BUFF TO ADD AT

007346  000000                  DVTXA:  .WORD   0       ;DEVICE TX ADDR
007350  000000                  DVTCC:  .WORD   0       ;DEVICE TX CHAR COUNT
007352  000000                  DVTCT:  .WORD   0       ;DEVICE TX MESSAGE COUNT
007354  000000                  TXMTOT: .WORD   0       ;TX MSG TOTAL
007356  000000                  TTOTCC: .WORD   0       ;TX BUFFER CHAR. COUNT
007360  000000                  TCURAD: .WORD   0       ;CURRENT ADDR. OF TX BUFF TO ADD AT

007362  000000                  DVRXA:  .WORD   0       ;DEVICE RX ADDR
007364  000000                  DVRCC:  .WORD   0       ;DEVICE RX CHAR COUNT
007366  000000                  DVRCT:  .WORD   0       ;DEVICE RX MESSAGE COUNT
007370  000000                  RXMTOT: .WORD   0       ;RX MSG TOTAL

007372  000000                  LNCNT:  .WORD   0       ;NUMBER OF OPERATOR AWAKE MSGS
007374  000000                  NOBUF:  .WORD   0       ;NUMBER OF NO BUFFS
007376  000000                  PSCNT:  .WORD   0       ;PASS COUNTER
007400  000000                  ERRCNT: .WORD   0       ;ERROR COUNTER
007402  000000                  STADD:  .WORD   0       ;START ADDR.
007404  000000                  ENADD:  .WORD   0       ;END ADDR. FOR DUMP
007406  000000                  BYTBIT: .WORD   0       ;BYTE BIT FOR DUMP ROUTINE

                                ;OTHER MESSSAGE RELATED STORAGE LOCATIONS
007410  000000                  MSGTYP: .WORD   0       ;TYPE OF DATA 0=0'S,1=1'S,2=10'S,3=01'S
                                                        ;4=CCITT,5=QUICK FOX,6=ALPHA/NUM,7=OPER
007412  000000                  CURCC:  .WORD   0       ;TX/RX/CMP CHAR COUNT
007414  000000                  CPTRR:  .WORD   0       ;CURRENT RX POINTER
007416  000000                  CPTR:   .WORD   0       ;CURRENT POINTER
007420  000000                  CURADD: .WORD   0       ;CURRENT TX/RX/CMP START ADDD
007422  000000                  TOTCC:  .WORD   0       ;TOTAL CHAR COUNT NOT MORE THEN 'BUFLIM'
007424  000000                  OFSET:  .WORD   0       ;OFFSET COUNT
007426  000000                  TEMP:   .WORD   0       ;TEMPORARY LOCATIONS (USED A LOT)
007430  000000                  TEMP1:  .WORD   0
007432  000000                  TEMP2:  .WORD   0
007434  000000                  TEMP3:  .WORD   0
007436  000000                  TEMP4:  .WORD   0
007440  000000                  TEMP5:  .WORD   0
007442  000000                  CONOTM: .WORD   0       ;CONTROL OUT ERROR MSG. ADDRESS
007444  000000                  CONTIN: .WORD   0       ;WORD FOR CONTORL IN
007446     000                  GOOD:   .BYTE   0       ;BYTE TO HOLD EXPECTED MESSAGE DATA BYTE FOR ERR REPORT
007447     000                  BAD:    .BYTE   0       ;BYTE TO HOLD RECEIVED MESSAGE DATA BYTE FOR ERR REPORT
007450  000000                  INDEX:  .WORD   0       ;WILL CONTAIN POINTER TO DMC OR DMR MESSAGES
007452  000000                  INDEXE: .WORD   0       ;WILL CONTAIN POINTER TO LAST OF DMC OR DMR MESSAGES
```

```
2765  007454  000000                    BEND:   .WORD   0       ;LAST LOCATION IN BASE TABLE TO BE PRINTED
2766
2767
2768
2769                                    ;MORE INDEPENDENT CODE STORAGE LOCATIONS
2770  007456  000000                    BDATA:  .WORD   0       ;POINTER TO BASE TABLE
2771  007460  000000                    LOGUNT: .WORD   0       ;LOC. TO HOLD LOGICAL UNIT NUMBER
2772  007462  000000                    PCADD:  .WORD   0       ;LOC. HOLD PC OF CALLIN ROUTINE
2773  007464  000000                    DCLFLG: .WORD   0       ;CLEANUP AND EXIT FLAG. 1=DO CLEANUP ROUTINE&EXIT
2774  007466  000000                    RESFLG: .WORD   0       ;LOC TO HOLD FLAG (-1) THAT A RESTART WAS GIVEN
2775  007470  000000                    MODTYP: .WORD   0       ;DCLT MODE OF OPERATION TYPE
2776                                                            ;  (0=REC-ONLY, 1=TX-ONLY, 2=PASSIVE-LOOPBK,
2777                                                            ;   3=ACTIVE-LOOPBK, 4=DOWN L.L., 5=TALK, 6=LISTEN)
2778  007472  000000                    MLTYP:  .WORD   0       ;MAINTENANCE LOOP TYPE (0=NONE, 1=INTERNAL TTL,
2779                                                            ;  2=CABLE, 3=MODEM-ANALOG LOOPBK (LOCAL),
2780                                                            ;  4=MODEM-DIGITAL LOOPBK (REMOTE), 5=MOP)
2781  007474  000000                    FHDPLX: .WORD   0       ;FULL OR HALF DUPLEX FLAG (1=FULL FROM P-TABLE)
2782  007476  000002                    PARAM:  .WORD   2       ;PROGRAM PARAMETERS
2783                                                            ; BIT0= STATUS MSGS TO OPR PRINTED (1=YES)
2784                                                            ; BIT1= DATA CHECKING DONE ON RCVD MSGS (1=YES)
2785                                                            ; BIT2= ECHO (TRANSMIT) RCV'D MSG.(PASSIVE)(1=YES)
2786                                                            ; BIT3= SPARE
2787                                                            ; BIT4= CRC CALC./CHECK DONE (1=YES)
2788                                                            ; BIT5= PROTOCOL EMULATION (1=YES)
2789                                                            ; BIT6= SPARE
2790  007500  000000                    RPASS:  .WORD   0       ;PASS NUMBER FROM RUN COMMAND
2791  007502  000000                    FLAG:   .WORD   0       ;DEVICE FLAG WORD
2792
2793                                    ;MODE DISPATCH TABLE
2794  007504  040636                    MODE:   .WORD   RXONLY  ;RX ONLY DISPATCH
2795  007506  040670                            .WORD   TXONLY  ;TX ONLY DISPATCH
2796  007510  040730                            .WORD   PLCK    ;PASSIVE LOOP BACK DISP
2797  007512  040764                            .WORD   ALCK    ;ACTIVE LOOP BACK DISP
2798  007514  042114                            .WORD   DLL     ;DOWN LINE LOAD DISP
2799  007516  042734                            .WORD   TALCK   ;TALK MODE DISPATCH
2800  007520  043154                            .WORD   LISCK   ;LISTEN MODE DISPATCH
2801
2802
2803                                    .SBTTL          CLOCK TABLES, EVENT LOG AND POINTERS
2804  007522  000000                    CLKCSR: .WORD   0       ;CLOCK CSR ADDRESS
2805  007524  000000                    CLKBR:  .WORD   0       ;CLOCK INTERRUPT LEVEL
2806  007526  000000                    CLKVEC: .WORD   0       ;CLOCK INTERRUPT VECTOR
2807  007530  000074                    CLKHZ:  .WORD   60.     ;CLOCK'S HERTZ RATE
2808  007532  000000                    CLKEN:  .WORD   0       ;CLOCK'S CSR VALUE TO INTRPT. ENABLE IT
2809
2810  007534  000000                    TIMMIN: .WORD   0       ;PLACE TO KEEP TIME-SINCE-START
2811  007536  000000                    TIMSEC: .WORD   0
2812  007540  000000                    TIMTCK: .WORD   0       ;PLACE TO KEEP # OF TICKS/SEC
2813
2814  007542  000000                    TIMER1: .WORD   0       ;EVENT TIMER #1 (TICKS)
2815  007544  000000                    TIMER2: .WORD   0       ;EVENT TIMER #2 (TICKS)
2816  007546  000000                    TIMER3: .WORD   0       ;EVENT TIMER #3 (SECONDS)
2817
```

```
                                ;EVENT LOG TABLE AND ITS NEXT ENTRY POINTER
007550  007552                  EVTPTR: .WORD   EVTLOG  ;POINTER TO NEXT FREE SPACE IN EVENT LOG
007552  000341                  EVTLOG: .BLKW   225.    ;EVENT LOG BUFFER
010454  000001                  EVTEND: .BLKW   1.      ;APPROXIMATE END OF EVENT TABLE (ALLOWS CIRCULAR QUE)

                                .SBTTL          MODEM DATA SECTION

010456  000000                  MODS:   .WORD   0       ;MODEM STATUS

                                ;TABLE OF MODEM SIGNAL BIT DEFINITIONS

010460  000004                  MOBITS: .WORD   CTS             ;CLEAR TO SEND (CIRCUIT CB)
010462  000010                          .WORD   DSR             ;DATA SET READY (CIRCUIT CC)
010464  000001                          .WORD   DCD             ;DATA CARRIER DETECT (CIRCUIT CF)
010466  000040                          .WORD   RTS             ;REQUEST TO SEND (CIRCUIT CA)
010470  000200                          .WORD   RI              ;RING INDICATOR (CIRCUIT CE)
010472  040000                          .WORD   SQD             ;SIGNAL QUALITY DETECT (CIRCUIT CG)
010474  001000                          .WORD   TM              ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)
010476                          MOBITE:

                                ;TABLE OF ADDRESSES OF MODEM SIGNAL MESSAGE POSITIONS

010476  017255                  MOMSGS: .WORD   EVMCTS          ;CLEAR TO SEND (CIRCUIT CB)
010500  017261                          .WORD   EVMDSR          ;DATA SET READY (CIRCUIT CC)
010502  017265                          .WORD   EVMDCD          ;DATA CARRIER DETECT (CIRCUIT CF)
010504  017271                          .WORD   EVMRTS          ;REQUEST TO SEND (CIRCUIT CA)
010506  017275                          .WORD   EVMRI           ;RING INDICATOR (CIRCUIT CE)
010510  017301                          .WORD   EVMSQD          ;SIGNAL QUALITY DETECT (CIRCUIT CG)
010512  017305                          .WORD   EVMTM           ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)

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

010514  015631                  EVTLST: .WORD   EDTXQ    ;TRANSMIT MESSAGE QUEUED
010516  015655                          .WORD   EDTXC    ;TRANSMIT OF MESSAGE COMPLETE
010520  015704                          .WORD   EDRXQ    ;RECEIVE MESSAGE SPACE QUEUED
010522  015731                          .WORD   EDRXC    ;MESSAGE RECEIVED - RECEIVE COMPLETE
010524  015757                          .WORD   EDDER    ;DEVICE INFORMATION
010526  016024                          .WORD   EDDVI    ;DEVICE INITIALIZE STARTED
010530  015774                          .WORD   EDDCK    ;DATA COMPARISON DONE
010532  014641                          .WORD   LPO      ;NULL STRING
010534  016052                          .WORD   EDDLE    ;DATA COMPARE LENGTH ERROR
010536  016107                          .WORD   EDDDE    ;DATA COMPARE DATA ERROR
010540  016142                          .WORD   EDEOP    ;END OF PASS
010542  016213                          .WORD   EDABO    ;^ C ABORT

                                ;;;FOLLOWING TABLE USED IN DOWNLINE LOAD ROUTINE.
                                ;;;;CONTAINS POINTERS TO ASCIZ DEVICE DESCRIPTIONS
010544  020217                  DLLIND: .WORD   DPM
010546  020222                          .WORD   DUM
010550  020225                          .WORD   DLM
010552  020230                          .WORD   DQM
010554  020233                          .WORD   DAM
010556  020236                          .WORD   DUPM
010560  020242                          .WORD   DMCM
010562  020246                          .WORD   DNM
```

```
2874  010564  020251                    .WORD   DLVM
2875  010566  020255                    .WORD   DMPM
2876  010570  020261                    .WORD   DTEM
2877  010572  020265                    .WORD   DVM
2878  010574  020270                    .WORD   DZM
2879  010576  020273                    .WORD   UNKM
2880  010600  020303                    .WORD   KDPM
2881  010602  020307                    .WORD   KDZM
2882  010604  020313                    .WORD   KLM
2883  010606  020316                    .WORD   DMVM
2884
2885                              ;LOCATIONS USED DURING EVENT REPORTING
2886
2887  010610  000000              EVTSEC: .WORD   0       ;TEMPORARY LOCS TO KEEP EVENT TIME WHILE REPORTING
2888  010612  000000              EVTMIN: .WORD   0
2889  010614  000000              EVTTCK: .WORD   0
2890  010616  000000              EVTADD: .WORD   0       ;TEMP. LOC. TO HOLD ADDRESS DURING EVENT REPORTING
2891  010620  000000              EVTBCT: .WORD   0       ;       ''      ''  BYTE COUNT   ''      ''      ''
2892  010622  000000              EVTTMP: .WORD   0       ;       ''      ''  OTHER DATA   ''      ''      ''
2893
2894                              ;REPORT CODING DISPATCH TABLE
2895
2896  010624  031014              RPTDSP: .WORD   RPTTXQ  ;TRANSMIT QUEUED ENTRY DECODING
2897  010626  031014                      .WORD   RPTTXQ  ;TRANSMIT COMPLETE ENTRY DECODING
2898  010630  031014                      .WORD   RPTTXQ  ;RECEIVER QUEUED ENTRY DECODING
2899  010632  031014                      .WORD   RPTTXQ  ;RECEIVER COMPLETE ENTRY DECODING
2900  010634  031066                      .WORD   RPTDER  ;DEVICE ERROR ENTRY DECODING
2901  010636  031162                      .WORD   RPTDVI  ;DEVICE INIT ENTRY DECODING
2902  010640  031356                      .WORD   RPTDCK  ;DATA COMPARISON ENTRY DECODING
2903  010642  030642                      .WORD   RPT     ;PLACE HOLDER
2904  010644  031356                      .WORD   RPTDLE  ;DATA COMPARISON LENGH ERROR
2905  010646  031302                      .WORD   RPTDDE  ;DATA COMPARISON DATA ERROR
2906  010650  031226                      .WORD   RPTEOP  ;END OF PASS
2907  010652  031226                      .WORD   RPTABO  ;^C ABORT
2908
2909
2910  010654  000000              DEV1:   .WORD   0               ;TEMP LOCS TO HOLD DATA FOR EVENT REPORTING
2911  010656  000000              DEV2:   .WORD   0               ;  AND SHOW MODE,... SUBROUTINE
2912  010660  000000              DEV3:   .WORD   0
2913  010662  000000              DEV4:   .WORD   0
2914
```

```
2915                                    .SBTTL          COMMAND LINE ACTION TREE
2916
2917                                    ;SAMPLE CLI TREE NODE   (ALWAYS AT LEAST 1 WORD)
2918                                    ; ----------------------
2919                                    ; ! ACTION  ! CHAR CODE !
2920                                    ; ----------------------
2921                                    ; !  MISS DISPLACEMENT  !       ONLY IF 'MISS'' ARGUMENT DEFINED
2922                                    ; ----------------------
2923                                    ; ! NEXT NODE DISPLMNT  !       ONLY IF ''ASCII'' ARGUMENT DEFINED
2924                                    ; ----------------------
2925                                    ; ! ASCIZ MATCH STRING  !       ONLY IF ''ASCII'' ARGUMENT DEFINED
2926                                    ; !             (.EVEN) !
2927                                    ; ----------------------
2928
2929
2930  010664                            CLITRE:
2931
2932                                    ;FIRST KEYWORD
2933  010664                                    CLI     CLISPA,0,N10$                   ;SKIP ANY LEADIN SPACES
2934  010670                            N10$:   CLI     <'?>,HLP,N42$                   ;IS THE FIRST NON-SP CHAR A ''?''
2935  010674                                    CLI     CLIEXI,0                        ;  IF YES DO ''HLP'' AND EXIT
2936  010676                            N42$:   CLI     CLISTR,HLP,N43$,<'HELP'>        ;ELSE, IS FIRST WORD A ''HELP''
2937  010712                                    CLI     CLIEXI,0                        ;  IF YES DO ''HLP'' AND EXIT
2938  010714                            N43$:   CLI     CLISTR,PRNT,N44$,<'PRINT'>      ;ELSE, IS FIRST WORD A ''PRINT''
2939  010730                                    CLI     CLIEXI,0                        ;  IF YES DO ''PRINT'' AND EXIT
2940  010732                            N44$:   CLI     CLISTR,EXIT,N45$,<'EXIT'>       ;ELSE,IS FIRST WORD AN ''EXIT'';REV B BY E
2941  010746                                    CLI     CLIEXI,0                        ;  IF YES DO ''EXIT'' AND EXIT
2942  010750                            N45$:   CLI     CLISTR,RUN,N46$,<'RUN'>         ;ELSE, IS FIRST WORD A ''RUN''
2943  010762                                    CLI     CLIBR,0,N80$                    ;  IF YES DO ''RUN'' & GOTO N80$
2944  010766                            N46$:   CLI     CLISTR,NOTNUF,N40$,<'DUMP'>     ;ELSE, IS FIRST WORD A ''DUMP''
2945  011002                                    CLI     CLIBR,0,N50$                    ;  IF YES GOTO N80$
2946  011006                            N40$:   CLI     CLISTR,CLEAR,N20$,<'CLEAR'>     ;ELSE, IS FIRST WORD A ''CLEAR''
2947  011022                                    CLI     CLIBR,NOTNUF,N100$              ;  IF YES DO ''CLR'' & GOTO N100$
2948  011026                            N20$:   CLI     <'S>,NOTNUF,N30$                ;ELSE, IS FIRST CHAR. A ''S''
2949  011032                                    CLI     CLISTR,SHOW,N25$,<'HOW'>        ;  IF YES IS REST OF WORD ''HOW''
2950  011044                                    CLI     CLIBR,0,N100$                   ;     IF YES, DO ''SHOW'',BR N100$
2951  011050                            N25$:   CLI     CLISTR,0,N30$,<'ET'>            ;  ELSE, IS REST OF WORD ''ET''
2952  011062                                    CLI     CLIBR,0,N110$                   ;     IF YES, DO ''SET'', BR N110$
2953  011066                            N30$:   CLI     CLIERR,0                        ;OTHERWISE ''ILL CMD'' - EXIT
2954
2955                                    ;SECOND KEYWORD (MODE=) FOR RUN COMMAND
2956
2957  011070                            N80$:   CLI     CLISPA,0,N30$                   ;SKIP LEADING SPS, IF NONE-ERR
2958  011074                            N81$:   CLI     CLISTR,NOTNUF,N30$,<'MODE'>     ;IS NEXT WORD ''MODE=''
2959  011110                                    CLI     <'=>,0,N30$                     ;  IF NO, IT'S WRONG -ERR -EXIT
2960  011114                                    CLI     CLISTR,ATVMOD,N82$,<'ACTIVE'>   ;IS NEXT WORD ''ACTIVE''
2961  011132                                    CLI     CLIBR,0,N115$                   ;  IF YES, DO ''ACTIVE'',BR N115$
2962  011136                            N82$:   CLI     CLISTR,PASMOD,N83$,<'PASSIVE'>  ;IS NEXT WORD ''PASSIVE''
2963  011154                                    CLI     CLIBR,0,N115$                   ;  IF YES, DO ''PASSVE'',BR N115$
2964  011160                            N83$:   CLI     CLISTR,RECMOD,N84$,<'RECEIVE'>  ;IS NEXT WORD ''RECEIVE''
2965  011176                                    CLI     CLIBR,0,N115$                   ;  IF YES, DO ''RECVE'',BR N115$
2966  011202                            N84$:   CLI     CLISTR,LISMOD,N85$,<'LISTEN'>   ;IS NEXT WORD ''LISTEN''
2967  011220                                    CLI     CLIBR,0,N115$                   ;  IF YES, DO ''LISTEN'',BR N115$
2968  011224                            N85$:   CLI     CLISTR,DLLMOD,N86$,<'DOWNLINELOAD'>      ;IS NEXT WORD 'DOW...''
2969  011250                                    CLI     CLIBR,0,N115$                   ;  IF YES, DO ''DWNLL'',BR N115$
2970  011254                            N86$:   CLI     <'T>,0,N30$                     ;IS NEXT CHAR A ''T''
```

```
 2971  011260                          CLI     CLISTR,TRAMOD,N87$,<'RANSMIT'>  ; IS REST OF WORD 'RANSMIT'
 2972  011276                          CLI     CLIBR,0,N115$                   ;  IF YES, DO "TRANSM",BR N115$
 2973  011302                  N87$:   CLI     CLISTR,TALMOD,N30$,<'ALK'>      ; IS REST OF WORD "ALK"
 2974  011314                          CLI     CLIBR,0,N115$                   ;  IF YES, DO "TALK",BR N115$
 2975                                                                          ; IF NO, ERROR - EXIT
 2976
 2977                                  ;SECOND KEYWORD (FOR CLEAR OR SHOW)
 2978  011320                  N100$:  CLI     CLISPA,0,N30$                   ;SKIP LEADING SPACES, NONE=ERR
 2979  011324                  N102$:  CLI     CLISTR,CSHEXP,N104$,<'EXPECT'>  ;IS NEXT WORD "EXPE..."
 2980  011342                          CLI     CLIEXI,0                        ; IF YES, DO CLR-EXP,EXIT
 2981  011344                  N104$:  CLI     CLISTR,CSHTRN,N30$,<'TRANSMIT'> ;IS NEXT WORD "TRANS.."
 2982  011364                          CLI     CLIEXI,0                        ; IF YES, DO CLR-TRN,EXIT
 2983                                                                          ;IF NO - ERROR - EXIT
 2984
 2985
 2986                                  ;SECOND KEYWORD (FOR SET)
 2987  011366                  N110$:  CLI     CLISPA,0,N30$
 2988  011372                  N111$:  CLI     CLISTR,SETEXP,N112$,<'EXPECT'>
 2989  011410                          CLI     CLIBR,0,N120$
 2990  011414                  N112$:  CLI     CLISTR,SETTRN,N30$,<'TRANSMIT'>
 2991  011434                          CLI     CLIBR,0,N120$
 2992
 2993                                  ;GET ADDRESSES FOR DUMP COMMAND
 2994  011440                  N50$:   CLI     CLIALP,0,N51$
 2995  011444                  N51$:   CLI     CLISPA,0,N52$
 2996  011450                  N52$:   CLI     CLIOCT,DMPS,N30$
 2997  011454                          CLI     <'->,NOTNUF,N125$
 2998  011460                          CLI     CLIOCT,DMPE,N30$
 2999  011464                          CLI     <'/>,NOTNUF,N125$
 3000  011470                          CLI     <'B>,DMPQ,N30$
 3001  011474                          CLI     CLIBR,0,N125$
 3002
 3003                                  ;QUALIFIERS FOR THE RUN COMMAND
 3004  011500                  N115$:  CLI     CLIALP,0,N114$
 3005  011504                  N114$:  CLI     <'/>,NOTNUF,N125$
 3006  011510                          CLI     CLISTR,NO,N116$,<'NO'>
 3007  011522                  N116$:  CLI     <'C>,0,N117$
 3008  011526                          CLI     CLISTR,CHECK,N117$,<'HECK'>
 3009  011542                          CLI     CLIBR,0,N115$
 3010
 3011
 3012                                  ;N113$: CLI     CLISTR,CRC,N30$,<'RC16'>
 3013                                  ;       CLI     CLIBR,0,N115$
 3014
 3015  011546                  N117$:  CLI     CLISTR,STATUS,N118$,<'STATUS'>
 3016  011564                          CLI     CLIBR,0,N115$
 3017  011570                  N118$:  CLI     CLISTR,ECHO,N130$,<'ECHO'>
 3018  011604                          CLI     CLIBR,0,N115$
 3019
 3020
 3021  011610                  N130$:  CLI     CLISTR,0,N131$,<'PASS'>
 3022  011624                          CLI     CLIBR,0,N150$
 3023  011630                  N131$:  CLI     CLISTR,0,N132$,<'LOOP'>
 3024  011644                          CLI     CLIBR,0,N140$
 3025
 3026  011650                  N132$:  CLI     CLISTR,MOSC,N30$,<'MODEM'>      ;MODEM ACTION
```

```
3027  011664                        CLI     CLIBR,0,N115$                   ;;ADDED BY EC
3028
3029                              ;GET MESSAGE TYPE FOR SET MESSAGE COMMANDS
3030  011670                    N120$:  CLI     <'=>,0,N30$
3031
3032                              ;   LOOK FOR DEFAULT MESSAGE NAME
3033  011674                    N60$:   CLI     CLISTR,CMSG1,N61$,<'ONES'>
3034  011710                            CLI     CLIBR,0,N121$
3035  011714                    N61$:   CLI     CLISTR,CMSG0,N62$,<'ZEROES'>
3036  011732                            CLI     CLIBR,0,N121$
3037  011736                    N62$:   CLI     CLISTR,CMSG2,N63$,<'1ALT'>
3038  011752                            CLI     CLIBR,0,N121$
3039  011756                    N63$:   CLI     CLISTR,CMSG3,N64$,<'0ALT'>
3040  011772                            CLI     CLIBR,0,N121$
3041  011776                    N64$:   CLI     CLISTR,CMSG5,N65$,<'ITEP'>
3042  012012                            CLI     CLIBR,0,N121$
3043  012016                    N65$:   CLI     CLISTR,CMSG4,N66$,<'CCITT'>
3044  012032                            CLI     CLIBR,0,N121$
3045  012036                    N66$:   CLI     CLISTR,CMSG6,N67$,<'ALPHA'>
3046  012052                            CLI     CLIBR,0,N121$
3047  012056                    N67$:   CLI     CLISTR,SETET,N68$,<'TRANSMIT'>  ;REV B BY EC
3048  012076                            CLI     CLIBR,0,N125$                   ;REV B BY EC
3049
3050                              ;   LOOK FOR QUOTED MESSAGE
3051  012102                    N68$:   CLI     <''>,OPRMSG,N30$
3052  012106                    N70$:   CLI     <''>,ENDQO,N71$
3053  012112                            CLI     CLIBR,0,N121$
3054  012116                    N71$:   CLI     CLISPA,0,N72$
3055  012122                    N72$:   CLI     CLIALN,0,N73$           ;ONLY A-Z,SP,TAB, OR 0-9 BETWEEN '"S
3056  012126                            CLI     CLIBR,0,N70$
3057  012132                    N73$:   CLI     CLIERR,BADCHR           ;PRINT ERROR IF NONE LEGAL CHAR FOR '"S
3058
3059                              ;GET QUALIFIERS (SIZE OR COPY) FOR SET MESSAGE COMMANDS
3060  012134                    N121$:  CLI     CLIALP,0,N123$
3061  012140                    N123$:  CLI     <'/>,NOTNUF,N125$
3062  012144                            CLI     CLISTR,SIZE,N122$,<'SIZE'>
3063  012160                            CLI     CLIBR,0,N126$
3064  012164                    N122$:  CLI     CLISTR,QCOPY,N30$,<'COPY'>
3065  012200                            CLI     CLIBR,0,N126$
3066
3067                              ;NUMER FOR SIZE OR COPY
3068  012204                    N126$:  CLI     <'=>,0,N30$
3069  012210                            CLI     CLIDEC,NUM,N30$
3070  012214                            CLI     CLIBR,0,N121$
3071
3072                              ;GET MAINTENANCE LOOP TYPE FOR RUN ''LOOP'' QUIALIFIER
3073  012220                    N140$:  CLI     <'=>,0,N30$
3074
3075
3076  012224                    N141$:  CLI     CLISTR,TTLLOP,N142$,<'INTERNALTTL'>
3077  012246                            CLI     CLIBR,0,N115$
3078  012252                    N142$:  CLI     CLISTR,CBLLOP,N143$,<'CABLE'>
3079  012266                            CLI     CLIBR,0,N115$
3080  012272                    N143$:  CLI     CLISTR,LMDLOP,N144$,<'LOCALMODEM'>
3081  012314                            CLI     CLIBR,0,N115$
3082  012320                    N144$:  CLI     CLISTR,RMDLOP,N30$,<'REMOTEMODEM'>
```

```
 3083  012342                          CLI     CLIBR,0,N115$
 3084
 3085                                ;GET LINE NUMBER FOR "PASS" RUN QUALIFIER
 3086  012346                        N150$:  CLI     <'=>,0,N30$
 3087  012352                                CLI     CLIDEC,PASC,N30$
 3088  012356                                CLI     CLIBR,0,N115$
 3089
 3090
 3091
 3092                                ;END-OF-LINE
 3093  012362                        N125$:  CLI     CLIEXI,0
 3094
```

```
3095
3096
3097                                    ;DEVICE DEPENDENT STORAGE LOCATIONS FOR
3098                                    ;  CURRENT DEVICE PARAMTERS
3099
3100   012364                           SEL0:
3101   012364   000000                 BSEL0:  .WORD   0               ;ADDRESSES OF REGISTERS SEL0 THRU BSEL7
3102   012366   000000                 BSEL1:  .WORD   0
3103   012370                           SEL2:
3104   012370   000000                 BSEL2:  .WORD   0
3105   012372   000000                 BSEL3:  .WORD   0
3106   012374                           SEL4:
3107   012374   000000                 BSEL4:  .WORD   0
3108   012376   000000                 BSEL5:  .WORD   0
3109   012400                           SEL6:
3110   012400   000000                 BSEL6:  .WORD   0
3111   012402   000000                 BSEL7:  .WORD   0
3112
3113
3114   012404   000000                 INVEC:  .WORD   0               ;INPUT INTERRUPT VECTOR ADDRESS
3115   012406   000000                 OUTVEC: .WORD   0               ;OUTPUT INTERRUPT VECTOR ADDRESS
3116   012410   000000                 INTPRI: .WORD   0               ;INTERRUPT PRIORITY
3117   012412   000000                 OPTYP:  .WORD   0               ;DEVICE OPTION TYPE(0=DMC,5=DMR-DMC MODE
                                                                       ;7=DMR).
3118
3119
3120
3121
3122
3123                                    ;       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
                                ;
012414                                  DEVTYP  <DMR,DMC-11>
012414                                                                  L$DVTYP::
012414  046504  026122  046504                                                .ASCIZ  /DMR,DMC-11/
012422  026503  030461     000
        012430                                                          .EVEN


                                .SBTTL          PROGRAM IDENTIFICATION
                                ; TEST DESCRIPTION
                                ;
012430                                  DESCRIPT        <CZCLKC0 DMR, DMC-11 DATA COMM. LINK TEST>
012430                                                                  L$DESC::
012430  055103  046103  041513                                                .ASCIZ  /CZCLKC0 DMR, DM
012436  020060  046504  026122
012444  042040  041515  030455
012452  020061  040504  040524
012460  041440  046517  027115
012466  046040  047111  020113
012474  042524  052123     000
        012502                                                          .EVEN
                                .EVEN
```

```
3162                                    .SBTTL          GLOBAL FORMAT STATEMENTS, MESSAGES, AND ASCII INFO
3163
012502 041504 052114 000076  CLI$PM: .ASCIZ  /DCLT>/
012510 050122 037124 000040  CLI$RP: .ASCIZ  /RPT> /                    ;REV B BY EC
012516 047045 040445 044477  CLIERM: .ASCIZ  /%N%A?ILL CMD-BAD SYNTX?/
012546 047045 040445 044477  CLINUF: .ASCIZ  /%N%A?INCMPLTE CMD?/
012571    045 022516 037501  CLINBG: .ASCIZ  /%N%A?NUM TOO BIG?/
012613    045 022516 037501  CLIBRX: .ASCIZ  /%N%A?BAD RADIX?/
012633    045 022516 037501  CLIBDL: .ASCIZ  /%N%A?''LOOP'' VALID ONLY IN ACTIVE?/
012675    045 022516 037501  CLINPS: .ASCIZ  /%N%A?''ECHO'' VALID ONLY IN PASSIVE?/
012740 047045 040445 044477  CLIBCR: .ASCIZ  /%N%A?ILL CHR- ''A-Z,0-9,SP,TAB'' ONLY?/
013005    045 022516 037501  CLISE0: .ASCIZ  /%N%A?''SIZE=0'' NOT VALID?/
013036 047045 040445 052077  CLIPW:  .ASCIZ  /%N%A?TRANSMIT & EXPECT LIST MUST BE IDENTICAL FOR LOOP?/ ;REV B BY EC
013126 040523 052124 046105  DLLQ1:  .ASCIZ  /SATTELITE PASSWORD= /          ;REV B BY EC
013153    045 022516 052101  HLP0:   .ASCIZ  /%N%ATHIS IS DCLT. TYPE ''H'' OR ''?'' FOR DETAILS/
013231    045 022516 000124  HLPF:   .ASCIZ  /%N%T/
013236 041504 052114 041440  HLP1:   .ASCIZ  /DCLT CMDS:/
013251    040 046103 040505  HLP2:   .ASCII  / CLEAR OR SHOW  EXPECTLIST OR TRANSMITLIST/<15><12>
013325    040 051120 047111          .ASCII  / PRINT/<15><12>
013335    040 054105 052111          .ASCII  / EXIT/<15><12>  ;REV B BY EC
013344 042040 046525 020120          .ASCIZ  ? DUMP START-END/B?
013366 051440 052105 042440  HLP3:   .ASCIZ  ? SET EXPECTMSG OR TRANSMITMSG=TYPE/SIZE=N OR /COPY=N?
013453    040 042523 020124  HLP3A:  .ASCIZ  / SET EXPECT=TRANSMIT/  ;REV B BY EC
013500 020040 052040 050131  HLP4:   .ASCIZ  ?   TYPE=ONES,ZEROES,1ALT,0ALT,ITEP,CCITT,ALPHA?
013557    040 020040 020040  HLP4A:  .ASCIZ  /       OR ''OPR SPCD=A-Z,SP,TAB,0-9 IN QUOTES''/
013635    040 052522 020116  HLP5:   .ASCIZ  ? RUN MODE=MTYP/LOOP=LTYP/CHECK,STATUS,ECHO,MODEM,PASS=N?
013725    040 020040 052115  HLP6:   .ASCII  /   MTYP=TRAN,REC,ACT,PAS,TAL,LIS,DOWN/<15><12>
013774 020040 046040 054524          .ASCIZ  /   LTYP=INT,CAB,LOC,REM/
014024 047045 040445 054524  RHLP0:  .ASCIZ  /%N%ATYPE ''H'' OR ''?'' FOR HELP !/          ;REV B BY EC
014063    104 046103 020124  RHLP1:  .ASCIZ  /DCLT REPORT CMDS :/              ;REV B BY EC
014106 047514 020107 020055  RHLP2:  .ASCIZ  /LOG - PRINT DCLT EVENT LOG/                  ;REV B BY EC
014141    105 044530 020124  RHLP3:  .ASCIZ  /EXIT - EXIT REPORT LEVEL/        ;REV B BY EC
014172 042510 050114 026440  RHLP4:  .ASCIZ  /HELP - PRINT THIS MESSAGE/      ;REV B BY EC
014224 040502 042523 042457  RHLP5:  .ASCIZ  !BASE/ERROR - PRINT ONLY ERRORS!          ;REV B BY EC
014263    102 051501 027505  RHLP6:  .ASCIZ  !BASE/FULL - PRINT ENTIRE TABLE!          ;REV B BY EC
014322 040502 042523 047457  RHLP7:  .ASCIZ  !BASE/OFFSET=NNN - PRINT SINGLE LOCATION!<15><12> ;REV B BY EC
014374 047045 040445 040502  RPTIV:  .ASCIZ  /%N%ABASE OFFSET=%O3%A TOO BIG !/          ;REV B BY EC
014434 047045 040445 051515  SHMSG:  .ASCIZ  ?%N%AMSG: TYPE=%T%A/SIZE=%D3?
014470 042532 047522 051505  SHTYP0: .ASCIZ  /ZEROES/
014477    117 042516 000123  SHTYP1: .ASCIZ  /ONES/
014504 040461 052114    000  SHTYP2: .ASCIZ  /1ALT/
014511    060 046101 000124  SHTYP3: .ASCIZ  /0ALT/
014516 041503 052111 000124  SHTYP4: .ASCIZ  /CCITT/
014524 052111 050105    000  SHTYP5: .ASCIZ  /ITEP/
014531    101 050114 040510  SHTYP6: .ASCIZ  /ALPHA/
014537    117 051120 051440  SHTYP7: .ASCIZ  /OPR SPEC/
014550 042522 042503 053111  MO0:    .ASCIZ  /RECEIVE/
014560 051124 047101 046523  MO1:    .ASCIZ  /TRANSMIT/
014571    120 051501 044523  MO2:    .ASCIZ  /PASSIVE/
014601    101 052103 053111  MO3:    .ASCIZ  /ACTIVE/
014610 047504 047127 044514  MO4:    .ASCIZ  /DOWNLINELOAD/
014625    124 046101 000113  MO5:    .ASCIZ  /TALK/
014632 044514 052123 047105  MO6:    .ASCIZ  /LISTEN/
014641    000                LP0:    .ASCIZ  //
014642 046057 047517 036520  LP00:   .ASCIZ  ?/LOOP=?
014651    111 052116 051105  LP1:    .ASCIZ  ?INTERNAL?
```

```
014662  040503  046102  000105  LP2:    .ASCIZ  ?CABLE?
014670  047514  040503  046514  LP3:    .ASCIZ  ?LOCALMODEM?
014703     122  046505  052117  LP4:    .ASCIZ  ?REMOTEMODEM?
014717     116     117          PNST:   .ASCII  /NO/
014721     123  040524  052524  PST:    .ASCIZ  /STATUS/
014730  047516                  PNCK:   .ASCII  /NO/
014732  044103  041505  000113  PCK:    .ASCIZ  /CHECK/
014740  047516                  PNEC:   .ASCII  /NO/
014742  041505  047510     000  PEC:    .ASCIZ  /ECHO/

014747     116     117          PNMS:   .ASCII  /NO/                    ;ADDED BY EC
014751     115  042117  046505  PMS:    .ASCIZ  /MODEM/                 ;ADDED BY EC
014757     045  022516  046101  LISP:   .ASCIZ  /%N%ALIS>/
014770  046124  037113     000  OPRMM:  .ASCIZ  /TLK>/
014775     124  044510  020123  L5060:  .ASCIZ  /THIS A 50. OR 60. HZ. LSI-11:/
        015034                  .EVEN



                                ;
                                ; FORMAT STATEMENTS USED IN PRINT CALLS
                                ;

015034  047045  040445  047504  DLLCM:  .ASCIZ  /%N%ADOWN LINE LOAD COMPLETED SUCCESSFULLY/
015106  047045  040445  040502  NOCLK:  .ASCIZ  /%N%ABAD CLOCK - PROGRAM WILL HANG ON ''TIMEOUT''!!/
015167     115  054101  020056  TABEX:  .ASCIZ  /MAX. CHAR. MSG COUNT EXCEEDED -/
015227     102  043125  042506  BUFEX:  .ASCIZ  /BUFFER FULL -/
015245     045  022516  022524  MSGTRN: .ASCIZ  /%N%T%A MSG. NOT BUILT !!/
015276  047045  040445  044103  MSGTRU: .ASCIZ  /%N%ACHAR. COUNT EXCEEDS BUFF LIMIT - MSG TRUNCATED/
015361     045  022516  032523  SHF0:   .ASCIZ  ?%N%S5%AMODE=%T%T%T%A/PASS=%Z5?

015417     045  022516  032523  SHF1:   .ASCIZ  ?%N%S5%S5%S5%A/%T%A/%T%A/%T%A/%T?

015457     045  032523  040445  EFM2:   .ASCIZ  /%S5%ATOTAL MISMATCHES IN MSG = %D5/
015522  047045  051445  022463  PCPM:   .ASCIZ  /%N%S3%ACALLED FROM PC=%O6/
015554  051445  022465  041501  EFM11:  .ASCIZ  /%S5%ACOMPARE COUNT=%D5%S3%ARECEIVE COUNT=%D5/

                                ;EVENT DESCRIPTION MESSAGES

015631     124  040522  051516  EDTXQ:  .ASCIZ  /TRANSMIT MSG QUEUED/
015655     124  040522  051516  EDTXC:  .ASCIZ  /TRANSMIT MSG COMPLETED/
015704  042522  042503  053111  EDRXQ:  .ASCIZ  /RECEIVE SPACE QUEUED/
015731     122  041505  044505  EDRXC:  .ASCIZ  /RECEIVE MSG COMPLETED/
015757     104  053105  041511  EDDER:  .ASCIZ  /DEVICE ERROR/
015774  040504  040524  041440  EDDCK:  .ASCIZ  /DATA COMPARISON STARTED/
016024  042504  044526  042503  EDDVI:  .ASCIZ  /DEVICE INIT AND SETUP/
016052  040504  040524  041440  EDDLE:  .ASCIZ  /DATA COMPARISON LENGTH ERROR/
016107     104  052101  020101  EDDDE:  .ASCIZ  /DATA COMPARISON DATA ERROR/
016142  047105  020104  043117  EDEOP:  .ASCIZ  /END OF PASS/
016156  041101  047516  046522  EDMOS:  .ASCIZ  /ABNORMAL MODEM STATUS CHANGE/          ;ADDED BY EC
016213     136  020103  041101  EDABO:  .ASCIZ  /^C ABORT/

                                ;*****************************************************************
                                ;THESE TWO STORAGE AREAS MUST NOT BE SEPERATED !!!!
```

```
                                        ;EVENT REPORTING MESSAGES
016224  020040  040502  042523  BASM1A: .ASCIZ  /  BASE TABLE/

016241     045  031523  047445  BASM3:  .ASCIZ  /%S3%03/
016250  051445  022463  033117  BASM2:  .ASCIZ  /%S3%06/
016257     045  022516  033117  BASM1:  .ASCIZ  /%N%06/



016265     045  022516  052101  NULEVT: .ASCIZ  ?%N%ATHE DCLT EVENT LOG IS EMPTY?
016325     045  022516  037101  EVTF0:  .ASCIZ  ?%N%A>>>DCLT EVENT LOG ENTRY <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<?
016420  047045  042045  022465  EVTF1:  .ASCIZ  /%N%D5%A:%Z2%A:%Z2%S3%T/
016447     045  022516  031523  EVTF2:  .ASCIZ  /%N%S3%AADDR OF MSG=%06%S3%ABYTE COUNT=%D5/
016521     045  022516  031523  EVTF3:  .ASCIZ  /%N%S3%T%N/
016533     045  031523  047445  EVTF3C: .ASCIZ  /%S3%06%S3%06/
016550  051445  022463  033117  EVTF3D: .ASCIZ  /%S3%06%S3%06%S3%T/
016572  047045  051445  022463  EVTF4:  .ASCIZ  /%N%S3%AADDR OF MSG=%06%S3%ABYTE COUNT=%D5%S3%ANO. OF CMP ERRS=%D5/
016674  047045  051445  022463  EVTF4A: .ASCIZ  /%N%S3%AADDR OF MSG=%06%S3%ARX BYTES=%D5%S3%ACOMPARE BYTES=%D5/
016772  047045  051445  022463  EVTF4B: .ASCIZ  /%N%S3%APASS=%D5%S3%AERRORS=%D5%S3%ANOBUFFS=%D5/
017051     045  032523  040445  EVTF5A: .ASCIZ  /%S5%ABYTE # IN MSG.=%D5%S3%AEXPTD=%03%S3%ARECVD=%03/

017135     045  022516  034523  EVMOCG: .ASCIZ  /%N%S9%ACHANGED TO:/

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

017160  047045  051445  022470  EVMOHD: .ASCIZ  /%N%S8%AMODEM STATUS: CTS DSR DCD RTS RI  SQD TM/
017240  047045  051445  022471  EVMOST: .ASCII  /%N%S9%S9%S5%A/
017255     130     040     040  EVMCTS: .BYTE   'X,40,40,40
017261     130     040     040  EVMDSR: .BYTE   'X,40,40,40
017265     130     040     040  EVMDCD: .BYTE   'X,40,40,40
017271     130     040     040  EVMRTS: .BYTE   'X,40,40,40
017275     130     040     040  EVMRI:  .BYTE   'X,40,40,40
017301     130     040     040  EVMSQD: .BYTE   'X,40,40,40
017305     130     040     040  EVMTM:  .BYTE   'X,40,40,40
017311     000                          .BYTE   0
                                        .EVEN
```

```
                                .SBTTL  BASE TABLE ADDRESS
                                ;THIS SECTION IS USED BY A M9301-YJ BOOT ROM FOR DOING DOWN-LINE-LOAD.
                                ;MUST BE IN THE AREA OF "017370 + 256. BYTES" + A FEW

                                ;;;;;!!!!!! BEWARE !!!! DO NOT ALLOW THE ABOVE ASCIZ MESSAGES TO EXPAND INTO
                                ;;;;;!!!!!!!!!!!!!!!!!!!!  THIS REGION.
                                .EVEN
            017370                      .=17370
017370      000400              BASE:   .BLKB   256.            ;BASE TABLE ADDRESS
            020000                      .=20000
```

```
                                        .SBTTL  ASCIZ MESSAGES CONTINUED AFTER BASE TABLE REGION


                                        ;EXECUTION STATUS MESSAGES TO BE PRINTED TO KEEP OPERATOR AWAKE
020000  047045  000                     CR:     .ASCIZ  /%N/                    ;CR FOR LINES IN A ROW
020003  045  031523  040445             STXQ:   .ASCIZ  /%S3%ATXQ/              ;ABOUT TO TRANSMIT
020014  051445  022463  052101          STXC:   .ASCIZ  /%S3%ATXC/              ;TX COMPLETED
020025  045  031523  040445             SRXQ:   .ASCIZ  /%S3%ARXQ/              ;ABOUT TO RECEIVE
020036  051445  022463  042501          SDVE:   .ASCIZ  /%S3%AERR/              ;DEVICE ERROR
020047  045  031523  040445             SCM:    .ASCIZ  /%S3%ACMP/              ;ABOUT TO DO DATA CHECKING OF RECVD VS. EXPTD
020060  051445  022463  044501          SDVI:   .ASCIZ  /%S3%AINI/              ;DEVICE ABOUT TO BE INITIALIZED
020071  045  031523  040445             SCML:   .ASCIZ  /%S3%ACML/              ;COMPARE LENGTH ERROR
020102  051445  022463  041501          SCMD:   .ASCIZ  /%S3%ACMD/              ;COMPARE DATA ERROR
020113  045  031523  040445             SEOP:   .ASCIZ  /%S3%AEOP/              ;END  OF PASS
020124  051445  022463  046501          SMSC:   .ASCIZ  /%S3%AMSC/              ;MODEM STATUS CHANGE  ADDED BY EC


                                        ;REV B BY EC
                                        ;;NEXT ASCIZ LINES ARE USED IN SATELLITE ID MESSAGES
020135  045  022516  051501             SECRM:  .ASCIZ  /%N%ASECONDARY BOOT REQ FROM %T%A DEVICE-TYPE= %D3/
020217  104  000120                     DPM:    .ASCIZ  /DP/
020222  052504  000                     DUM:    .ASCIZ  /DU/
020225  104  000114                     DLM:    .ASCIZ  /DL/
020230  050504  000                     DQM:    .ASCIZ  /DQ/
020233  104  000101                     DAM:    .ASCIZ  /DA/
020236  052504  000120                  DUPM:   .ASCIZ  /DUP/
020242  046504  000103                  DMCM:   .ASCIZ  /DMC/
020246  047104  000                     DNM:    .ASCIZ  /DN/
020251  104  053114  000                DLVM:   .ASCIZ  /DLV/
020255  104  050115  000                DMPM:   .ASCIZ  /DMP/
020261  104  042524  000                DTEM:   .ASCIZ  /DTE/
020265  104  000126                     DVM:    .ASCIZ  /DV/
020270  055104  000                     DZM:    .ASCIZ  /DZ/
020273  125  045516  047516             UNKM:   .ASCIZ  /UNKNOWN/
020303  113  050104  000                KDPM:   .ASCIZ  /KDP/
020307  113  055104  000                KDZM:   .ASCIZ  /KDZ/
020313  113  000114                     KLM:    .ASCIZ  /KL/
020316  046504  000126                  DMVM:   .ASCIZ  /DMV/
                                                .EVEN
```

```
                                        ;REV B BY EC
                                        .SBTTL  DMR BASE TABLE DESCRIPTION MESSAGES
020322  047125  042504  044506  DMUNKN: .ASCIZ  /UNDEFINED DATA /       ;LOCATION UNDEFINED BY SPEC
020342  040502  042523  052040  DMR000: .ASCIZ  /BASE TABLE UPDATE INDEX POINTER/
020402  040502  042523  052040  DMR001: .ASCIZ  /BASE TABLE UPDATE LIMIT/
020432  042502  044507  047116  DMR002: .ASCIZ  /BEGINNING OF BASE TABLE DATA/
020467     116  045501  020123  DMR003: .ASCIZ  /NAKS RCVD..BUFFER TEMP UNAVAILABLE/
020532  040516  051513  051040  DMR004: .ASCIZ  /NAKS RCVD..HEADER BCC ERROR/
020566  040516  051513  051040  DMR005: .ASCIZ  /NAKS RCVD..DATA BCC ERROR/
020620  040516  051513  051440  DMR006: .ASCIZ  /NAKS SENT..BUFFER TEMP UNAVAILABLE/
020663     116  045501  020123  DMR007: .ASCIZ  /NAKS SENT..HEADER BCC ERROR/
020717     116  045501  020123  DMR010: .ASCIZ  /NAKS SENT..DATA BCC ERROR/
020751     122  050105  020123  DMR011: .ASCIZ  /REPS SENT..CUMUL REP SENT/
021003     122  050105  020123  DMR012: .ASCIZ  /REPS RCVD..CUMUL REP RCVD/
021035     116  045501  020123  DMR013: .ASCIZ  /NAKS RCVD..REP RESPONSE/
021065     116  045501  020123  DMR014: .ASCIZ  /NAKS RCVD..RCV OVERRUN/
021114  040516  051513  051040  DMR015: .ASCIZ  /NAKS RCVD..MSG HDR FORMAT/
021146  040516  051513  051040  DMR016: .ASCIZ  /NAKS RCVD..MSG TOO LONG/
021176  040516  051513  051440  DMR017: .ASCIZ  /NAKS SENT..REP RESPONSE/
021226  040516  051513  051440  DMR020: .ASCIZ  /NAKS SENT..RCV OVERRUN/
021255     116  045501  020123  DMR021: .ASCIZ  /NAKS SENT..MSG HDR FORMAT/
021307     130  044515  020124  DMR022: .ASCIZ  /XMIT UNDERRUN COUNT/
021333     103  046101  020114  DMR023: .ASCIZ  /CALL SET UP FAILURE COUNT/
021365     101  052103  020123  DMR024: .ASCIZ  /ACTS FAILURE COUNT/
021410  040503  051122  042511  DMR025: .ASCIZ  /CARRIER DETECT LOST COUNT(WHILE RECEIVING)/
021463     122  041505  044505  DMR026:.ASCIZ   /RECEIVER INACTIVE COUNT/
021513     123  051124  040505  DMR027: .ASCIZ  /STREAMING TIME-OUT COUNT/
021544  046530  054502  024124  DMR030: .ASCIZ  /XMBYT(LSB) - TOTAL # BYTES XMITTED, 32 BIT COUNTER/
021627     130  041115  052131  DMR031: .ASCIZ  ?XMBYT(2/4) - # BYTES XMITTED?
021664  046530  054502  024124  DMR032: .ASCIZ  ?XMBYT(3/4) - # BYTES XMITTED?
021721     130  041115  052131  DMR033: .ASCIZ  ?XMBYT(MSB) - # BYTES XMITTED?
021756  041522  054502  024124  DMR034: .ASCIZ  /RCBYT(LSB) - TOTAL # BYTES RECEIVED, 32 BIT COUNTER/
022042  041522  054502  024124  DMR035: .ASCIZ  ?RCBYT(2/4) - # BYTES RECEIVED (CONT)?
022107     122  041103  052131  DMR036: .ASCIZ  ?RCBYT(3/4) - # BYTES RECEIVED (CONT)?
022154  041522  054502  024124  DMR037: .ASCIZ  /RCBYT(MSB) - # BYTES RECEIVED/
022212  047111  047503  050115  DMR040: .ASCIZ  /INCOMPLETE SELECTION COUNT/
022245     116  020117  042522  DMR041: .ASCIZ  /NO REPLY TO SELECTION COUNTER/
022303     110  043511  042510  DMR042: .ASCIZ  /HIGHEST MESSAGE SUCCESSFULLY RECEIVED/
022351     110  043511  042510  DMR043: .ASCIZ  /HIGHEST MESSAGE TRANSMITTED/
022405     110  043511  042510  DMR044: .ASCIZ  /HIGHEST MESSAGE ACKNOWLEGED/
022441     116  054105  020124  DMR045: .ASCIZ  /NEXT MESSAGE TO TRANSMIT/
022472  040514  052123  046440  DMR046: .ASCIZ  /LAST MESSAGE TO COMPLETE TRANSMISSION/
022540  052503  051122  047105  DMR047: .ASCIZ  /CURRENT MESSAGE BEING TRANSMITTED/
022602  051124  047101  046523  DMR050: .ASCIZ  /TRANSMIT END OF QUEUE/
022630  051124  047101  046523  DMR051: .ASCIZ  /TRANSMIT BEGINNING OF QUEUE/
022664  042522  042503  053111  DMR052: .ASCIZ  /RECEIVE END OF QUEUE/
022711     122  041505  044505  DMR053: .ASCIZ  /RECEIVE BEGINNING OF QUEUE/
022744  040514  042524  052123  DMR054: .ASCIZ  /LATEST NAK REASON/
022766  051120  043517  040522  DMR055: .ASCIZ  ?PROGRAMMABLE REP/SELECT-TIMER PRESET VALUE?
023041     111  052123  052122  DMR056: .ASCIZ  ?ISTRT/ASTRT/REP/SELECT-TIMER COMPARE LEVEL?
023114  041501  044524  042526  DMR057: .ASCIZ  /ACTIVE TIME COUNT/
023136  044124  042522  044123  DMR060: .ASCIZ  /THRESHOLD LEVEL NAKS RCVD/
023170  044124  042522  044123  DMR061: .ASCIZ  /THRESHOLD COUNT NAKS RCVD/
023222  044124  042522  044123  DMR062: .ASCIZ  /THRESHOLD LEVEL NAKS SEND EXCEPT NO BUF/
023272  044124  042522  044123  DMR063: .ASCIZ  /THRESHOLD COUNT NAKS SEND EXCEPT NO BUF/
023342  044124  042522  044123  DMR064: .ASCIZ  /THRESHOLD LEVEL - REPS SENT/
```

```
023376  044124  042522  044123  DMR065: .ASCIZ  /THRESHOLD COUNT - REPS SENT/
023432  044124  042522  044123  DMR066: .ASCIZ  /THRESHOLD LEVEL - NO BUF AVAILABLE/
023475     124  051110  051505  DMR067: .ASCIZ  /THRESHOLD COUNT - NO BUF AVAILABLE/
023540  042523  020105  046504  DMR177: .ASCIZ  /SEE DMR TECH MANUAL FOR DESCRIPTION/
```

```
                                        ;REV B BY EC
                                        .SBTTL  DMC BASE TABLE DATA DESCRIPTION MESSAGES
023604  047101  020120  020055  DMC002: .ASCIZ  /ANP - CONSTANT 0/
023625     116  046124  020122  DMC003: .ASCIZ  /NTLR - NAKS RCVD..NO BUFFERS/
023662  044116  051104  026440  DMC004: .ASCIZ  /NHDR - NAKS RCVD..MSG HEADER BAD/
023723     104  052101  020122  DMC005: .ASCIZ  /DATR - NAKS RCVD..DATA BAD/
023756  052116  051514  026440  DMC006: .ASCIZ  /NTLS - NAKS SENT..NO BUFFERS/
024013     116  042110  020123  DMC007: .ASCIZ  /NHDS - NAKS SENT..BAD HEADER/
024050  040504  051524  026440  DMC010: .ASCIZ  /DATS - NAKS SENT..BAD DATA/
024103     122  050105  051503  DMC011: .ASCIZ  /REPCS - REPS SENT/
024125     122  050105  051103  DMC012: .ASCIZ  /REPCR - REPS RECD/
024147     102  051501  020105  DMC013: .ASCIZ  /BASE - CORE TABLE BASE ADDRESS/
024206  042523  020105  046504  DMC377: .ASCIZ  /SEE DMC TECH MANUAL FOR DESCRIPTION/
```

```
                                ;DEVICE ERROR MESSAGES

024252  044524  042515  047440  DVEM0:  .ASCII  /TIME OUT WAITING FOR RDI TO CLEAR/
024313     015  020012  020040          .ASCIZ  <15><12>/   SEL0     SEL2  /
024340  044524  042515  047440  DVEM1:  .ASCII  /TIME OUT WAITING FOR RDI TO SET/
024377     015  020012  020040          .ASCIZ  <15><12>/   SEL0     SEL2  /
024424  044524  042515  047440  DVEM3:  .ASCII  /TIME OUT WAITING FOR RUN TO SET/
024463     015  020012  020040          .ASCIZ  <15><12>/   SEL0     SEL2  /
024510  044524  042515  047440  DVEM4:  .ASCII  /TIME OUT WAITING FOR OUTPUT INTERRUPT/
024555     015  020012  020040          .ASCIZ  <15><12>/   SEL0     SEL2  /
024602  047111  052520  020124  DVEM5:  .ASCII  /INPUT INTERRUPT WHEN EXPECTING OUTPUT/
024647     015  020012  020040          .ASCIZ  <15><12>/   SEL0     SEL2  /
024674  046111  042514  040507  DVEM6:  .ASCII  /ILLEGAL OUTPUT INTERRUPT/
024724  005015  020040  051440          .ASCIZ  <15><12>/   SEL2     SEL6  /
024751     103  047117  051124  DVEM7:  .ASCII  /CONTROL OUT INSTEAD OF BA-CC OUT/
025011     015  020012  020040          .ASCIZ  <15><12>/   SEL2     SEL6  /

025036  054124  041040  043125  DVEM8:  .ASCII  /TX BUFF COMPLETED AND SHOULD BE RX/
025100  005015  020040  051440          .ASCIZ  <15><12>/   SEL4     SEL6  /
025125     122  020130  052502  DVEM9:  .ASCII  /RX BUFF COMPLETED AND SHOULD BE TX/
025167     015  020012  020040          .ASCIZ  <15><12>/   SEL4     SEL6  /
025214  042040  053517  020116  DLLAB:  .ASCII  / DOWN LINE LOAD ABORTED/
025243     015  020012  020040          .ASCIZ  <15><12>/   RXBUF    TXBUF /

025270  051120  041517  042105  PROEM:  .ASCIZ  /PROCEDURE ERROR/
025310  047516  020116  054105  NXMM:   .ASCIZ  /NON EXIST MEM/
025326  042104  046503  020120  DDCSRM: .ASCIZ  /DDCMP START REC/
025346  044504  041523  047117  DISCOM: .ASCIZ  /DISCONNECT/
025361     114  051517  020124  LOSDAM: .ASCIZ  /LOST DATA/
025373     104  041504  050115  DDCMRM: .ASCIZ  /DDCMP MAINT REC/
025413     124  046511  020105  TIMOM:  .ASCIZ  /TIME OUT/
025424  040504  040524  041440  DATCKM: .ASCIZ  /DATA CHECK/
025437     122  047125  051440  RUNSBM: .ASCIZ  /RUN SET ILLEGALLY/
025461     122  020130  042111  RXIDM:  .ASCIZ  /RX IDLE/
025471     103  020104  046107  CDGLM:  .ASCIZ  /CD GLITCHED/
025505     103  051524  043040  CTSFM:  .ASCIZ  /CTS FALILED/
025521     124  020130  047516  TXNC:   .ASCIZ  /TX NOT COMPLETE/
025541     122  020130  047516  RXNC:   .ASCIZ  /RX NOT COMPLETE/
025561     123  041505  051040  RXM1:   .ASCIZ  /SEC REQ ERR WORD 1/
025604  042523  020103  042522  RXM2:   .ASCIZ  /SEC REQ ERR WORD 2/
        025630                          .EVEN
```

```
                                GLOBAL ERROR REPORT SECTION

                                .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

3164
3165  025630                            BGNMSG  ERR1
3166  025630                                                                    ERR1::
3167  025630                            PRINTB  #EVTF5A,OFSET,<B,GOOD>,<B,BAD>  ;INDIVIDUAL DATA COMPARE ERROR
3168  025630  005046                                                                   CLR     -(SP)
3169  025632  153716  007447                                                          BISB    BAD,(SP)
3170  025636  005046                                                                   CLR     -(SP)
3171  025640  153716  007446                                                          BISB    GOOD,(SP)
3172  025644  013746  007424                                                          MOV     OFSET,-(SP)
3173  025650  012746  017051                                                          MOV     #EVTF5A,-(SP)
3174  025654  012746  000004                                                          MOV     #4,-(SP)
3175  025660  010600                                                                   MOV     SP,R0
3176  025662  104414                                                                   TRAP    C$PNTB
3177  025664  062706  000012                                                          ADD     #12,SP
3178  025670                            ENDMSG
3179  025670                                                                  L10001:
3180  025670  104423                                                                   TRAP    C$MSG
3181
3182  025672                            BGNMSG  ERR2
3183  025672                                                                    ERR2::
3184  025672                            PRINTB  #EFM2,TEMP4                     ;TOTAL DATA COMPARE FAILS ERROR
3185  025672  013746  007436                                                          MOV     TEMP4,-(SP)
3186  025676  012746  015457                                                          MOV     #EFM2,-(SP)
3187  025702  012746  000002                                                          MOV     #2,-(SP)
3188  025706  010600                                                                   MOV     SP,R0
3189  025710  104414                                                                   TRAP    C$PNTB
3190  025712  062706  000006                                                          ADD     #6,SP
3191  025716                            ENDMSG
3192  025716                                                                  L10002:
3193  025716  104423                                                                   TRAP    C$MSG
3194
3195  025720                            BGNMSG  ERR10
3196  025720                                                                  ERR10::
3197  025720                            PRINTB  #EFM11,R4,TEMP3
3198  025720  013746  007434                                                          MOV     TEMP3,-(SP)
3199  025724  010446                                                                   MOV     R4,-(SP)
3200  025726  012746  015554                                                          MOV     #EFM11,-(SP)
3201  025732  012746  000003                                                          MOV     #3,-(SP)
3202  025736  010600                                                                   MOV     SP,R0
3203  025740  104414                                                                   TRAP    C$PNTB
3204  025742  062706  000010                                                          ADD     #10,SP
3205  025746                            ENDMSG
3206  025746                                                                  L10003:
```

```
3207  025746  104423                                                          TRAP    C$MSG
3208
3209  025750                          BGNMSG  ERR8
3210  025750                                                          ERR8::
3211  025750                          PRINTB  #EVTF3D,TEMP3,TEMP4,CONOTM
3212  025750  013746  007442                                                  MOV     CONOTM,-(SP)
3213  025754  013746  007436                                                  MOV     TEMP4,-(SP)
3214  025760  013746  007434                                                  MOV     TEMP3,-(SP)
3215  025764  012746  016550                                                  MOV     #EVTF3D,-(SP)
3216  025770  012746  000004                                                  MOV     #4,-(SP)
3217  025774  010600                                                          MOV     SP,R0
3218  025776  104414                                                          TRAP    C$PNTB
3219  026000  062706  000012                                                  ADD     #12,SP
3220  026004                          PRINTB  #PCPM,PCADD
3221  026004  013746  007462                                                  MOV     PCADD,-(SP)
3222  026010  012746  015522                                                  MOV     #PCPM,-(SP)
3223  026014  012746  000002                                                  MOV     #2,-(SP)
3224  026020  010600                                                          MOV     SP,R0
3225  026022  104414                                                          TRAP    C$PNTB
3226  026024  062706  000006                                                  ADD     #6,SP
3227  026030                          ENDMSG
3228  026030                                                          L10004:
3229  026030  104423                                                          TRAP    C$MSG
3230
3231  026032                          BGNMSG  ERR9
3232  026032                                                          ERR9::
3233  026032                          PRINTB  #EVTF3C,TEMP3,TEMP4
3234  026032  013746  007436                                                  MOV     TEMP4,-(SP)
3235  026036  013746  007434                                                  MOV     TEMP3,-(SP)
3236  026042  012746  016533                                                  MOV     #EVTF3C,-(SP)
3237  026046  012746  000003                                                  MOV     #3,-(SP)
3238  026052  010600                                                          MOV     SP,R0
3239  026054  104414                                                          TRAP    C$PNTB
3240  026056  062706  000010                                                  ADD     #10,SP
3241  026062                          PRINTB  #PCPM,PCADD
3242  026062  013746  007462                                                  MOV     PCADD,-(SP)
3243  026066  012746  015522                                                  MOV     #PCPM,-(SP)
3244  026072  012746  000002                                                  MOV     #2,-(SP)
3245  026076  010600                                                          MOV     SP,R0
3246  026100  104414                                                          TRAP    C$PNTB
3247  026102  062706  000006                                                  ADD     #6,SP
3248  026106                          ENDMSG
3249  026106                                                          L10005:
3250  026106  104423                                                          TRAP    C$MSG
3251
3252  026110                          BGNMSG  ERR13
3253  026110                                                          ERR13::
3254  026110                          PRINTB  #EVTF3C,TEMP3,TEMP4
3255  026110  013746  007436                                                  MOV     TEMP4,-(SP)
3256  026114  013746  007434                                                  MOV     TEMP3,-(SP)
3257  026120  012746  016533                                                  MOV     #EVTF3C,-(SP)
3258  026124  012746  000003                                                  MOV     #3,-(SP)
3259  026130  010600                                                          MOV     SP,R0
3260  026132  104414                                                          TRAP    C$PNTB
3261  026134  062706  000010                                                  ADD     #10,SP
3262  026140                          ENDMSG
```

```
3263  026140
3264  026140  104423
3265
3266  026142                          BGNMSG  ERR14
3267  026142
3268  026142                          PRINTB  #EVTF3D,TEMP3,TEMP4,CONOTM
3269  026142  013746  007442
3270  026146  013746  007436
3271  026152  013746  007434
3272  026156  012746  016550
3273  026162  012746  000004
3274  026166  010600
3275  026170  104414
3276  026172  062706  000012
3277  026176                          ENDMSG
3278  026176
3279  026176  104423
3280
3281  026200                          EXIT    MSG
3282  026200  000167
3283  026202  177772
3284
3285
```

```
L10006:
        TRAP    C$MSG

ERR14::
        MOV     CONOTM,-(SP)
        MOV     TEMP4,-(SP)
        MOV     TEMP3,-(SP)
        MOV     #EVTF3D,-(SP)
        MOV     #4,-(SP)
        MOV     SP,R0
        TRAP    C$PNTB
        ADD     #12,SP
L10007:
        TRAP    C$MSG

        .WORD   J$JMP
        .WORD   L10007-2-.
```

```
                                .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
                                ;--

026204                          CLKSET:
026204  012122                          MOV     (R1)+,(R2)+             ;LOAD CLOCK'S CSR ADDR. INTO "CLKCSR"
026206  012112                          MOV     (R1)+,(R2)              ;LOAD CLOCK'S INT. LEVEL INTO "CLKBR"
026210  006312                          ASL     (R2)                    ;ADJUST THE INT. LEVEL FOR LOADING INTO
026212  006312                          ASL     (R2)                    ;   THE PSW WITH A "SETVEC" CALL
026214  006312                          ASL     (R2)
026216  006312                          ASL     (R2)
026220  006322                          ASL     (R2)+
026222  012122                          MOV     (R1)+,(R2)+             ;LOAD CLOCK'S INT. VECTOR INTO "CLKVEC"
026224  012122                          MOV     (R1)+,(R2)+             ;LOAD CLOCK'S HERTZ RATE INTO "CLKHZ"
026226  000207                          RTS     PC
```

```
3333
3334                                    .SBTTL          CLOCK INTERRUPT SERVICE ROUTINE
3335                                    ;++
3336                                    ; FUNCTIONAL DESCRIPTION:
3337                                    ;       THIS IS THE CLOCK INTERRUPT SERVICE ROUTINE WHICH TAKES CARE OF
3338                                    ;       KEEPING THE "TIME-SINCE-START" AND COUNTING DOWN ANY OF THE
3339                                    ;       "EVENT" TIMERS. THE TIMERS ARE USED TO TIME COMPLETION OF  DEVICE
3340                                    ;       REQUESTS. THE "TIME-SINCE-START" IS USED TO BE LOGGED WITH EACH ENTRY
3341                                    ;       INTO THE EVENT LOG.
3342                                    ;
3343                                    ; IMPLICIT INPUTS:
3344                                    ;       TIMTCK: THE CURRENT NO. OF TICKS LEFT TO BE COUNTED UNTIL A SECOND
3345                                    ;               HAS BEEN COUNTED OFF
3346                                    ;       CLKHZ:  THE NO. OF TICKS IN A SECOND, DETERMINED BY THE SYS. LINE FREQ.
3347                                    ;       TIMMIN & TIMSEC: CURRENT VALUE OF "TIME-SINCE-START"
3348                                    ;                        IN MINUTES & SECONDS
3349                                    ;       TIMER 1,2, & S: CURRENT VALUES OF THE "EVENT TIMERS"
3350                                    ;
3351                                    ; IMPLICIT OUTPUTS:
3352                                    ;       NEW VALUE OF EVENT TIMER "1" DECREMENTED BY 1 TICK IF IT WAS NON-ZERO
3353                                    ;       NEW VALUE OF EVENT TIMER "2" DECREMENTED BY 1 TICK IF IT WAS NON-ZERO
3354                                    ;       NEW VALUE OF EVENT TIMER "S" DECREMENTED BY 1 SECOND IF IT WAS NON-ZERO
3355                                    ;
3356                                    ; FUNCTIONAL SIDE EFFECTS:
3357                                    ;       THE CLOCK IS DISABLED UPON ENTRY AND REENABLED WHEN LEAVING
3358                                    ;
3359                                    ; CALLING SEQUENCE:
3360                                    ;       THIS ROUTINE IS CALLED WHEN THE CLOCK INTERRUPTS THRU "CLKVEC".
3361                                    ;       THE ADDRESS OF THIS ROUTINE WAS LOADED INTO THE CLOCK'S INTERRUPT
3362                                    ;       VECTOR WITH A SUPERVISOR "SETVEC" CALL.
3363                                    ;--
3364
3365  026230                                    BGNSRV  CLKINT
3366  026230                                                                    CLKINT::
3367
3368  026230  005077  161266                    CLR     @CLKCSR         ;DISABLE THE CLOCK FORM INTERRUPTING
3369  026234  005337  007540                    DEC     TIMTCK          ;DECREMENT THE # OF TICKS/SEC.
3370  026240  001015                            BNE     1$              ;GO CHECK TIMERS (1&2-TICKS, 3-SECONDS)
3371  026242  013737  007530  007540            MOV     CLKHZ,TIMTCK    ;RESET THE # OF TICKS/SEC.
3372  026250  005237  007536                    INC     TIMSEC          ;INC # OF SECS-SINCE-START
3373  026254  022737  000074  007536            CMP     #60.,TIMSEC     ;SEE IF WE'VE COUNTED 60 SECS. YET
3374  026262  001004                            BNE     1$              ;IF NOT, GO CHECK TIMERS
3375  026264  005237  007534                    INC     TIMMIN          ; ELSE INC MINUTES-SINCE-START
3376  026270  005037  007536                    CLR     TIMSEC          ; AND RESTART SECOND COUNTER
3377
3378  026274  005737  007542            1$:     TST     TIMER1          ;SEE IF TIMER #1, TIMING ANYTHING
3379  026300  001402                            BEQ     2$              ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
3380  026302  005337  007542                    DEC     TIMER1          ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
3381  026306  005737  007544            2$:     TST     TIMER2          ;SEE IF TIMER #2, TIMING ANYTHING
3382  026312  001402                            BEQ     3$              ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
3383  026314  005337  007544                    DEC     TIMER2          ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
3384  026320  005737  007546            3$:     TST     TIMERS          ;SEE IF TIMER #3, TIMING ANYTHING
3385  026324  001406                            BEQ     4$              ; IF=0, NOTHING BEING TIMED, LEAVE
3386  026326  023737  007530  007540            CMP     CLKHZ,TIMTCK    ;SEE IF A SECOND HAS BEEN COUNTED OFF
3387  026334  001002                            BNE     4$              ; BR IF NO
3388  026336  005337  007546                    DEC     TIMERS          ; ELSE DECREMENT THE TIMER VALUE (BY 1 SEC.)
```

```
 3389  026342  013777  007532  161152  4$:     MOV     CLKEN,@CLKCSR   ;REENABLE THE CLOCK TO INTERRUPT
 3390  026350                                  ENDSRV
 3391  026350                                                                          L10010:
 3392  026350  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
                                ;--

026352                          LOGTXQ:
026352  012737  020003  007430          MOV     #STXQ,TEMP1     ;SET UP MSG. TO PRINT
026360  012737  000000  007426          MOV     #TXQ,TEMP       ;SET UP EVENT TYPE
026366  000510                          BR      LOGS1           ;GO LOG EVENT AND TIME

026370                          LOGTXC:
026370  012737  020014  007430          MOV     #STXC,TEMP1     ;SET UP MSG. TO PRINT
026376  012737  000002  007426          MOV     #TXC,TEMP       ;SET UP EVENT TYPE
026404  000501                          BR      LOGS1           ;GO LOG EVENT AND TIME

026406                          LOGRXQ:
026406  012737  020025  007430          MOV     #SRXQ,TEMP1     ;SET UP MSG. TO PRINT
026414  012737  000004  007426          MOV     #RXQ,TEMP       ;SET UP EVENT TYPE
026422  000472                          BR      LOGS1           ;GO LOG EVENT AND TIME

026424                          LOGRXC:
026424  012737  000006  007426          MOV     #RXC,TEMP       ;SET UP EVENT TYPE
026432  000466                          BR      LOGS1           ;GO LOG EVENT AND TIME
026434                          LGDVE:
```

```
                                                MOV     #SDVE,TEMP1     ;SET UP MSG. TO PRINT
                                                MOV     #DER,TEMP       ;SET UP EVENT TYPE
                                                BR      LOGS3           ;GO LOG EVENT AND TIME

                                        LOGDVI:
                                                MOV     #SDVI,TEMP1     ;SET UP MSG. TO PRINT
                                                MOV     #DVI,TEMP       ;SET UP EVENT TYPE
                                                MOVB    MODTYP,TEMP2
                                                MOVB    MLTYP,TEMP2+1
                                                MOV     RPASS,TEMP3
                                                MOV     PARAM,TEMP4     ;SET UP EVNT ENTRIES
                                                BR      LOGS3           ;GO LOG EVENT AND TIME

                                        LOGCMP:
                                                MOV     #SCM,TEMP1      ;SET UP MSG. TO PRINT
                                                MOV     #DCK,TEMP       ;SET UP EVENT TYPE
                                                BR      LOGS3
                                        LOGCML:
                                                MOV     #SCML,TEMP1
                                                MOV     #DLE,TEMP       ;SET UP MSG. AND TYPE
                                                BR      LOGS3           ;GO LOG EVENT AND TIME
                                        LOGCMD:
                                                MOV     #SCMD,TEMP1
                                                MOV     #DDE,TEMP
                                                BR      LOGS3           ;GO LOG MSG TYPE AND TIME
                                        LOGEOP:
                                                MOV     #SEOP,TEMP1
                                                MOV     #EOP,TE..
                                                BR      LOGS3           ;GO LOG MSG TYPE AND TIME

                                        LOGS1:  MOV     ERRCNT,-(SP)    ;SAVE CURRENT ERROR COUNT
                                                JSR     PC,DVMODS       ; GO GET MODEM STATUS
                                                MOV     (SP)+,R4        ;GET SAVED ERRCNT VALUE
                                                CMP     R4,ERRCNT       ;WHERE ANY ERRORS FOUND
                                                BEQ     1$              ; BR IF NONE
                                                JMP     LOGEX           ; ELSE, LEAVE WITHOUT LOGGING ANYTHING
                                                                        ;    BUT THE DEVICE ERROR FROM "DVMODS"
                                        1$:     MOV     MODS,TEMP4      ;AND PUT IT IN TEMP4

                                        LOGS3:
                                                CMP     #RXC,TEMP
                                                BEQ     LOGS5           ;IF RXC DONT PRINT
                                                BIT     #STATB,PARAM
                                                BEQ     LOGS5           ;IF NO STATUS SELECTED
                                                                        ;GO TO 5


                                                CMP     #10,LNCNT       ;HAVE WE DONE 10?
                                                BNE     LOGS4           ;IF NOT GO TO 4
                                                CLR     LNCNT           ;ESLE CLEAR IT

                                                PRINTF  #CR             ;ELSE PRINT CR
                                                                                MOV     #CR,-(SP)
                                                                                MOV     #1,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTF
                                                                                ADD     #4,SP
```

```
3505  026716
3506  026716  005237  007372                LOGS4:
3507  026722                                      INC     LNCNT           ;INC COUNTER OF # OF AWAKE MSGS
3508  026722  013746  007430                      PRINTF  TEMP1           ;PRINT OPERATOR AWAKE MSG.
3509  026726  012746  000001                                                      MOV     TEMP1,-(SP)
3510  026732  010600                                                             MOV     #1,-(SP)
3511  026734  104417                                                             MOV     SP,R0
3512  026736  062706  000004                                                     TRAP    C$PNTF
3513  026742  010346                                                             ADD     #4,SP
3514  026744  013703  007550                LOGS5:  MOV     R3,-(SP)        ;SAVE R3 ON THE STACK
3515  026750  113723  007426                        MOV     EVTPTR,R3
3516  026754  013737  007530  007426              MOVB    TEMP,(R3)+      ;LOG EVENT
3517  026762  163737  007540  007426              MOV     CLKHZ,TEMP
3518  026770  113723  007426                      SUB     TIMTCK,TEMP
3519  026774  113723  007536                      MOVB    TEMP,(R3)+      ;LOG TIME SINCE START
3520  027000  113723  007534                      MOVB    TIMSEC,(R3)+
3521  027004  013723  007432                      MOVB    TIMMIN,(R3)+    ;TICKS,SECS AND MINS.
3522  027010  013723  007434                      MOV     TEMP2,(R3)+     ;LOG EVNT ENTRY 3
3523  027014  013723  007436                      MOV     TEMP3,(R3)+     ;LOG EVNT ENTRY 4
3524  027020  020327  010454                      MOV     TEMP4,(R3)+     ;LOG EVNT ENTRY 5
3525  027024  103404                            CMP     R3,#EVTEND
3526                                             BLO     LOGS2           ;IF EVENT LOG FULL GO
3527  027026  012713  177777                                             ;CONTINUE;ELSE GO TO 2
3528  027032  012703  007552                    MOV     #-1,(R3)        ;LOG A TABLE END
3529  027036  010337  007550                    MOV     #EVTLOG,R3      ;PUT R3 TO START OF TABLE
3530  027042  012603                LOGS2:  MOV     R3,EVTPTR       ;RESTORE POINTER
3531  027044  000207                        MOV     (SP)+,R3        ;RESTORE R3
3532                                 LOGEX:  RTS     PC
3533
```

```
                        .SBTTL  REPORT BASE TABLE OR EVENT LOG
                        ;REV B BY EC
                        ;THE FOLLOWING COMMANDS ADDED TO REVISION B CZCLK
                        ;;DMR/DMC DCLT PROGRAM
                        ;;RPT> LOG
                        ;;      BASE/ERROR
                        ;;      BASE/FULL
                        ;;      BASE/OFFSET=NN
                        ;;      HELP
                        ;;      EXIT

027046  010246  REPORT: MOV     R2,-(SP)        ;SAVE R2,R3,R4 ON THE STACK
027050  010346          MOV     R3,-(SP)
027052  010446          MOV     R4,-(SP)

                        ;PRINT HELP MESSAGE
027054                  PRINTF  #RHLP0          ;BASIC HELP MESSAGE
027054  012746  014024                                    MOV     #RHLP0,-(SP)
027060  012746  000001                                    MOV     #1,-(SP)
027064  010600                                            MOV     SP,R0
027066  104417                                            TRAP    C$PNTF
027070  062706  000004                                    ADD     #4,SP

027074  105037  003561  GETRCL: CLRB    P$GDBD          ;INIT GOOD/BAD FLAG  -1=BAD INPUT
027100  105037  003560          CLRB    P$NNUF          ;INIT MORE COMMAND LINE INPUT NEEDED

                        ;PRINT PROMPT 'RPT>'
027104                  GMANID  CLI$RP,CMDBUF,A,0,1,72.,NO
027104  104443                                            TRAP    C$GMAN
027106  000406                                            BR      10000$
027110  003060                                            .WORD   CMDBUF
027112  000142                                            .WORD   T$CODE
027114  012510                                            .WORD   CLI$RP
027116  000000                                            .WORD   0
027120  000001                                            .WORD   T$LOLIM
027122  000110                                            .WORD   T$HILIM
027124                                                  10000$:
027124  012737  003060  003544          MOV     #CMDBUF,P$BUFA  ;INPUT BUFFER
027132  012737  027266  003546          MOV     #CLIRT,P$TREE   ;REPORT CLI TREE
027140  012737  027464  003550          MOV     #CLIRAC,P$ACT   ;ACTION ROUTINES
027146  005037  003204                  CLR     QUALFG
027152  004737  032642                  JSR     PC,P$TRV        ;GO PARSE COMMAND LINE
027156  105737  003561                  TSTB    P$GDBD          ;COMMAND OK ?
027162  001412                          BEQ     1$              ;YES,BRANCH
027164                                  PRINTF  #CLIERM         ;PRINT INVALID INPUT MESSAGE
027164  012746  012516                                            MOV     #CLIERM,-(SP)
027170  012746  000001                                            MOV     #1,-(SP)
027174  010600                                                    MOV     SP,R0
027176  104417                                                    TRAP    C$PNTF
027200  062706  000004                                            ADD     #4,SP
027204  000137  027074                  JMP     GETRCL          ;TRY AGAIN

027210  105737  003560          1$:     TSTB    P$NNUF          ;MORE COMMAND NEEDED ?
027214  001412                          BEQ     10$             ;NO,BRANCH
027216                                  PRINTF  #CLINUF         ;INCOMPLETE MESSAGE
027216  012746  012546                                            MOV     #CLINUF,-(SP)
```

```
                                                                                        MOV     #1,-(SP)
                                                                                        MOV     SP,R0
                                                                                        TRAP    C$PNTF
                                                                                        ADD     #4,SP
027222  012746  000001
027226  010600
027230  104417
027232  062706  000004
027236  000137  027074            JMP     GETRCL                  ;TRY AGAIN
027242  023727  003202  000002  10$:  CMP     KEYWD1,#RPEXT           ;EXIT COMMAND ?
027250  001402                    BEQ     20$                     ;YES,BRANCH
027252  000137  027074            JMP     GETRCL                  ;GET ANOTHER COMMAND
027256  012604                20$:  MOV     (SP)+,R4                ;RESTORE R4
027260  012603                    MOV     (SP)+,R3                ;RESTORE R3
027262  012602                    MOV     (SP)+,R2                ;RESTORE R2
027264  000207                    RTS     PC                      ;RETURN
```

```
 3603                                .SBTTL  COMMAND LINE PARSING TREE FOR REPORT
 3604  027266                        CLIRT:  CLI CLISPA,0,R10$               ;SKIP SPACES IN COMMAND LINE
 3605  027272                        R10$:   CLI <'?>,RPHLP,R11$             ;IF INPUT = ? THEN PRINT HELP MESSAGE
 3606  027276                                CLI CLIEXI,0                    ;;AND EXIT PARSER
 3607  027300                        R11$:   CLI CLISTR,RPHLP,R12$,<'HELP'>  ;IF INPUT = 'HELP' THEN PRINT HELP
 3608  027314                                CLI CLIEXI,0                    ;;MESSAGE AND EXIT PARSER
 3609  027316                        R12$:   CLI CLISTR,RPEXT,R13$,<'EXIT'>  ;IF INPUT = 'EXIT' THEN SET KEYWORD =
 3610  027332                                CLI CLIEXI,0                    ;;RPEXT AND EXIT PARSER
 3611  027334                        R13$:   CLI CLISTR,RPLOG,R14$,<'LOG'>   ;IF INPUT = 'LOG' THEN GO PRINT EVENT
 3612  027346                                CLI CLIEXI,0                    ;;LOG AND EXIT PARSER
 3613  027350                        R14$:   CLI CLISTR,RNOTNF,R30$,<'BASE'>;IF INPUT = 'BASE' THEN MORE COMMAND
 3614  027364                                CLI CLIBR,0,R15$                ;;LINE IS NEEDED
 3615  027370                        R15$:   CLI <'/>,RNOTNF,R125$           ;IF INPUT = '/' THEN LOOK FOR MORE
 3616  027374                                CLI CLISTR,RPSWE,R16$,<'ERROR'> ;IF INPUT = 'ERROR' THEN GO PRINT
 3617  027410                                CLI CLIEXI,0                    ;;ERROR INFORMATION
 3618  027412                        R16$:   CLI CLISTR,RPSWF,R17$,<'FULL'>  ;IF INPUT = 'FULL' THEN GO PRINT
 3619  027426                                CLI CLIEXI,0                    ;;ENTIRE BASE TABLE
 3620  027430                        R17$:   CLI CLISTR,RNOTNF,R30$,<'OFFSET'>;IF INPUT = 'OFFSET' THEN LOOK FOR
 3621  027446                                CLI <'=>,0,R30$                 ;;'='
 3622  027452                                CLI CLIOCT,RPSWO,R30$           ;IF INPUT = OCTAL VALUE THEN GO
 3623  027456                                CLI CLIEXI,0                    ;;PRINT SINGLE BASE TABLE ITEM
 3624  027460                        R30$:   CLI CLIERR,0
 3625  027462                        R125$:  CLI CLIEXI,0
```

```
                                .SBTTL  CLI ACTION DISPATCHER AND ROUTINES
027464  006302                  CLIRAC: ASL     R2              ;SET UP INDEX
027466  016202  027502                  MOV     10$(R2),R2      ;
027472  062702  027502                  ADD     #10$,R2         ;
027476  004712                          JSR     PC,(R2)         ;GO DO ACTION
027500  000207                          RTS     PC              ;RETURN
027502  000026                  10$:    .WORD   ACTRNL-10$      ;NULL
027504  000030                          .WORD   ACTRHL-10$      ;HELP ROUTINE
027506  000074                          .WORD   ACTREX-10$      ;EXIT ROUTINE
027510  000104                          .WORD   ACTRLG-10$      ;REPORT EVENT LOG ROUTINE
027512  000120                          .WORD   ACTSWE-10$      ;REPORT ERRORS ROUTINE
027514  000262                          .WORD   ACTSWF-10$      ;REPORT ENTIRE BASE TABLE
027516  000374                          .WORD   ACTSWO-10$      ;REPORT SINGLE BASE ADDRESS
027520  000020                          .WORD   ACTRNF-10$      ;MORE COMMAND NEEDED

                                        ;MORE COMMAND NEEDED
027522  112737  177777  003560  ACTRNF: MOVB    #-1,P$NNUF      ;MORE COMMAND NEEDED
027530  000207                  ACTRNL: RTS     PC              ;NULL

                                        ;PRINT HELP MESSAGE
027532  012702  003230          ACTRHL: MOV     #RHLPTB,R2      ;INDEX FOR HELP MESSAGES
027536                          1$:     PRINTF  #HLPF,(R2)+     ;PRINT IT
027536  012246                                                          MOV     (R2)+,-(SP)
027540  012746  013231                                                  MOV     #HLPF,-(SP)
027544  012746  000002                                                  MOV     #2,-(SP)
027550  010600                                                          MOV     SP,R0
027552  104417                                                          TRAP    C$PNTF
027554  062706  000006                                                  ADD     #6,SP
027560  020227  003246                  CMP     R2,#RHLPEN      ;LAST MESSAGE ?
027564  001364                          BNE     1$              ;NO,BRANCH
027566  012737  000001  003202          MOV     #RPHLP,KEYWD1   ;SET KEYWORD
027574  000207                          RTS     PC              ;RETURN

                                        ;EXIT REPORT LEVEL
027576  012737  000002  003202  ACTREX: MOV     #RPEXT,KEYWD1   ;SET KEYWORD AND RETURN
027604  000207                          RTS     PC

                                        ;PRINT ERROR LOG
027606  004737  030574          ACTRLG: JSR     PC,REPLOG       ;GO PRINT EVENT LOG
027612  012737  000003  003202          MOV     #RPLOG,KEYWD1   ;SET KEYWORD
027620  000207                          RTS     PC              ;RETURN

                                        ;PRINT ONLY ERROR LOCATIONS
027622  005737  012412          ACTSWE: TST     OPTYP           ;DMR ?
027626  001026                          BNE     10$             ;YES,BRANCH
027630  012737  003432  007450          MOV     #DMCIND,INDEX   ;SETUP DMC MESSAGES
027636  062737  000006  007450          ADD     #6,INDEX        ;POINT TO CORRECT MESSAGE
027644  012737  003462  007452          MOV     #DMCEND,INDEXE  ;LAST DMC ADDRESS
027652  012737  017370  007454          MOV     #BASE,BEND      ;SET UP LAST ADDRESS
027660  062737  000012  007454          ADD     #12,BEND        ;;TO BE PRINTED
027666  012737  017370  007456          MOV     #BASE,BDATA     ;BASE TABLE START ADDRESS
027674  062737  000003  007456          ADD     #3,BDATA        ;ERROR START ADDRESS
027702  000425                          BR      20$             ;
027704  012737  003250  007450  10$:    MOV     #DMRIND,INDEX   ;SETUP FOR DMR MESSAGES
027712  062737  000006  007450          ADD     #6,INDEX        ;POINT TO FIRST ERROR MESSAGE
027720  012737  003430  007452          MOV     #DMREND,INDEXE  ;LAST DMR MESSAGE
```

```
 3682  027726  012737  017370  007454          MOV     #BASE,BEND      ;SETUP LAST ADDRESS
 3683  027734  062737  000041  007454          ADD     #41,BEND        ;;TO BE PRINTED
 3684  027742  012737  017370  007456          MOV     #BASE,BDATA     ;START ADDRESS BASE TABLE
 3685  027750  062737  000003  007456          ADD     #3,BDATA        ;START ADDRESS ERRORS
 3686  027756  004737  030336          20$:    JSR     PC,RPBASE       ;GO PRINT DATA
 3687  027762  000207                          RTS     PC              ;RETURN
 3688
 3689                                          ;PRINT FULL BASE TABLE
 3690  027764  005737  012412          ACTSWF: TST     OPTYP           ;DMR ?
 3691  027770  001020                          BNE     10$             ;YES,BRANCH
 3692  027772  012737  003432  007450          MOV     #DMCIND,INDEX   ;SETUP DMC MESSAGES
 3693  030000  012737  003462  007452          MOV     #DMCEND,INDEXE  ;LAST MESSAGE
 3694  030006  012737  017370  007454          MOV     #BASE,BEND      ;TABLE START ADDRESS
 3695  030014  062737  000377  007454          ADD     #377,BEND       ;PRINT 256. BYTES OF DATA
 3696  030022  012737  017370  007456          MOV     #BASE,BDATA     ;FIRST ADDRESS TO PRINT
 3697  030030  000417                          BR      20$
 3698  030032  012737  003250  007450  10$:    MOV     #DMRIND,INDEX   ;SETUP DMR MESSAGES
 3699  030040  012737  003430  007452          MOV     #DMREND,INDEXE  ;LAST DMR MESSAGE
 3700  030046  012737  017370  007454          MOV     #BASE,BEND      ;TABLE START ADDRESS
 3701  030054  062737  000177  007454          ADD     #177,BEND       ;PRINT 128. BYTES OF DATA
 3702  030062  012737  017370  007456          MOV     #BASE,BDATA     ;FIRST ADDRESS TO PRINT
 3703  030070  004737  030336          20$:    JSR     PC,RPBASE       ;GO PRINT DATA
 3704  030074  000207                          RTS     PC              ;RETURN
 3705
 3706                                          ;PRINT SINGLE TABLE LOCATION
 3707  030076  105037  003560          ACTSWO: CLRB    P$NNUF          ;INIT NOT ENOUGH FLAG
 3708  030102  005737  012412                  TST     OPTYP           ;DMR?
 3709  030106  001004                          BNE     5$              ;YES,BRANCH
 3710  030110  012737  000377  007454          MOV     #377,BEND       ;BASE TABLE FOR DMC = 256 BYTES
 3711  030116  000403                          BR      7$              ;BRANCH
 3712  030120  012737  000177  007454  5$:     MOV     #177,BEND       ;BASE TABLE FOR DMR = 128 BYTES
 3713  030126  023737  003554  007454  7$:     CMP     P$NUM,BEND      ;DMC = 256 BYTES   DMR = 128 BYTES
 3714  030134  101416                          BLOS    10$             ;YES,BRANCH
 3715  030136                                  PRINTF  #RPTIV,P$NUM    ;PRINT ILLEGAL VALUE
 3716  030136  013746  003554                                                  MOV     P$NUM,-(SP)
 3717  030142  012746  014374                                                  MOV     #RPTIV,-(SP)
 3718  030146  012746  000002                                                  MOV     #2,-(SP)
 3719  030152  010600                                                          MOV     SP,R0
 3720  030154  104417                                                          TRAP    C$PNTF
 3721  030156  062706  000006                                                  ADD     #6,SP
 3722  030162  112737  177777  003561          MOVB    #-1,P$GDBD      ;SET BAD DATA
 3723  030170  000461                          BR      30$             ;RETURN
 3724  030172  013701  003554          10$:    MOV     P$NUM,R1        ;OFFSET VALUE
 3725  030176  006301                          ASL     R1              ;MULTIPLY BY 2
 3726  030200  005737  012412                  TST     OPTYP           ;DMR ?
 3727  030204  001025                          BNE     15$             ;YES,BRANCH
 3728  030206  012737  003432  007450          MOV     #DMCIND,INDEX   ;DMC MESSAGES
 3729  030214  060137  007450                  ADD     R1,INDEX        ;GET RIGHT MESSAGE
 3730  030220  012737  003462  007452          MOV     #DMCEND,INDEXE  ;LAST DMC MESSAGE
 3731  030226  012737  017370  007454          MOV     #BASE,BEND      ;TABLE ADDRESS
 3732  030234  063737  003554  007454          ADD     P$NUM,BEND      ;LAST ADDRESS
 3733  030242  012737  017370  007456          MOV     #BASE,BDATA     ;BASE ADDRESS
 3734  030250  063737  003554  007456          ADD     P$NUM,BDATA     ;ADD OFFSET
 3735  030256  000424                          BR      20$             ;GO PRINT DATA
 3736  030260  012737  003250  007450  15$:    MOV     #DMRIND,INDEX   ;SETUP FOR DMR MESSAGES
 3737  030266  060137  007450                  ADD     R1,INDEX        ;GET CORRECT MESSAGE
```

```
 3738  030272  012737  003430  007452          MOV     #DMREND,INDEXE  ;LAST DMR MESSAGE
 3739  030300  012737  017370  007454          MOV     #BASE,BEND      ;TABLE ADDRESS
 3740  030306  063737  003554  007454          ADD     P$NUM,BEND      ;LAST ADDRESS
 3741  030314  012737  017370  007456          MOV     #BASE,BDATA     ;TABLE ADDRESS
 3742  030322  063737  003554  007456          ADD     P$NUM,BDATA     ;ADD OFFSET
 3743  030330  004737  030336          20$:    JSR     PC,RPBASE       ;GO PRINT SINGLE LOCATION
 3744  030334  000207                  30$:    RTS     PC              ;RETURN
```

```
                        ;;PRINT BASE TABLE SUBROUTINE
                        ;FUNCTIONAL DESCRIPTION - THIS ROUTINE IS USED TO PRINT DATA
                        ;       STORED IN THE BASE TABLE AREA IN MEMORY. THIS BASE
                        ;       TABLE IS UPDATED BY THE DMR OR DMC. THE USER HAS THE
                        ;       OPTION OF PRINTING THE FULL TABLE, PRINTING THE FIRST
                        ;       FEW ERROR LOCATIONS OR A SINGLE LOCATION.
                        ;
                        ;DEFINITIONS
                        ;               INDEX - CONTAINS POINTER TO DMR OR DMC DATA
                        ;                       DESCRIPTION MESSAGES.
                        ;               INDEXE - CONTAINS POINTER TO LAST DMR OR DMC
                        ;                       DESCRIPTION MESSAGES.
                        ;               BEND - LAST LOCATION IN TABLE TO BE PRINTED.
                        ;               BDATA - ADDRESS OF DATA TO BE PRINTED.
                        ;
                        ;       THE ABOVE VARIABLES MUST BE ASSIGNED THE CORRECT VALUES
                        ;       BEFORE THIS SUBROUTINE IS CALLED.


030336  010146          RPBASE: MOV     R1,-(SP)        ;SAVE R1
030340  010246                  MOV     R2,-(SP)        ;SAVE R2
030342                          PRINTF  #BTHEAD         ;PRINT BRIEF HEADER MESSAGE
030342  012746  030476                                                  MOV     #BTHEAD,-(SP)
030346  012746  000001                                                  MOV     #1,-(SP)
030352  010600                                                          MOV     SP,R0
030354  104417                                                          TRAP    C$PNTF
030356  062706  000004                                                  ADD     #4,SP
030362  013702  007450                  MOV     INDEX,R2        ;POINTER TO MESSAGES
030366  013701  007456                  MOV     BDATA,R1        ;ADDRESS OF DATA
030372  010137  007434          10$:    MOV     R1,TEMP3        ;SAVE CURRENT ADDRESS OF DATA
030376  112137  007430                  MOVB    (R1)+,TEMP1     ;READ DATA
030402  020237  007452                  CMP     R2,INDEXE       ;END OF MESSAGES?
030406  002402                          BLT     20$             ;NO,BRANCH
030410  013702  007452                  MOV     INDEXE,R2       ;'SEE MANUAL' MESSAGE
030414  012237  007432          20$:    MOV     (R2)+,TEMP2     ;READ MESSAGE ADDRESS
030420                          PRINTF  #DMFMT,TEMP3,<B,TEMP1>,TEMP2    ;PRINT DATA AND MESSAGE
030420  013746  007432                                                  MOV     TEMP2,-(SP)
030424  005046                                                          CLR     -(SP)
030426  153716  007430                                                  BISB    TEMP1,(SP)
030432  013746  007434                                                  MOV     TEMP3,-(SP)
030436  012746  030547                                                  MOV     #DMFMT,-(SP)
030442  012746  000004                                                  MOV     #4,-(SP)
030446  010600                                                          MOV     SP,R0
030450  104417                                                          TRAP    C$PNTF
030452  062706  000012                                                  ADD     #12,SP
030456  020137  007454                  CMP     R1,BEND         ;LAST ADDRESS ?
030462  101743                          BLOS    10$             ;NO,BRANCH
030464  105037  003560                  CLRB    P$NNUF          ;CLEAR ENOUGH FLAG
030470  012602                          MOV     (SP)+,R2        ;RESTORE R2
030472  012601                          MOV     (SP)+,R1        ;RESTORE R1
030474  000207                          RTS     PC              ;RETURN
030476                          .NLIST BEX
030476  047045  040445  042101  BTHEAD: .ASCIZ /%N%AADDRESS%S2%ACONTENTS%S6%ADESCRIPTION/
030547  045  022516  030523     DMFMT:  .ASCIZ /%N%S1%06%S5%03%S5%T/
        030574                  .EVEN
                                .LIST BEX
```

```
                                        .SBTTL  PRINT EVENT LOG
                                        ;PRINT THE EVENT LOG
030574  010246                  REPLOG: MOV     R2,-(SP)        ;SAVE R2
030576  010346                          MOV     R3,-(SP)        ;SAVE R3
030600  010446                          MOV     R4,-(SP)        ;SAVE R4
030602  013702  007550                  MOV     EVTPTR,R2       ;MAKE R2 A POINTER TO EVENT TABLE
030606  023727  007552  177777          CMP     EVTLOG,#-1      ;SEE IF EVENT TABLE IS EMPTY
030614  001034                          BNE     RPT0            ;BR IF NO
030616                                  PRINTS  #NULEVT         ;IF EMPTY TELL OPRERATOR.
                                                                        MOV     #NULEVT,-(SP)
                                                                        MOV     #1,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTS
                                                                        ADD     #4,SP
030632  062706  000004
030636  000137  031432                  JMP     ENDEVT          ;AND END

030642  162702  000012          RPT:    SUB     #12,R2          ;NOW POINT BACK TO TOP OF ENTRY U
                                                                ;JUST PRINTED

030646  020227  007552                  CMP     R2,#EVTLOG      ;POINTING TO TOP OF EVNT LOG QUEUE?
030652  001010                          BNE     RPT1            ; BR IF NO
030654  012702  010454                  MOV     #EVTEND,R2      ;SET R2 TO POINT TO BOTTOM OF LOG
030660  026227  177776  177777          CMP     -2(R2),#-1
030666  001007                          BNE     RPT0            ;IF END OF LOG IS NOT EMPTY
030670  000137  031432                  JMP     ENDEVT          ;CONTINUE...ELSE EXIT

030674  020237  007550          RPT1:   CMP     R2,EVTPTR       ;ARE WE BACK TO POINTER?
030700  001002                          BNE     RPT0            ;IF NOT CONTINUE
030702  000137  031432                  JMP     ENDEVT          ;IF SO EXIT....

030706  162702  000012          RPT0:   SUB     #12,R2          ;POINT R2 TO START OF ENTRY
030712                          RPTAA:  PRINTS  #EVTF0          ;PRINT EVENT ENTRY HEADER
                                                                        MOV     #EVTF0,-(SP)
                                                                        MOV     #1,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTS
                                                                        ADD     #4,SP
030712  112203                          MOVB    (R2)+,R3        ;PUT EVENT TYPE INTO R3
030714  112237  010614                  MOVB    (R2)+,EVTTCK
030720  112237  010610                  MOVB    (R2)+,EVTSEC    ;PUT EVENT TIME (TICKS,SECS,MINS IN TEMP LOC.S)
030724  112237  010612                  MOVB    (R2)+,EVTMIN
030730                                  PRINTS  #EVTF1,EVTMIN,EVTSEC,EVTTCK,EVTLST(R3)  ;PRINT EVENT TIME AND DESCRIPT.
                                                                        MOV     EVTLST(R3),-(SP)
                                                                        MOV     EVTTCK,-(SP)
                                                                        MOV     EVTSEC,-(SP)
                                                                        MOV     EVTMIN,-(SP)
                                                                        MOV     #EVTF1,-(SP)
                                                                        MOV     #5,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTS
                                                                        ADD     #14,SP
031010  000173  010624                  JMP     @RPTDSP(R3)     ;DISPATCH TO DECODING SECTION FOR SPECIFIC TYPE

031014  012237  010616          RPTTXQ: MOV     (R2)+,EVTADD    ;STORE MESSAGE ADDRESS FOR PRINTING
031020  012237  010620                  MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT FOR PRINTING
031024  012203                          MOV     (R2)+,R3        ;STORE MODEM STATUS FOR PRINTING
```

```
3853  031026
3854  031026  013746  010620                PRINTS  #EVTF2,EVTADD,EVTBCT    ;PRINT ADDR,BYTE CNT
3855  031032  013746  010616                                                        MOV     EVTBCT,-(SP)
3856  031036  012746  016447                                                        MOV     EVTADD,-(SP)
3857  031042  012746  000003                                                        MOV     #EVTF2,-(SP)
3858  031046  010600                                                                MOV     #3,-(SP)
3859  031050  104416                                                                MOV     SP,R0
3860  031052  062706  000010                                                        TRAP    C$PNTS
3861  031056  004737  031442                                                        ADD     #10,SP
3862  031062  000137  030642                JSR     PC,RPTMSB       ;GO PRINT MODEM STATUS
3863                                        JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
3864  031066  012237  010622          RPTDER: MOV     (R2)+,EVTTMP    ;GET ADDRESS OF DEVICE INFO MESSAGE
3865  031072  012237  010654                  MOV     (R2)+,DEV1      ;STORE DEVICE REG CONTENTS FOR PRINTING
3866  031076  012237  010656                  MOV     (R2)+,DEV2
3867  031102                                  PRINTS  #EVTF3,EVTTMP   ;PRINT DEVICE REG CONTENTS.
3868  031102  013746  010622                                                        MOV     EVTTMP,-(SP)
3869  031106  012746  016521                                                        MOV     #EVTF3,-(SP)
3870  031112  012746  000002                                                        MOV     #2,-(SP)
3871  031116  010600                                                                MOV     SP,R0
3872  031120  104416                                                                TRAP    C$PNTS
3873  031122  062706  000006                                                        ADD     #6,SP
3874  031126                                  PRINTS  #EVTF3C,DEV1,DEV2
3875  031126  013746  010656                                                        MOV     DEV2,-(SP)
3876  031132  013746  010654                                                        MOV     DEV1,-(SP)
3877  031136  012746  016533                                                        MOV     #EVTF3C,-(SP)
3878  031142  012746  000003                                                        MOV     #3,-(SP)
3879  031146  010600                                                                MOV     SP,R0
3880  031150  104416                                                                TRAP    C$PNTS
3881  031152  062706  000010                                                        ADD     #10,SP
3882  031156  000137  030642                JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
3883
3884  031162  005037  010654          RPTDVI: CLR     DEV1
3885  031166  005037  010656                  CLR     DEV2            ;CLEAR UPPER BYTES OF DEV1 & DEV2 BEFORE USE
3886  031172  112237  010654                  MOVB    (R2)+,DEV1      ;STORE SETUP OPERATION PARAMETERS FOR PRINTING
3887  031176  112237  010656                  MOVB    (R2)+,DEV2
3888  031202  012237  010660                  MOV     (R2)+,DEV3
3889  031206  012237  010662                  MOV     (R2)+,DEV4
3890  031212  010246                         MOV     R2,-(SP)        ;SAVE R2 ON THE STACK
3891  031214  004737  032340                 JSR     PC,SHWOP        ;GO PRINT MODE, MAINT-LOOP TYPE, PARAMTERS.
3892  031220  012602                         MOV     (SP)+,R2        ;RESTORE R2
3893  031222  000137  030642                 JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
3894                                  ;;REPORT END OF PASS OR ^C ABORT
3895  031226                          RPTABO:
3896  031226  012237  010616          RPTEOP: MOV     (R2)+,EVTADD
3897  031232  012237  010620                  MOV     (R2)+,EVTBCT
3898  031236  012237  010622                  MOV     (R2)+,EVTTMP
3899  031242                                  PRINTS  #EVTF4B,EVTADD,EVTBCT,EVTTMP    ;PRINT ADDR,RXBYTES,CMPBYTES.
3900  031242  013746  010622                                                        MOV     EVTTMP,-(SP)
3901  031246  013746  010620                                                        MOV     EVTBCT,-(SP)
3902  031252  013746  010616                                                        MOV     EVTADD,-(SP)
3903  031256  012746  016772                                                        MOV     #EVTF4B,-(SP)
3904  031262  012746  000004                                                        MOV     #4,-(SP)
3905  031266  010600                                                                MOV     SP,R0
3906  031270  104416                                                                TRAP    C$PNTS
3907  031272  062706  000012                                                        ADD     #12,SP
3908
```

```
 3909  031276  000137  030642                    JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY
 3910
 3911
 3912  031302  012237  010616          RPTDDE:  MOV     (R2)+,EVTADD    ;STORE MESSAGE ADDRESS FOR PRINTING
 3913  031306  012237  010620                   MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT FOR PRINTING
 3914  031312  012237  010622                   MOV     (R2)+,EVTTMP    ;STORE TOTAL # OF CMP ERRORS
 3915  031316                                   PRINTS  #EVTF4,EVTADD,EVTBCT,EVTTMP     ;PRINT ADDR, BYTE CNT, # CMP ERRS
 3916  031316  013746  010622                                                             MOV     EVTTMP,-(SP)
 3917  031322  013746  010620                                                             MOV     EVTBCT,-(SP)
 3918  031326  013746  010616                                                             MOV     EVTADD,-(SP)
 3919  031332  012746  016572                                                             MOV     #EVTF4,-(SP)
 3920  031336  012746  000004                                                             MOV     #4,-(SP)
 3921  031342  010600                                                                     MOV     SP,R0
 3922  031344  104416                                                                     TRAP    C$PNTS
 3923  031346  062706  000012                                                             ADD     #12,SP
 3924  031352  000137  030642                    JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY
 3925
 3926  031356                           RPTDLE:
 3927  031356  012237  010616          RPTDCK:  MOV     (R2)+,EVTADD    ;STORE MSG ADDR FOR PRINT
 3928  031362  012237  010620                   MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT
 3929  031366  012237  010622                   MOV     (R2)+,EVTTMP    ;STORE BYTE COUNT COMP
 3930  031372                                   PRINTS  #EVTF4A,EVTADD,EVTBCT,EVTTMP    ;PRINT ADDR,RXBYTES,CMPBYTES.
 3931  031372  013746  010622                                                             MOV     EVTTMP,-(SP)
 3932  031376  013746  010620                                                             MOV     EVTBCT,-(SP)
 3933  031402  013746  010616                                                             MOV     EVTADD,-(SP)
 3934  031406  012746  016674                                                             MOV     #EVTF4A,-(SP)
 3935  031412  012746  000004                                                             MOV     #4,-(SP)
 3936  031416  010600                                                                     MOV     SP,R0
 3937  031420  104416                                                                     TRAP    C$PNTS
 3938  031422  062706  000012                                                             ADD     #12,SP
 3939
 3940  031426  000137  030642                    JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY
 3941
 3942  031432  012604                   ENDEVT:  MOV     (SP)+,R4        ;RESTORE R4,R3,R2
 3943  031434  012603                            MOV     (SP)+,R3
 3944  031436  012602                            MOV     (SP)+,R2
 3945  031440  000207                            RTS     PC              ;RETURN TO CALLING ROUTINE
 3946
 3947
 3948                                   ;REPORT MODEM STATUS SUBROUTINE
 3949                                   ;       PART OF STATISICAL REPORTING (DUMPING EVENT LOG)
 3950
 3951  031442                           RPTMSB:  PRINTS  #EVMOHD                 ;PRINT MODEM STATUS HEADER
 3952  031442  012746  017160                                                             MOV     #EVMOHD,-(SP)
 3953  031446  012746  000001                                                             MOV     #1,-(SP)
 3954  031452  010600                                                                     MOV     SP,R0
 3955  031454  104416                                                                     TRAP    C$PNTS
 3956  031456  062706  000004                                                             ADD     #4,SP
 3957  031462  012704  010460                   MOV     #MOBITS,R4      ;MAKE R4 A POINTER TO MODEM SIG. BIT DEF. TABLE
 3958  031466  012705  010476                   MOV     #MOMSGS,R5      ;MAKE R5 A POINTER TO MODEM MSG. POSITION TABLE
 3959  031472  005714                   6$:     TST     (R4)            ;SEE IF BIT AVAIABLE FROM DEVICE
 3960  031474  001004                           BNE     7$              ;BR IF THAT MODEM SIG. AVAIABLE
 3961  031476  112735  000130                   MOVB    #'X,@(R5)+      ;ELSE PUT 'X" IN REPORT IF SIGNAL NOT AVAILABLE
 3962  031502  005724                           TST     (R4)+           ;BUMP R4 TO POINT TO NEXT BIT DEFINITION
 3963  031504  000407                           BR      9$              ;GO SEE IF CHECKED ALL MODEM SIGNALS
 3964  031506  032403                   7$:     BIT     (R4)+,R3        ;IF THERE, SEE IF THAT BIT IN DEVICE'S ENTRY=1
```

```
3965  031510  001403                          BEQ     8$              ;BR IF BIT (SIGNAL) VALUE =0
3966  031512  112735  000061                  MOVB    #'1,@(R5)+      ;IF=1, PUT "1" IN REPORT MESSAGE
3967  031516  000402                          BR      9$              ;GO SEE IF ALL MODEM SIGNALS CHECKED
3968  031520  112735  000060          8$:     MOVB    #'0,@(R5)+      ;IF BIT(SIGNAL)=0, PUT "0" IN REPORT MESSAGE
3969  031524  020427  010476          9$:     CMP     R4,#MOBITE      ;SEE IF ALL BITS(SIGNALS) CHECKED
3970  031530  002760                          BLT     6$              ;LOOP UNTIL ALL SIGNALS(BITS) CHECKED
3971  031532                                  PRINTS  #EVMOST         ;THEN PRINT MODEM SIGNAL VALUE MESSAGE
3972  031532  012746  017240                                                  MOV     #EVMOST,-(SP)
3973  031536  012746  000001                                                  MOV     #1,-(SP)
3974  031542  010600                                                          MOV     SP,R0
3975  031544  104416                                                          TRAP    C$PNTS
3976  031546  062706  000004                                                  ADD     #4,SP
3977  031552  000207                          RTS     PC              ;RETURN TO EVENT DECODING
3978
3979
```

```
                                .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
                                ;
                                ;--

031554  013702  007402  DUMPSR: MOV     STADD,R2        ;SET R2 UP TO STARTING ADDR.
031560  005003          DUM4:   CLR     R3              ;CLEAR R3
031562                  PRINTF  #BASM1,R2               ;PRINT ADDRESS
031562  010246                                                  MOV     R2,-(SP)
031564  012746  016257                                          MOV     #BASM1,-(SP)
031570  012746  000002                                          MOV     #2,-(SP)
031574  010600                                                  MOV     SP,R0
031576  104417                                                  TRAP    C$PNTF
031600  062706  000006                                          ADD     #6,SP
031604  005737  007406  DUM3:   TST     BYTBIT          ;IS THIS BYTE OR WORD
031610  001416          BEQ     DUM1                    ;BR IF WORD
031612  112237  007426  MOVB    (R2)+,TEMP              ;MOV BYTE TO TEMP
031616                  PRINTF  #BASM3,<B,TEMP>         ;PRINT BYTE
031616  005046                                                  CLR     -(SP)
031620  153716  007426                                          BISB    TEMP,(SP)
031624  012746  016241                                          MOV     #BASM3,-(SP)
031630  012746  000002                                          MOV     #2,-(SP)
031634  010600                                                  MOV     SP,R0
031636  104417                                                  TRAP    C$PNTF
031640  062706  000006                                          ADD     #6,SP
031644  000411          BR      DUM2
031646          DUM1:   PRINTF  #BASM2,(R2)+            ;PRINT WORD
031646  012246                                                  MOV     (R2)+,-(SP)
031650  012746  016250                                          MOV     #BASM2,-(SP)
031654  012746  000002                                          MOV     #2,-(SP)
031660  010600                                                  MOV     SP,R0
031662  104417                                                  TRAP    C$PNTF
031664  062706  000006                                          ADD     #6,SP
031670  020237  007404  DUM2:   CMP     R2,ENADD        ;COMPARE FOR LAST ADD
```

```
 4036  031674  003005                        BGT     DUMEX           ;IF DONE EXIT
 4037  031676  005203                        INC     R3              ;ELSE BUMP R3
 4038  031700  022703  000010                CMP     #8.,R3          ;HAVE WE PRINTED 8 ACCROSS
 4039  031704  001725                        BEQ     DUM4            ;IF SO GO BACK TO 4
 4040  031706  000736                        BR      DUM3            ;ELSE GO BACK AND PRINT ANOTHER
 4041                                                                ;BYTE OR WORD
 4042  031710  000207              DUMEX:    RTS     PC              ;RETURN TO CALLER
 4043
```

```
                                .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
                                ;--

031712  063737  007412  007422 ADDCC:  ADD     CURCC,TOTCC     ;ADD CURRENT TO TOTAL
031720  022737  001000  007422          CMP     #BUFLIM,TOTCC   ; COMPARE TO 'BUFLIM'
031726  103027                       BHIS    ADDC1           ;IF NOT MORE THEN 'BUFLIM' EXIT

                                ; PRINT MESSAGE AND TRUNCATE COUNT

031730                          PRINTF  #MSGTRU
031730  012746  015276                                                  MOV     #MSGTRU,-(SP)
031734  012746  000001                                                  MOV     #1,-(SP)
031740  010600                                                          MOV     SP,R0
031742  104417                                                          TRAP    C$PNTF
031744  062706  000004                                                  ADD     #4,SP
031750  163737  007412  007422          SUB     CURCC,TOTCC     ;SUB CURRENT FROM TOTAL
031756  012737  001000  007426          MOV     #BUFLIM,TEMP    ;MOV 'BUFLIM' TO TEMP
031764  163737  007422  007426          SUB     TOTCC,TEMP      ;SUB TOTAL FROM 'BUFLIM'
031772  013737  007426  007412          MOV     TEMP,CURCC      ;AND ESTABLISH NEW CURRENT
032000  063737  007412  007422          ADD     CURCC,TOTCC     ;ADD 'ADJUSTED CURRENT' TO TOTAL CHAR. CNT.
032006  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.
                                ;--

032010                          BLDBUF:
032010  010246                          MOV     R2,-(SP)        ;SAVE R2 AND R3 ON THE STACK
032012  010346                          MOV     R3,-(SP)
032014  013702  007416                  MOV     CPTR,R2

032020  013722  007420          BLDB1:  MOV     CURADD,(R2)+    ;PUT CURRENT ADD ON POINTER TAB
032024  013722  007412                  MOV     CURCC,(R2)+     ;PUT CURRENT CC ON POINTER TAB
032030  010237  007416                  MOV     R2,CPTR         ;PUT UPDATED R2 BACK TO CURRENT POINT
032034  013702  007410                  MOV     MSGTYP,R2       ;GET MESSAGE TYPE TO USE AS INDEX
032040  006302                          ASL     R2              ;DOUBLE FOR WORD INDEX
032042  013737  007420  007426          MOV     CURADD,TEMP     ;MOVE CURRENT ADD TO TEMP
032050  063737  007412  007426          ADD     CURCC,TEMP      ;ADD CHAR COUNT TO IT TO GET END
032056  013703  007420                  MOV     CURADD,R3       ;SET R3 TO CURRENT START ADD
032062  016237  002150  007432  BLDB2:  MOV     DMSGCT(R2),TEMP2        ;GET BYTE COUNT
032070  016204  002176                  MOV     DMSGAD(R2),R4   ;PUT STARTING FROM ADD IN R4
032074  060437  007432                  ADD     R4,TEMP2        ;ADD IT TO TEMP2 TO GET END OF FROM
032100  112423                  BLDB3:  MOVB    (R4)+,(R3)+     ;MOV BYTE FROM PATTERN TO BUFFER
032102  020337  007426                  CMP     R3,TEMP         ;ALL DONE?
032106  001404                          BEQ     BLDBEX          ;IF SO EXIT
032110  020437  007432                  CMP     R4,TEMP2        ;IS PATTERN COUNT EXPIRED
032114  001762                          BEQ     BLDB2           ;IF SO GO START AGAIN
032116  000770                          BR      BLDB3           ;IF NOT GET ANOTHER BYTE
032120  063737  007412  007420  BLDBEX: ADD     CURCC,CURADD    ;BUMP CURADD
032126  012603                          MOV     (SP)+,R3        ;RESTORE R3 AND R2
032130  012602                          MOV     (SP)+,R2
032132  000207                          RTS     PC              ;RETURN TO CALLER
```

```
                        .SBTTL  CREATE FACSIMILE OF TX BUFFER AND MESSAGE LIST

                        ;++
                        ;       THIS ROUTINE ADDED FOR REV B BY EC
                        ;FUNCTIONAL DESCRIPTION:
                        ;       FACSIMILE: THIS ROUTINE IS USED TO CREATE A FACSIMILE OF THE
                        ;                  OF THE TRANSMIT LIST AND TRANSMIT BUFFER IN THE
                        ;                  EXPECTED LIST AND EXPECTED BUFFER. THE ROUTINE IS
                        ;                  NORMALLY CALLED WHEN USER COMMAND ''SET E [XPECT]=
                        ;                  T [RANSMIT] IS ENTERED.
                        ;
                        ;       CALLING SEQUENCE: JSR   PC,FACSIMILE
                        ;
                        ;--
                        ;       DEFINITIONS  CMPBUF = EXPECTED DATA BUFFER  HOLDS MAX 512 BYTES
                        ;                    TXBUF  = TRANSMIT DATA BUFFER  HOLDS MAX 512 BYTES
                        ;                    TTOTCC = NUMBER OF BYTES IN TXBUF
                        ;                    PTRTAB = TOP OF MESSAGE LIST POINTER TABLE
                        ;                    CTOTCC = NUMBER OF BYTES IN EXPECT MESSAGE
                        ;                    CMPTOT = NUMBER OF EXPECTED MESSAGES
                        ;                    CMPPTR = EXPECTED MESSAGE LIST POINTER
                        ;                    TXPTR  = TRANSMIT MESSAGE LIST POINTER
                        ;                    TXMTOT = NUMBER OF TRANSMIT MESSAGES
                        ;                    CCURAD = STORAGE ADDRESS OF MESSAGE IN CMPBUF
                        ;                    MSGLIN = MAXIMUM NUMBER OF MESSAGES THAT CAN BE STORED
                        ;
                        ;       BEGIN FACSIMILE ROUTINE
                        ;       (*COPY TXBUF ==> CMPBUF*)
                        ;       ..SAVE R1
                        ;       ..INIT R1
                        ;       ..REPEAT
                        ;       ....[CMPBUF]R1=[TXBUF]R1
                        ;       ....R1=R1+1
                        ;       ..UNTIL R1 = BUFLIM
                        ;
                        ;       (*NOW CALCULATE EXPECT LIST MESSAGE POINTER*)
                        ;       ..CMPPTR = PTRTAB + (2 * MSGLIM)
                        ;
                        ;       (*NOW PRIME THE WHILE - DO LOOP*)
                        ;       ..TXPTR = PTRTAB
                        ;       ..CCURAD = CMPBUF
                        ;       ..TXPTR = TXPTR + 2
                        ;       ..CTOTCC = [TXPTR]
                        ;       ..CMPTOT = 0
                        ;       ..WHILE TXMTOT <> CMPTOT DO
                        ;       ....[CMPPTR] = CCURAD
                        ;       ....CMPPTR = CMPPRT + 2
                        ;       ....[CMPPTR] = CTOTCC
                        ;       ....TXPTR = TXPTR + 4
                        ;       ....CCURAD = CCURAD + CTOTCC
                        ;       ....CTOTCC = [TXPTR]
                        ;       ....CMPPTR = CMPPTR + 2
                        ;       ....CMPTOT = CMPTOT + 1
                        ;       ..END WHILE DO
                        ;       ..CTOTCC = TTOTCC
                        ;       END FACSIMILE ROUTINE
```

```
 4193
 4194  032134                                 FACSIMILE:
 4195  032134  010146                                 MOV     R1,-(SP)                ;SAVE R1
 4196  032136  005001                                 CLR     R1                      ;INIT R1
 4197  032140  116161  003562  005562  10$:          MOVB    TXBUF(R1),CMPBUF(R1)    ;COPY TX BUFFER TO EXPECTED BUFFER
 4198  032146  005201                                 INC     R1                      ;BUMP INDEX
 4199  032150  020127  001000                        CMP     R1,#BUFLIM              ;ALL DATA COPIED ?
 4200  032154  001371                                 BNE     10$                     ;NO,BRANCH
 4201
 4202  032156  012701  000017            20$:        MOV     #MSGLIM,R1              ;MESSAGE LIMIT
 4203  032162  006301                                 ASL     R1                      ;MULTIPLY BY 2
 4204  032164  006301                                 ASL     R1                      ;MULTIPLY BY 2
 4205  032166  012737  006562  007336                MOV     #PTRTAB,CMPPTR          ;TOP OF POINTER TABLE
 4206  032174  060137  007336                        ADD     R1,CMPPTR               ;START OF EXPECTED POINTER TABLE
 4207  032200  005001                                 CLR     R1                      ;INIT R1
 4208
 4209                                          ;SET UP WHILE - DO LOOP
 4210  032202  012737  006562  007334                MOV     #PTRTAB,TXPTR           ;TX POINTER NOW AT TOP OF TABLE
 4211  032210  012737  005562  007344                MOV     #CMPBUF,CCURAD          ;TRANSFER ADDRESS OF 1ST MESSAGE
 4212  032216  062737  000002  007334                ADD     #2,TXPTR                ;BUMP POINTER
 4213  032224  017737  155104  007342                MOV     @TXPTR,CTOTCC           ;BYTE COUNTER 1ST MESSAGE
 4214  032232  005037  007340                        CLR     CMPTOT                  ;INIT EXPECTED MESSAGE COUNT
 4215
 4216                                          ;WHILE TX MESSAGE TOTAL <> EXPECTED MESSAGE TOTAL DO
 4217  032236  023737  007354  007340  30$:         CMP     TXMTOT,CMPTOT           ;ALL MESSAGES COPIED ?
 4218  032244  001430                                 BEQ     40$                     ;YES,BRANCH
 4219  032246  013777  007344  155062                MOV     CCURAD,@CMPPTR          ;TRANSFER ADDRESS OF MESSAGE
 4220  032254  062737  000002  007336                ADD     #2,CMPPTR               ;BUMP POINTER
 4221  032262  013777  007342  155046                MOV     CTOTCC,@CMPPTR          ;BYTE COUNT OF MESSAGE
 4222  032270  062737  000004  007334                ADD     #4,TXPTR                ;BUMP TX MESSAGE POINTER
 4223  032276  063737  007342  007344                ADD     CTOTCC,CCURAD           ;CALC. TRANSFER ADDRESS
 4224  032304  017737  155024  007342                MOV     @TXPTR,CTOTCC           ;BYTE COUNT NEXT MESSAGE
 4225  032312  062737  000002  007336                ADD     #2,CMPPTR               ;BUMP POINTER
 4226  032320  005237  007340                        INC     CMPTOT                  ;INCREMENT MESSAGE COUNT
 4227  032324  000744                                 BR      30$                     ;DO IT AGAIN
 4228                                          ;END WHILE - DO
 4229  032326  013737  007356  007342  40$:         MOV     TTOTCC,CTOTCC           ;COPY TOTAL CHARACTER COUNT
 4230
 4231                                          ;END ROUTINE
 4232  032334  012601                          MOV     (SP)+,R1                ;RESTORE R1
 4233  032336  000207                          RTS     PC                      ;RETURN
 4234
 4235
```

```
                                .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
                                ;--

032340  013702  010654          SHWOP:  MOV     DEV1,R2         ;GET THE MODE TYPE IN R2
032344  006302                          ASL     R2              ;MAKE IT A WORD TABLE OFFSET
032346  016237  003514  007426          MOV     MODES(R2),TEMP  ;GET ADDRESS OF MODE-IN-ASCII
032354  013702  010656                  MOV     DEV2,R2         ;GET MAINTENANCE LOOP TYPE
032360  006302                          ASL     R2
032362  012737  014642  007434          MOV     #LP00,TEMP3     ;LOAD TEMP3 TO POINT TO ''/LOOP=''
032370  005702                          TST     R2              ;SEE IF /LOOP=XXXXX OR NONE
032372  001003                          BNE     10$             ; BR IF /LOOP= OF SOME KIND
032374  012737  014641  007434          MOV     #LP0,TEMP3      ;IF NO LOOP THEN DON'T PRINT ''/LOOP=''
032402  016237  003532  007430  10$:    MOV     LOOPS(R2),TEMP1 ;GET ADDRESS OF LOOP-IN-ASCII
032410  013737  010660  007432          MOV     DEV3,TEMP2      ;GET NUMBER OF PASSES
032416                                  PRINTS  #SHF0,TEMP,TEMP3,TEMP1,TEMP2
032416  013746  007432                                                  MOV     TEMP2,-(SP)
032422  013746  007430                                                  MOV     TEMP1,-(SP)
032426  013746  007434                                                  MOV     TEMP3,-(SP)
032432  013746  007426                                                  MOV     TEMP,-(SP)
032436  012746  015361                                                  MOV     #SHF0,-(SP)
032442  012746  000005                                                  MOV     #5,-(SP)
032446  010600                                                          MOV     SP,R0
032450  104416                                                          TRAP    C$PNTS
032452  062706  000014                                                  ADD     #14,SP

032456  005002                          CLR     R2              ;NOW SET UP FOR QUALIFIERS IN ASCII
032460  012737  014721  007426          MOV     #PST,TEMP
032466  032737  000001  010662          BIT     #STATB,DEV4     ;SEE IF /STATUS OR /NOSTATUS
032474  001003                          BNE     1$              ;BR IF /STATUS
032476  012737  014717  007426          MOV     #PNST,TEMP
032504  012737  014732  007430  1$:     MOV     #PCK,TEMP1
032512  032737  000002  010662          BIT     #DATCKB,DEV4    ;SEE IF /CHECK OR /NOCHECK
032520  001003                          BNE     2$              ;BR IF /CHECK
032522  012737  014730  007430          MOV     #PNCK,TEMP1
032530  012737  014742  007432  2$:     MOV     #PEC,TEMP2
032536  032737  000004  010662          BIT     #ECHOB,DEV4     ;SEE IF /ECHO OR /NOECHO
032544  001003                          BNE     3$              ;BR IF /ECHO
032546  012737  014740  007432          MOV     #PNEC,TEMP2
```

```
 4292  032554  012737  014751  007440  3$:     MOV     #PMS,TEMP5      ;ASSUME /MODEM ;REV B EC
 4293  032562  032737  000010  010662          BIT     #MOCHK,DEV4     ;MODEM CHECK ? ;REV B EC
 4294  032570  001003                          BNE     5$              ;YES,BRANCH ;REV B EC
 4295  032572  012737  014747  007440          MOV     #PNMS,TEMP5     ;"/NOMODEM" MESSAGE ;REV B EC
 4296
 4297
 4298  032600                          5$:     PRINTS  #SHF1,TEMP,TEMP1,TEMP2,TEMP5    ;,TEMP3,TEMP4 **;SEE NOTE ABOVE
 4299  032600  013746  007440                                                                  MOV     TEMP5,-(SP)
 4300  032604  013746  007432                                                                  MOV     TEMP2,-(SP)
 4301  032610  013746  007430                                                                  MOV     TEMP1,-(SP)
 4302  032614  013746  007426                                                                  MOV     TEMP,-(SP)
 4303  032620  012746  015417                                                                  MOV     #SHF1,-(SP)
 4304  032624  012746  000005                                                                  MOV     #5,-(SP)
 4305  032630  010600                                                                          MOV     SP,R0
 4306  032632  104416                                                                          TRAP    C$PNTS
 4307  032634  062706  000014                                                                  ADD     #14,SP
 4308  032640  000207                          RTS     PC              ;RETURN
 4309
 4310
```

```
                                .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
                                ;--

032642                          P$TRV:
032642  013704  003544                  MOV     P$BUFA,R4
032646  013703  003546                  MOV     P$TREE,R3
032652  105714                  P$TR5:  TSTB    (R4)                    ;SEE IF ANY CHARS LEFT IN INPUT STRING
032654  001441                          BEQ     P$EXIT                  ;BR IF NO
032656  121327  000013                  CMPB    (R3),#11.               ;SEE IF SPECIAL CLI CHAR CODE OR ASCII
032662  003023                          BGT     20$                     ;BR IF REGULAR ASCII CHAR.
032664  111305                          MOVB    (R3),R5                 ;GET SPECIAL CHAR CODE INTO R5
032666  006305                          ASL     R5
032670  016505  032704                  MOV     10$(R5),R5              ;BUILD TRAVERSE ROUTINE ADDRESS
032674  062705  032704                  ADD     #10$,R5
032700  004715                          JSR     PC,(R5)                 ;JSR TO SPECIAL CLI TRAVERSE ROUTINE
032702  000763                          BR      P$TR5                   ;GO SEE IF MORE OF STRING LEFT

                                        ;TRAVERSE TABLE FOR 'CLI FUNTIONS'
032704  000114                  10$:    .WORD   TRVERR-10$              ;TAKE ERROR ACTION
032706  000134                          .WORD   TRVEXI-10$              ;TAKE EXIT ACTION
032710  000152                          .WORD   TRVBR-10$               ;TAKE BRANCH ACTION
032712  000162                          .WORD   TRVBIF-10$              ;TEST P$GDBD & TAKE BRANCH
032714  000204                          .WORD   TRVSPA-10$              ;SKIP SPACES OR TABS IN CMD LINE
032716  000270                          .WORD   TRVNUM-10$              ;TRAVERSE NUMERIC FIELD
032720  000604                          .WORD   TRVALP-10$              ;TRAVERSE ALPHABETICS
032722  000650                          .WORD   TRVALN-10$              ;TRAVERSE ALPHANUMERICS
032724  000270                          .WORD   TRVOCT-10$              ;SAME AS TRVNUM
032726  000256                          .WORD   TRVDEC-10$              ;SAME AS CLINUM BUT DECIMAL
032730  000736                          .WORD   TRVSTR-10$              ;FIND ASCIZ MATCH IN CMD LINE

                                ;NOT A SPECIAL CODE

032732  121314                  20$:    CMPB    (R3),(R4)               ;SEE IF FIRST CHAR OF STRING IS A MATCH
032734  001403                          BEQ     22$                     ;BR IF A MATCH
032736  004737  033002                  JSR     PC,TRVBRC               ;IF NOT A MATCH, GO TAKE MISS BRANCH
032742  000743                          BR      P$TR5                   ; THEN GO BACK PT'G TO MISS NODE
032744  004737  032762          22$:    JSR     PC,TRVACT               ;IF A MATCH, GO DO ACTION DEFINED BY
032750  062703  000004                  ADD     #4,R3                   ; ACTION CODE IN CLI NODE, THEN
                                                                        ; ADJUST PTR TO NEXT CLI NODE
032754  005204                          INC     R4                      ;ADJUST BUF PTR TO NEXT CHAR IF MATCH
032756  000735                          BR      P$TR5
```

```
 4367
 4368  032760  000207                P$EXIT: RTS     PC                      ;RETURN FROM PARSER
 4369
 4370                                ;-----------------------------------------------------------------------
 4371
 4372                                ;GOTO USER ACTION ROUTINE
 4373  032762  116302  000001        TRVACT: MOVB    1(R3),R2                ;GET ACTION CODE FROM CLI NODE
 4374  032766  042702  177400                BIC     #177400,R2              ;CLEAR ANY SIGN EXTENSION
 4375  032772  013705  003550                MOV     P$ACT,R5                ;GET ADDRESS OF CLI ACTION ROUTINE
 4376  032776  004715                       JSR     PC,(R5)                 ;GO DO ACTION DEFINED BY CODE
 4377  033000  000207                       RTS     PC                      ;RETURN TO CALLING CODE
 4378
 4379                                ;TAKE BRANCH IN TREE
 4380  033002  016305  000002        TRVBRC: MOV     2(R3),R5                ;GET BRANCH DISPLACEMENT FROM TREE
 4381  033006  060503                       ADD     R5,R3                   ; AND POINT R3 TO THE "MISS" NODE
 4382  033010  000207                       RTS     PC                      ; RETURN TO P$TRV
 4383
 4384                                ;NO BRANCH TAKEN
 4385  033012  062703  000004        TRVNOB: ADD     #4,R3                   ;THINGS OK, UPDATE R3 TO POINT TO NEXT
 4386  033016  000207                       RTS     PC                      ; NODE AND RETURN TO P$TRV
 4387
 4388                                ;-----------------------------------------------------------------------
 4389  033020  004737  032762        TRVERR: JSR     PC,TRVACT               ;TAKE ERROR ACTION
 4390  033024  112737  177777  003561        MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
 4391  033032  005726                       TST     (SP)+                   ;GET RID OF "JSR PUSH TO TRVERR"
 4392  033034  000137  032760                JMP     P$EXIT                  ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
 4393
 4394  033040  004737  032762        TRVEXI: JSR     PC,TRVACT               ;TAKE EXIT ACTION
 4395  033044  105037  003561                CLRB    P$GDBD                  ;SET GOOD/BAD FLAG TO "SUCCESS (0)"
 4396  033050  005726                       TST     (SP)+                   ;GET RID OF "JSR PUSH TO TRVEXI"
 4397  033052  000137  032760                JMP     P$EXIT                  ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
 4398
 4399  033056  004737  032762        TRVBR:  JSR     PC,TRVACT               ;GO TAKE BRANCH ACTION
 4400  033062  000137  033002                JMP     TRVBRC
 4401
 4402  033066  004737  032762        TRVBIF: JSR     PC,TRVACT
 4403  033072  105737  003561                TSTB    P$GDBD                  ;SEE IF P$GDBD SET OR CLEARED BY ACTION
 4404  033076  001402                       BEQ     1$                      ;IF CLEAR FALL THRU TO NEXT NODE
 4405  033100  000137  033002                JMP     TRVBRC                  ;ELSE TAKE THE "MISS" BRANCH
 4406  033104  000137  033012        1$:     JMP     TRVNOB                  ;JUST UPDATE TO NEXT NODE IF THINGS OK
 4407
 4408  033110  005005                TRVSPA: CLR     R5                      ;CLEAR "SPACE OR TAB FOUND" FLAG
 4409  033112  121427  000011        1$:     CMPB    (R4),#11                ;SEE IF CHAR. IN CMD LINE= TAB
 4410  033116  001003                       BNE     2$                      ;BR IF NO, NOT A TAB
 4411  033120  005204                       INC     R4                      ;INC INPUT STRING POINTER
 4412  033122  005205                       INC     R5                      ;INDICATE A TAB FOUND
 4413  033124  000772                       BR      1$                      ;GO CHECK NEXT CHAR
 4414
 4415  033126  121427  000040        2$:     CMPB    (R4),#40                ;SEE IF CHAR. IN CMD LINE= SPACE
 4416  033132  001003                       BNE     10$                     ;BR IF NO, NON-SPACE OR NON-TAB CHAR.
 4417  033134  005204                       INC     R4                      ;INC INPUT STRING POINTER
 4418  033136  005205                       INC     R5                      ;INDICATE A SPACE FOUND
 4419  033140  000764                       BR      1$                      ;GO CHECK NEXT CHAR
 4420  033142  005705                10$:    TST     R5                      ;SEE IF ANY SPACES OR TABS FOUND
 4421  033144  001404                       BEQ     15$                     ;BR IF NO, TAKE NO ACTION
 4422  033146  004737  032762                JSR     PC,TRVACT               ;GO TAKE ACTION IF ANY FOUND
```

```
 4423  033152  000137  033012                  JMP     TRVNOB                  ;JUST GO UPDATE R3 TO NEXT NODE IF OK
 4424  033156  000137  033002          15$:    JMP     TRVBRC                  ;TAKE BRANCH (MISS) IF NONE FOUND
 4425
 4426
 4427  033162  012737  000012  003556  TRVDEC: MOV     #10.,P$RADX             ;USE DECIMAL AS RADIX AND ASSUME +
 4428  033170  000137  033202          TRVOCT: JMP     TRVNMA
 4429  033174                                  ;(SAME AS TRVNUM SINCE DEFAULT RADIX IS OCTAL)
 4430  033174  012737  000010  003556  TRVNUM: MOV     #8.,P$RADX              ;USE OCTAL AS RADIX AND ASSUME +
 4431  033202  005005                  TRVNMA: CLR     R5                      ;CLEAR DIGIT COUNTER
 4432  033204  121427  000053                  CMPB    (R4),#'+                ;SEE IF THERE'S A + SIGN THERE
 4433  033210  001001                          BNE     10$                     ; BR IF NO
 4434  033212  000406                          BR      11$                     ; ELSE P$RADX ALREADY SAYS +, JUST BR
 4435  033214  121427  000055          10$:    CMPB    (R4),#'-                ;SEE IF THERE'S A - SIGN THERE
 4436  033220  001004                          BNE     1$                      ; BR IF NO
 4437  033222  112737  177777  003557          MOVB    #-1,P$RADX+1            ;SET 'MINUS FLAG'' (HI BYTE OF P$RADX)
 4438  033230  005204                  11$:    INC     R4                      ;BUMP R4 TO POINT TO FIRST CHAR
 4439
 4440  033232  121427  000060          1$:     CMPB    (R4),#60                ;SEE IF CHAR. LESS THAN A ''0''
 4441  033236  002434                          BLT     2$                      ;BR IF YES (NOT NUMERIC)
 4442  033240  121427  000067                  CMPB    (R4),#67                ;SEE IF CHAR. GREATER THAN A ''7''
 4443  033244  003426                          BLE     13$                     ; BR IF YES
 4444  033246  123727  003556  000012          CMPB    P$RADX,#10.             ;SEE IF IN DECIMAL MODE
 4445  033254  001417                          BEQ     12$                     ; BR IF YES (CAN USE HIGHER LIMIT)
 4446  033256  121427  000071                  CMPB    (R4),#71                ;SEE IF DIGIT WAS A 8 OR 9
 4447  033262  003022                          BGT     2$                      ;BR IF NON-NUMERIC
 4448  033264                                  PRINTF  #CLIBRX                 ;ELSE WAS A 8 OR 9 WHEN IN OCTAL RADIX
 4449  033264  012746  012613                                                          MOV     #CLIBRX,-(SP)
 4450  033270  012746  000001                                                          MOV     #1,-(SP)
 4451  033274  010600                                                                  MOV     SP,R0
 4452  033276  104417                                                                  TRAP    C$PNTF
 4453  033300  062706  000004                                                          ADD     #4,SP
 4454  033304  112737  177777  003561          MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
 4455  033312  000474                          BR      5$                      ; PRINT ERROR AND TAKE MISS
 4456
 4457  033314  121427  000071          12$:    CMPB    (R4),#71                ;SEE IF CHAR. GREATER THAN A ''9''
 4458  033320  003003                          BGT     2$                      ;BR IF YES (NOT NUMERIC)
 4459  033322  005204                  13$:    INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR.
 4460  033324  005205                          INC     R5                      ;INDICATE A NUMERIC FOUND
 4461  033326  000741                          BR      1$                      ;GO LOOK AT NEXT CHAR.
 4462
 4463  033330  005705                  2$:     TST     R5                      ;SEE IF FOUND ANY NUMERICS
 4464  033332  001464                          BEQ     5$                      ;BR IF NO, TAKE ''MISS'' BRANCH
 4465  033334  010401                          MOV     R4,R1                   ;GET POINTER TO START OF NUMERIC STRING
 4466  033336  160501                          SUB     R5,R1
 4467  033340  005037  003554                  CLR     P$NUM                   ;CLEAR LOC. WHERE VALUE WILL BE STORED
 4468  033344  112102                  3$:     MOVB    (R1)+,R2                ;GET ASCII CHAR AND CONVERT IT TO A #
 4469  033346  162702  000060                  SUB     #60,R2
 4470  033352  006337  003554                  ASL     P$NUM                   ;SHIFT CURRENT VALUE TO MAKE ROOM
 4471  033356  103437                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
 4472  033360  013737  003554  003552          MOV     P$NUM,P$CNT             ;SAVE FOR LATER IN CASE DECIMAL RADIX
 4473  033366  006337  003554                  ASL     P$NUM
 4474  033372  103431                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
 4475  033374  006337  003554                  ASL     P$NUM
 4476  033400  103426                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
 4477  033402  123727  003556  000012          CMPB    P$RADX,#10.             ;SEE IF DECIMAL RADIX
 4478  033410  001004                          BNE     4$                      ;BR IF NOT EQUAL
```

```
 4479  033412  063737  003552  003554          ADD     P$CNT,P$NUM
 4480  033420  103416                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
 4481  033422  060237  003554          4$:     ADD     R2,P$NUM
 4482  033426  103413                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
 4483  033430  005305                          DEC     R5
 4484  033432  001344                          BNE     3$
 4485  033434  105737  003557                  TSTB    P$RADX+1                ;SEE IF NUM WAS PRECEDED BY A - SIGN
 4486  033440  001402                          BEQ     15$                     ; BR IF NO
 4487  033442  005437  003554                  NEG     P$NUM                   ; ELSE NEGATE THE NUMBER BEFORE LEAVING
 4488  033446  004737  032762          15$:    JSR     PC,TRVACT               ;SINCE NUMERIC FOUND, GO TAKE ACTION
 4489  033452  000137  033012                  JMP     TRVNOB                  ;GO POINT R3 TO NEXT NODE
 4490
 4491  033456                          7$:     PRINTF  #CLINBG                 ;PRINT NUMBER TOO BIG ERROR
 4492  033456  012746  012571                                                          MOV     #CLINBG,-(SP)
 4493  033462  012746  000001                                                          MOV     #1,-(SP)
 4494  033466  010600                                                                  MOV     SP,R0
 4495  033470  104417                                                                  TRAP    C$PNTF
 4496  033472  062706  000004                                                          ADD     #4,SP
 4497  033476  112737  177777  003561          MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
 4498  033504  000137  033002          5$:     JMP     TRVBRC                  ;TAKE "MISS" BRANCH
 4499
 4500
 4501  033510  005005                  TRVALP: CLR     R5                      ;CLEAR ALPHA FOUND FLAG
 4502  033512  121427  000101          1$:     CMPB    (R4),#101               ;SEE IF CHAR. LESS THAN A "A"
 4503  033516  002406                          BLT     2$                      ;BR IF YES (NOT ALPHA)
 4504  033520  121427  000132                  CMPB    (R4),#132               ;SEE IF CHAR. GREATER THAN A "Z"
 4505  033524  003003                          BGT     2$                      ;BR IF YES (NOT ALPHA)
 4506  033526  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR
 4507  033530  005205                          INC     R5                      ;INDICATE AN ALPHA WAS FOUND
 4508  033532  000767                          BR      1$                      ;GO LOOK AT NEXT CHAR.
 4509  033534  005705                  2$:     TST     R5                      ;SEE IF ANY ALPHA'S WERE FOUND
 4510  033536  001404                          BEQ     3$                      ;BR IF NO
 4511  033540  004737  032762                  JSR     PC,TRVACT               ;IF ANY FOUND TAKE ACTION
 4512  033544  000137  033012                  JMP     TRVNOB                  ;THEN UPDATE R3 TO NEXT NODE -NO BRANCH
 4513  033550  000137  033002          3$:     JMP     TRVBRC                  ;NONE FOUND, TAKE MISS BRANCH
 4514
 4515  033554  005005                  TRVALN: CLR     R5                      ;CLEAR ALPHANUM FOUND FLAG
 4516  033556  121427  000060          10$:    CMPB    (R4),#60                ;SEE IF CHAR. LESS THAN A "0"
 4517  033562  002417                          BLT     2$                      ;BR IF YES (NOT NUMERIC OR ALPHA)
 4518  033564  121427  000072                  CMPB    (R4),#72                ;SEE IF CHAR. GREATER THAN A "9"
 4519  033570  003003                          BGT     1$                      ;BR IF YES (NOT NUMERIC)
 4520  033572  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR.
 4521  033574  005205                          INC     R5                      ;INDICATE A NUMERIC FOUND
 4522  033576  000767                          BR      10$                     ;GO LOOK AT NEXT CHAR.
 4523  033600  121427  000101          1$:     CMPB    (R4),#101               ;SEE IF CHAR. LESS THAN A "A"
 4524  033604  002406                          BLT     2$                      ;BR IF YES (NOT ALPHA)
 4525  033606  121427  000132                  CMPB    (R4),#132               ;SEE IF CHAR. GREATER THAN A "Z"
 4526  033612  003003                          BGT     2$                      ;BR IF YES (NOT ALPHA)
 4527  033614  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR
 4528  033616  005205                          INC     R5                      ;INDICATE AN ALPHA FOUND
 4529  033620  000756                          BR      10$                     ;GO LOOK AT NEXT CHAR.
 4530  033622  005705                  2$:     TST     R5                      ;SEE IF ANY ALPHANUM'S WERE FOUND
 4531  033624  001404                          BEQ     3$                      ;BR IF NO
 4532  033626  004737  032762                  JSR     PC,TRVACT               ;IF ANY FOUND TAKE ACTION
 4533  033632  000137  033012                  JMP     TRVNOB                  ;THEN UPDATE R3 TO NEXT NODE -NO BRANCH
 4534  033636  000137  033002          3$:     JMP     TRVBRC                  ;NONE FOUND, TAKE MISS BRANCH
```

```
 4535
 4536
 4537
 4538  033642  010401                TRVSTR: MOV     R4,R1                   ;POINT R1 TO CMD STRING
 4539  033644  010305                        MOV     R3,R5
 4540  033646  062705  000006                ADD     #6,R5                   ;POUNT R5 TO MATCH STRING FROM CLI NODE
 4541  033652  005037  003552                CLR     P$CNT                   ;CLEAR CHAR MATCH COUNT
 4542  033656  105715                2$:     TSTB    (R5)                    ;SEE IF END OF MATCH STRING YET
 4543  033660  001411                        BEQ     10$                     ;BR IF YES
 4544  033662  105711                        TSTB    (R1)                    ;SEE IF END OF CMD LINE YET
 4545  033664  001407                        BEQ     10$                     ;BR IF YES
 4546  033666  121115                        CMPB    (R1),(R5)               ;SEE IF CHARACTERS MATCH
 4547  033670  001005                        BNE     10$                     ;BR IF NO
 4548  033672  005237  003552                INC     P$CNT                   ;MATCH -INCREMENT MATCH COUNT
 4549  033676  005201                        INC     R1                      ;UPDATE STRING POINTERS
 4550  033700  005205                        INC     R5
 4551  033702  000765                        BR      2$                      ;BR TO CONTINUE CHECKING CHARS.
 4552
 4553  033704  005737  003552        10$:    TST     P$CNT                   ;WHEN DONE SEE IF ANY MATCHES FOUND
 4554  033710  001406                        BEQ     15$                     ;BR IF NO, GO TAKE THE MISS BRANCH
 4555  033712  010104                        MOV     R1,R4                   ;POINT CMD POINTER TO END OF STRING &
 4556  033714  004737  032762                JSR     PC,TRVACT               ;IF A MATCH FOUND, GO DO MATCH ACTION
 4557  033720  066303  000004                ADD     4(R3),R3                ;UPDATE R3 TO NEXT NODE (NO BRANCH)
 4558  033724  000207                        RTS     PC                      ; (NO RETURN THRU TRVNOB SINCE DIFFERNT
 4559                                                                        ;  DISPLACEMENT DUE TO MATCH STRING)
 4560  033726  000137  033002        15$:    JMP     TRVBRC                  ; GO TAKE BRANCH
 4561
 4562                                                                        ; (PARSED OK), -1 IF ILL CMD.....
 4563                                ;-----------------------------------------------------------------------
 4564
```

```
                                .SBTTL  REPORT CODING SECTION


                                ;++
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;--

033732                                  BGNRPT
033732                                                                  L$RPT::


033732  004737  027046                  JSR     PC,REPORT               ;CALL SUBROUTINE TO DUMP EVENT LOG
                                                                        ; AND BASE TABLE


033736                                  ENDRPT
033736                                                                  L10011:
033736  104425                                                                  TRAP    C$RPT
```

```
                                .SBTTL  PROTECTION TABLE

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

033740                                  BGNPROT
033740                                                                  L$PROT::

033740  177777                         -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
033742  177777                         -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
033744  177777                         -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

033746                                  ENDPROT
```

```
                                .SBTTL  INITIALIZE SECTION

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

033746                                  BGNINIT
033746                                                                  L$INIT::

033746  005037  003202                  CLR     KEYWD1                  ;INIT USER COMMAND VARIABLE
033752  005737  007464                  TST     DCLFLG                  ;CLEANUP & EXIT ?
033756  001403                          BEQ     INIT1                   ;NO,BRANCH
033760  005037  007464                  CLR     DCLFLG                  ;CLEAR FLAG
033764                                  DOCLN                           ;GO CLEANUP
                                                                                TRAP    C$DCLN
033764  104444

033766  012737  177777  007466  INIT1:  MOV     #-1,RESFLG              ;SET RESTART FLAG
033774                                  READEF  #EF.START               ;IF HERE CAUSE OF START,DO SOME INIT
033774  012700  000040                                                          MOV     #EF.START,R0
034000  104447                                                                  TRAP    C$REFG

034002                                  BCOMPLETE       START
034002  103417                                                                  BCS     START

034004                                  READEF  #EF.RESTART             ;IF HERE CAUSE OF RESTART, DO SOME INIT
034004  012700  000037                                                          MOV     #EF.RESTART,R0
034010  104447                                                                  TRAP    C$REFG

034012                                  BCOMPLETE       RESTRT
034012  103513                                                                  BCS     RESTRT

034014                                  READEF  #EF.CONTINUE            ;SEE IF WE'RE HERE CAUSE OF A CONTINUE
034014  012700  000036                                                          MOV     #EF.CONTINUE,R0
034020  104447                                                                  TRAP    C$REFG

034022                                  BNCOMPLETE      S1              ;BR IF NOT HERE CAUSE OF CONITNUE
034022  103002                                                                  BCC     S1

034024  000137  034530                  JMP     ENDIT                   ;JMP IF HERE CAUSE OF A CONTINUE
034030                          S1:     READEF  #EF.NEW                 ;SEE IF THIS IS A 'NEW PASS''
034030  012700  000035                                                          MOV     #EF.NEW,R0
034034  104447                                                                  TRAP    C$REFG

034036                                  BCOMPLETE       NEW             ;IF YES, BR AROUND LOGUNIT # SETUP
034036  103521                                                                  BCS     NEW

034040  000523                          BR      GETPRM

034042  005037  007466          START:  CLR     RESFLG                  ;CLEAR RESTART FLAG SINCE HERE ON START
034046  005037  007526                  CLR     CLKVEC                  ;CLEAR CLK VECTOR PTR. AS A FLAG IN
                                                                        ; NO CLOCK IS FOUND.
034052  012702  007522                  MOV     #CLKCSR,R2              ;SETUP R2 AS A PTR. TO CLOCK INFO BLOCK
034056                                  CLOCK   L,R1                    ;LOOK FOR A LINE CLOCK
034056  012700  000114                                                          MOV     #'L,R0
034062  104462                                                                  TRAP    C$CLCK
034064  010001                                                                  MOV     R0,R1

034066                                  BNCOMPLETE      S2              ; IF NONE THERE GO LOOK FOR A P-CLOCK
034066  103006                                                                  BCC     S2

034070  004737  026204                  JSR     PC,CLKSET               ; GO SET UP CLOCK INFO TABLE & CLK VEC.
034074  012737  000100  007532          MOV     #LCLKEN,CLKEN           ;SETUP THE ENABLE LINE CLOCK DATA
034102  000457                          BR      RESTRT

034104                          S2:     CLOCK   P,R1                    ;LOOK FOR A P-CLOCK SINCE NO LINE CLOCK
```

```
                                                                                MOV     #'P,R0
                                                                                TRAP    C$CLCK
                                                                                MOV     R0,R1
                        BNCOMPLETE      S3              ; IF NONE THERE GO SEE IF THIS IS LSI
                                                                                BCC     S3
                        JSR     PC,CLKSET               ; ELSE GO SET UP CLOCK INFO & VECTOR
                        ADD     #2,CLKCSR               ;POINT CLKCSR TO P-CLK COUNT SET REG.
                        MOV     #PCLKCT,@CLKCSR         ;LOAD CLK SET REG. WITH COUNT VALUE
                        SUB     #2,CLKCSR               ;POINT CLKCSR BAC TO P-CLK CSR
                        MOV     #PCLKEN,CLKEN           ;SETUP THE ENABLE THE P-CLK DATA
                        BR      RESTRT

                S3:     READBUS                         ;READ BUS TYPE TO SEE IF ON AN LSI
                                                                                TRAP    C$RDBU
                        BNCOMPLETE      S4              ;BR IF NOT, NO CHANCE OF A CLOCK
                                                                                BCC     S4
                        MOV     #100,CLKVEC             ;LOAD 100 AS CLK VECTOR
                        CLR     CLKBR                   ;LOAD 0 AS CLK INT. LEVEL
                        MOV     #CLKEN,CLKCSR           ;KLUDGE UP THE CSR & ENABLE DATA LOCS
                        GMANID  L5060,CLKHZ,D,377,50..,60..,YES
                                                                                TRAP    C$GMAN
                                                                                BR      10000$
                                                                                .WORD   CLKHZ
                                                                                .WORD   T$CODE
                                                                                .WORD   L5060
                                                                                .WORD   377
                                                                                .WORD   T$LOLIM
                                                                                .WORD   T$HILIM
                                                                        10000$:
                        BR      RESTRT

                S4:     PRINTF  #NOCLK                  ;INFORM OPR. NO CLOCK, & EXIT INIT
                                                                                MOV     #NOCLK,-(SP)
                                                                                MOV     #1,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTF
                                                                                ADD     #4,SP

                RESTRT: CLR     TIMMIN                  ;CLEAR TIME SINCE START LOCATIONS
                        CLR     TIMSEC
                        MOV     CLKHZ,TIMTCK            ;LOAD TICKS/SEC
                        MOV     #EVTLOG,R2              ;INIT EVENT TABLE TO ALL 1'S AFTER EACH
                        MOV     R2,EVTPTR               ; START OR RES AND INIT TABLE POINTER
                1$:     MOV     #-1,(R2)+
                        CMP     R2,#EVTEND              ;SEE IF REACHED END OF TABLE
                        BNE     1$                      ;LOOP UNTIL DONE

                NEW:    MOV     #-1,LOGUNT              ;INITIALIZE LOGICAL UNIT #

                GETPRM: INC     LOGUNT                  ;POINT TO NEXT LOGICAL UNIT
                        CMP     LOGUNT,L$UNIT           ;SEE IF PAST MAX. LOG. UNIT #
                        BGE     NEW                     ;BR IF YES, AND START OVER

                        GPHARD  LOGUNT,R1               ;GET THE P-TABLE FOR THIS LOG. UNIT
                                                                                MOV     LOGUNT,R0
                                                                                TRAP    C$GPHRD
```

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

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

                        ;DEVICE DEPENDENT PART OF GETTING INFO FROM P-TABLE

                        MOV     2(R1),SEL0              ;STORE AWAY CSR ADDRESSES
                        MOV     2(R1),BSEL1
                        INC     BSEL1
                        MOV     2(R1),SEL2
                        ADD     #2,SEL2
                        MOV     2(R1),BSEL3
                        ADD     #3,BSEL3
                        MOV     2(R1),SEL4
                        ADD     #4,SEL4
                        MOV     2(R1),BSEL5
                        ADD     #5,BSEL5
                        MOV     2(R1),SEL6
                        ADD     #6,SEL6
                        MOV     2(R1),BSEL7
                        ADD     #7,BSEL7

                        MOV     4(R1),INVEC             ;STORE AWAY INPUT INTERRUPT VECTOR
                        MOV     4(R1),OUTVEC
                        ADD     #4,OUTVEC               ;BUILD OUTPUT INTERRUPT VECTOR
                        MOV     6(R1),INTPRI            ;STORE AWAY INTERRUPT PRIORITY
                        MOV     12(R1),OPTYP            ;STORE AWAY DEVICE OPTION TYPE

ENDIT:
                        SETVEC  CLKVEC,#CLKINT,#340     ;SETUP CLOCK VECTOR
                                                                    MOV     #340,-(SP)
                                                                    MOV     #CLKINT,-(SP)
                                                                    MOV     CLKVEC,-(SP)
                                                                    MOV     #3,-(SP)
                                                                    TRAP    C$SVEC
                                                                    ADD     #10,SP

                        ;DEVICE DEPENDENT VECTOR SETUP

                        SETVEC  INVEC,#DVINS,INTPRI     ;SETUP INPUT INTERRUPT VECTOR
                                                                    MOV     INTPRI,-(SP)
                                                                    MOV     #DVINS,-(SP)
                                                                    MOV     INVEC,-(SP)
                                                                    MOV     #3,-(SP)
                                                                    TRAP    C$SVEC
                                                                    ADD     #10,SP
                        SETVEC  OUTVEC,#DVOUTS,INTPRI   ;SETUP OUTPUT INTERRUPT VECTOR
                                                                    MOV     INTPRI,-(SP)
                                                                    MOV     #DVOUTS,-(SP)
                                                                    MOV     OUTVEC,-(SP)
                                                                    MOV     #3,-(SP)
                                                                    TRAP    C$SVEC
                                                                    ADD     #10,SP
```

```
 4770
 4771  034632                          SETPRI  #PRI00          ;SET THE "RUN" PRIORITY TO 0
 4772  034632  012700  000000                                          MOV     #PRI00,R0
 4773  034636  104441                                                  TRAP    C$SPRI
 4774  034640                          EXIT    INIT
 4775  034640  104432                                                  TRAP    C$EXIT
 4776  034642  000002                                                  .WORD   L10013-.
 4777
 4778
 4779                                  .EVEN
 4780
 4781  034644                          ENDINIT
 4782  034644                                                  L10013:
 4783  034644  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.
                                ;--

034646                                  BGNAUTO
034646                                                                  L$AUTO::

034646                                  ENDAUTO
034646                                                                  L10014:
034646  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.
                                ;--

034650                                  BGNCLN
034650                                                                          L$CLEAN::

034650  004737  045774                  JSR     PC,DVBTUP       ;GO UPDATE BASE TABLE
034654  005077  152642                  CLR     @CLKCSR         ;DISABLE CLOCK
034660                                  SETPRI  #PRI07          ;SET PROCESSOR PRIORITY BACK TO 7
034660  012700  000340                                                          MOV     #PRI07,R0
034664  104441                                                                  TRAP    C$SPRI
034666  022737  000057  003202          CMP     #EXIT,KEYWD1    ;"EXIT" COMMAND ?
034674  001416                          BEQ     EXITCLN         ;YES,BRANCH
                                ;;^C WAS ENTERED-- LOG IT
034676  012737  000026  007426          MOV     #ABO,TEMP       ;EVENT TYPE
034704  013737  007374  007436          MOV     NOBUF,TEMP4     ;;BUFFER NOT AVAILABLE
034712  013737  007376  007432          MOV     PSCNT,TEMP2     ;;PASSES
034720  013737  007400  007434          MOV     ERRCNT,TEMP3    ;;ERRORS
034726  004737  026742                  CALL    LOGS5           ;GO LOG IT
034732                          EXITCLN:EXIT    CLN
034732  104432                                                                  TRAP    C$EXIT
034734  000002                                                                  .WORD   L10015-.

                                .EVEN

034736                          ENDCLN
034736                                                                  L10015:
034736  104412                                                                  TRAP    C$CLEAN
```

```
 4833                                   .SBTTL  DROP UNIT SECTION
 4834
 4835                                   ;++
 4836                                   ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
 4837                                   ; TO NO LONGER BE TESTED.
 4838                                   ;--
 4839
 4840  034740                                   BGNDU
 4841  034740                                                                        L$DU::
 4842
 4843
 4844  034740                                   EXIT    DU
 4845  034740  000167                                                                       .WORD   J$JMP
 4846  034742  000000                                                                       .WORD   L10016-2-.
 4847
 4848
 4849                                   .EVEN
 4850
 4851  034744                                   ENDDU
 4852  034744                                                                        L10016:
 4853  034744  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.
                                ;--

034746                                  BGNAU
034746                                                                  L$AU::

                                        EXIT    AU
034746  000167                                                                  .WORD   J$JMP
034750  000000                                                                  .WORD   L10017-2-.

                                        .EVEN

034752                                  ENDAU
034752                                                                  L10017:
034752  104452                                                                  TRAP    C$AU
```

```
                                .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.
                                ;--


034754                                  BGNTST
034754                                                                          T1::

                                .SBTTL          PROGRAM SETUP SECTION

034754  013777  007532  152540          MOV     CLKEN,@CLKCSR   ;ENABLE THE CLOCK

034762                          GTXRXB:
034762  005001                  GTRA2:  CLR     R1
034764  012737  000001  007542          MOV     #1,TIMER1       ;SET TIMER TO COUNT 1 TICK
034772  005737  007542          1$:     TST     TIMER1          ;CHECK FOR IT TO BE COUNTED OFF
034776  001412                          BEQ     GTRA3           ;BRANCH IF CLOCK EXISTS (COUNTED A TICK)
035000  005301                          DEC     R1
035002  001373                          BNE     1$              ;KEEP CHECKING UNTIL R1 DOES FULL COUNTDWN
035004                                  PRINTF  #NOCLK          ;PRINT BAD CLK MSG AND WARN OF HANG IF TIMEOUT
035004  012746  015106                                                  MOV     #NOCLK,-(SP)
035010  012746  000001                                                  MOV     #1,-(SP)
035014  010600                                                          MOV     SP,R0
035016  104417                                                          TRAP    C$PNTF
035020  062706  000004                                                  ADD     #4,SP

035024  005737  007466          GTRA3:  TST     RESFLG          ;SEE IF HERE AFTER A RESTART.
035030  001117                          BNE     GTRA5           ;BR IF HERE CAUSE OF A RESTART

                                        ; CLEAR COUNTS AND SET UP DEFAULTS

035032  005037  007422          GTRA4:  CLR     TOTCC           ;CLEAR TOTAL CHAR. COUNT TEMP. LOC.
035036  005037  007356                  CLR     TTOTCC          ; CLEAR TOTAL CHAR. COUNT FOR TX BUFF
035042  005037  007342                  CLR     CTOTCC          ; CLEAR TOTAL CHAR. COUNT FOR CMP BUFF
035046  012701  006562                  MOV     #PTRTAB,R1      ;INIT TRANSMIT MESSAGE POINTER
035052  010137  007334                  MOV     R1,TXPTR
035056  005037  007332                  CLR     RXPTR           ; ZERO RX POINTER
035062  012702  000017                  MOV     #MSGLIM,R2
035066  006302                          ASL     R2
035070  006302                          ASL     R2
035072  010137  007336                  MOV     R1,CMPPTR
035076  060237  007336                  ADD     R2,CMPPTR       ;INIT COMPARE MESSAGE POINTER

035102  012737  000005  007410          MOV     #5,MSGTYP       ;SET UP DEFAULT MSG TYPE (QUICK FOX - ITEP MSG)
035110  013737  002162  007412          MOV     MSG5C,CURCC     ;SET UP DEFAULT CHAR COUNT
035116  012737  003562  007360          MOV     #TXBUF,TCURAD   ;SET UP CURRENT ADD TO START OF TX BUFFER
035124  012737  005562  007344          MOV     #CMPBUF,CCURAD  ;SET UP CURRENT ADD TO START OF CMP BUFFER

035132  013737  007360  007420          MOV     TCURAD,CURADD   ;SETUP CURRENT ADDR TO START OF TXBUF
```

```
 4934  035140  013737  007334  007416          MOV     TXPTR,CPTR      ;SETUP CURRENT POINTER TABLE POINTER FOR TXBUF
 4935  035146  004737  032010                  JSR     PC,BLDBUF       ; GO BUILD POINTER TABLE AND BUFFER
 4936  035152  012737  000001  007354          MOV     #1,TXMTOT       ;BUMP TOTAL MESSAGE COUNT
 4937
 4938  035160  013737  007336  007416          MOV     CMPPTR,CPTR     ;SET UP START OF COMPARE POINTER TABLE
 4939  035166  013737  007344  007420          MOV     CCURAD,CURADD   ;SET UP CURRENT ADDR. TO START OF CMPBUF
 4940  035174  012737  000005  007410          MOV     #5,MSGTYP
 4941  035202  013737  002162  007412          MOV     MSG5C,CURCC
 4942  035210  004737  032010                  JSR     PC,BLDBUF       ;PUT DEFAULT MESSAGE INTO CMPBUF
 4943  035214  012737  000001  007340          MOV     #1,CMPTOT       ;BUMP THE COMP MESG COUNT
 4944  035222  012737  000003  007470          MOV     #ACT,MODTYP     ;SET DEFAULT MODE= ACTIVE
 4945  035230  005037  007472                  CLR     MLTYP           ;SET DEFAULT MAINTENANCE LOOP MODE =NONE
 4946  035234  012737  000001  007500          MOV     #1,RPASS        ;SET UP DEFAULT "RUN PASS" COUNT TO 1
 4947  035242  012737  000002  007476          MOV     #2,PARAM        ;SET UP PROG. PARAMETERS - DATACHECKING ENABLED
 4948                                          ;                       OPERATOR STATUS MSGS. PRINT OFF
 4949  035250                                  PRINTF  #HLP0
 4950  035250  012746  013153                                                  MOV     #HLP0,-(SP)
 4951  035254  012746  000001                                                  MOV     #1,-(SP)
 4952  035260  010600                                                          MOV     SP,R0
 4953  035262  104417                                                          TRAP    C$PNTF
 4954  035264  062706  000004                                                  ADD     #4,SP
 4955  035270  013737  007470  010654  GTRA5:  MOV     MODTYP,DEV1
 4956  035276  013737  007472  010656          MOV     MLTYP,DEV2
 4957  035304  013737  007500  010660          MOV     RPASS,DEV3
 4958  035312  013737  007476  010662          MOV     PARAM,DEV4
 4959  035320  004737  032340                  JSR     PC,SHWOP        ;PRINT TO OPERATOR THE CURRENT MODE.....
 4960
 4961  035324                                  MANUAL                  ;SEE IF MANUAL INTERVENTION ALLOWED
 4962  035324  104450                                                          TRAP    C$MANI
 4963  035326                                  BCOMPLETE       GETCL   ;  BR IF YES (UAM=0 AND NOT CHAINED)
 4964  035326  103412                                                          BCS     GETCL
 4965  035330  005737  007500                  TST     RPASS           ;SEE IF THIS IS FIRST "DCLT PASS"
 4966  035334  001002                          BNE     1$              ; BR IF NOT COMPLETED 1 PASS
 4967  035336                                  EXIT    TST             ; IF DONE 1 PASS IN UNATTENDED MODE - EXIT
 4968  035336  104432                                                          TRAP    C$EXIT
 4969  035340  010574                                                          .WORD   L10020-.
 4970  035342  012737  000001  007472  1$:     MOV     #TTL,MLTYP      ;SET UP DEFAULT FOR UNATTENDED MODE
 4971  035350  000137  040400                  JMP     GTR9            ;  "R M=ACT/LO=I/PAS=1/NOST/CH" AND RUN
 4972
 4973                                  .SBTTL          COMMAND LINE FETCH & INTERPRETATION SECTION
 4974
 4975  035354  105037  003561          GETCL:  CLRB    P$GDBD          ;CLEAR CMD LINE PARSING ERROR FLAGS
 4976  035360  105037  003560                  CLRB    P$NNUF
 4977  035364                                  GMANID  CLI$PM,CMDBUF,A,0,1,72.,NO      ;GET A COMMAND LINE FROM OPR.
 4978  035364  104443                                                          TRAP    C$GMAN
 4979  035366  000406                                                          BR      10000$
 4980  035370  003060                                                          .WORD   CMDBUF
 4981  035372  000142                                                          .WORD   T$CODE
 4982  035374  012502                                                          .WORD   CLI$PM
 4983  035376  000000                                                          .WORD   0
 4984  035400  000001                                                          .WORD   T$LOLIM
 4985  035402  000110                                                          .WORD   T$HILIM
 4986  035404                                                          10000$:
 4987  035404  012737  003060  003544          MOV     #CMDBUF,P$BUFA
 4988  035412  012737  010664  003546          MOV     #CLITRE,P$TREE
 4989  035420  012737  036346  003550          MOV     #CLIACT,P$ACT
```

```
 4990  035426  005037  003204                  CLR     QUALFG                  ;CLEAR QUALIFIER FLAG LOCATION
 4991  035432  004737  032642                  JSR     PC,P$TRV                ;GO PARSE COMMAND LINE
 4992  035436  105737  003561                  TSTB    P$GDBD                  ;SEE IF PARSED OK OR AN ERROR
 4993  035442  001412                          BEQ     1$
 4994  035444                                  PRINTF  #CLIERM
 4995  035444  012746  012516                                                          MOV     #CLIERM,-(SP)
 4996  035450  012746  000001                                                          MOV     #1,-(SP)
 4997  035454  010600                                                                  MOV     SP,R0
 4998  035456  104417                                                                  TRAP    C$PNTF
 4999  035460  062706  000004                                                          ADD     #4,SP
 5000  035464  000137  035354                  JMP     GETCL
 5001  035470  105737  003560          1$:     TSTB    P$NNUF                  ;SEE IF INCOMPLETE COMMAND TYPED
 5002  035474  001412                          BEQ     10$
 5003  035476                                  PRINTF  #CLINUF
 5004  035476  012746  012546                                                          MOV     #CLINUF,-(SP)
 5005  035502  012746  000001                                                          MOV     #1,-(SP)
 5006  035506  010600                                                                  MOV     SP,R0
 5007  035510  104417                                                                  TRAP    C$PNTF
 5008  035512  062706  000004                                                          ADD     #4,SP
 5009  035516  000137  035354                  JMP     GETCL
 5010
 5011                                          ;REV B BY EC
 5012  035522  023727  003202  000060  10$:     CMP     KEYWD1,#SETET           ;WAS "SET EXPECT = TRANMIT" ENTERED?
 5013  035530  001711                          BEQ     GETCL                   ;YES,BRANCH
 5014
 5015  035532  023727  003202  000005          CMP     KEYWD1,#HLP             ;SEE IF HELP WAS TYPED
 5016  035540  001705                          BEQ     GETCL                   ;GO GET CMD AGAIN IF YES
 5017  035542  023727  003202  000055          CMP     KEYWD1,#PRNT            ;SEE IF PRINT WAS TYPED
 5018  035550  001701                          BEQ     GETCL                   ; GO GET CMD AGAIN IF YES
 5019  035552  023727  003202  000004          CMP     KEYWD1,#RUN             ;SEE IF RUN WAS TYPED
 5020  035560  001002                          BNE     11$                     ; BR IF NO
 5021  035562  000137  040400                  JMP     GTR9                    ; START EXEC. IF YES
 5022  035566  023727  003202  000052  11$:     CMP     KEYWD1,#DMPS            ;SEE IF DUMP WAS TYPED
 5023  035574  001004                          BNE     12$                     ; BR IF NO
 5024  035576  004737  031554                  JSR     PC,DUMPSR               ; ELSE, DUMP PART OF MEMORY
 5025  035602  000137  035354                  JMP     GETCL                   ; THEN RETURN TO GET ANOTHER CMD.
 5026                                          ;;EXIT COMMAND IS A REVISION B CHANGE BY EC
 5027  035606  023727  003202  000057  12$:     CMP     KEYWD1,#EXIT            ;EXIT COMMAND ?
 5028  035614  001005                          BNE     13$                     ;NO,BRANCH
 5029  035616  012737  000001  007464          MOV     #1,DCLFLG               ;SET CLEANUP & EXIT FLAG
 5030  035624                                  EXIT    TST                     ;GO BACK TO INIT ROUTINE
 5031  035624  104432                                                                  TRAP    C$EXIT
 5032  035626  010306                                                                  .WORD   L10020-.
 5033  035630  023727  003202  000001  13$:     CMP     KEYWD1,#CLEAR           ;SEE IF CLEAR WAS TYPED
 5034  035636  001646                          BEQ     GETCL                   ; IF YES, BACK TO GET ANOTHER CMD.
 5035  035640  023727  003202  000002          CMP     KEYWD1,#SHOW            ;SEE IF SHOW WAS TYPED
 5036  035646  001642                          BEQ     GETCL                   ; IF YES, BACK TO GET ANOTHER CMD.
 5037  035650  023727  003202  000010  4$:      CMP     KEYWD1,#SETEXP          ;SEE IF SET EXPECTED
 5038  035656  001512                          BEQ     2$                      ; BR IF YES (A SETEXP WAS TYPED)
 5039  035660  013737  007356  007422  5$:      MOV     TTOTCC,TOTCC
 5040  035666  023727  007422  001000          CMP     TOTCC,#BUFLIM           ;SEE IF BUFFER ALREADY FULL
 5041  035674  002414                          BLT     15$                     ; BR IF NOT FULL (BUFLIM # OF CHARS.)
 5042  035676                                  PRINTF  #MSGTRN,#BUFEX          ; ELSE TELL OPR. AND DON'T BUILD MSG.
 5043  035676  012746  015227                                                          MOV     #BUFEX,-(SP)
 5044  035702  012746  015245                                                          MOV     #MSGTRN,-(SP)
 5045  035706  012746  000002                                                          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
007334          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
007416          007334  MOV     TXPTR,CPTR              ;SETUP CHAR. COUNT, CURRENT ADDR, & PTR
007420          007360  MOV     TCURAD,CURADD
                        JSR     PC,ADDCC                ;ADD IN CHAR. COUNT AND CHECK TOTAL
                        JSR     PC,BLDBUF               ;GO BUILD MESSAGE IN BUFFER AND PTRS.
007334          007416  MOV     CPTR,TXPTR
007356          007422  MOV     TOTCC,TTOTCC            ;UPDATE CHAR. COUNT, CURR ADDR, & PTR
007360          007420  MOV     CURADD,TCURAD
                        INC     TXMTOT
                        DEC     QUALVL                  ;DEC THE COPY COUNT
                        BNE     5$
                        JMP     GETCL

007422          007342  2$:     MOV     CTOTCC,TOTCC    ;SETUP CHAR. COUNT, CURR. ADDR. & PTR
001000          007422          CMP     TOTCC,#BUFLIM   ;SEE IF BUFFER ALREADY FULL
                                BLT     16$             ; 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$:     MOV     #PTRTAB,R1
                                MOV     #MSGLIM,R2
                                ASL     R2
                                ASL     R2
                                MOV     R1,CMPPTR
                                ADD     R2,CMPPTR       ;INIT COMPARE MESSAGE POINTER
                                MOV     CMPTOT,R1
                                CMP     R1,#MSGLIM      ;SEE IF MSG COUNT EXCEEDED.
```

```
 5102  036220  002414                          BLT     18$                     ; BR IF NO
 5103  036222                                  PRINTF  #MSGTRN,#TABEX          ; ELSE TELL OPR. AND DON'T BUILD MSG.
 5104  036222  012746  015167                                                          MOV     #TABEX,-(SP)
 5105  036226  012746  015245                                                          MOV     #MSGTRN,-(SP)
 5106  036232  012746  000002                                                          MOV     #2,-(SP)
 5107  036236  010600                                                                  MOV     SP,R0
 5108  036240  104417                                                                  TRAP    C$PNTF
 5109  036242  062706  000006                                                          ADD     #6,SP
 5110  036246  000137  035354                  JMP     GETCL                   ; THEN GO GET A NEW COMMAND.
 5111  036252  006301                  18$:    ASL     R1                      ;# OF MSGS *4 = NEXT FREE PTR BLOCK
 5112  036254  006301                          ASL     R1
 5113  036256  060137  007336                  ADD     R1,CMPPTR
 5114  036262  013737  007336  007416          MOV     CMPPTR,CPTR
 5115  036270  013737  007344  007420          MOV     CCURAD,CURADD
 5116  036276  004737  031712                  JSR     PC,ADDCC                ;ADD IN XHAR. COUNT AND CHECK TOTAL
 5117  036302  004737  032010                  JSR     PC,BLDBUF
 5118  036306  013737  007416  007336          MOV     CPTR,CMPPTR
 5119  036314  005237  007340                  INC     CMPTOT
 5120  036320  013737  007420  007344          MOV     CURADD,CCURAD           ;UPDATE CHAR. COUNT, CURR ADDRR. & PTR
 5121  036326  013737  007422  007342          MOV     TOTCC,CTOTCC
 5122  036334  005337  003206                  DEC     QUALVL                  ;IF COPY WAS GIVEN, PUT MSG IN BUFF
 5123  036340  001261                          BNE     2$                      ; AGAIN
 5124  036342  000137  035354                  JMP     GETCL                   ;GO BACK UNTIL GET A "RUN"
 5125
 5126
 5127
 5128
 5129
```

```
                                        .SBTTL          ACTION TABLE AND ROUTINES
                                        ;               USER MUST CLEAR/SET P$GDBD IF USE "CLIBIF" IN CONNECTION WITH ACTION
                                        ;               R2 WILL HOLD ACTION CODE FROM PARSING (CLI) NODE
036346                                  CLIACT:
036346  006302                                  ASL     R2                      ;MULTIPLY ACTION CODE BY 2
036350  016202  036364                          MOV     10$(R2),R2              ;OFFSET VALUE
036354  062702  036364                          ADD     #10$,R2                 ;ADD BASE VALUE
036360  004712                                  JSR     PC,(R2)                 ;GO DO ACTION
036362  000207                                  RTS     PC                      ;RETURN TO TRVACT:

                                                                                ;BRIEF DESCRIPTION OF ACTIONS TAKEN
036364  000150                          10$:    .WORD   ACTNUL-10$              ;NULL
036366  000152                                  .WORD   ACTCLR-10$              ;CLEAR
036370  000162                                  .WORD   ACTSHO-10$              ;SHOW
036372  001604                                  .WORD   ACTCHK-10$              ;CHECK
036374  000262                                  .WORD   ACTRUN-10$              ;RUN
036376  000172                                  .WORD   ACTHLP-10$              ;HELP
036400  000306                                  .WORD   ACTCSE-10$              ;CLEAR OR SHOW EXPECTED
036402  000434                                  .WORD   ACTCST-10$              ;CLEAR OR SHOW TRANSMIT
036404  000756                                  .WORD   ACTSTE-10$              ;SET EXPECTED
036406  000766                                  .WORD   ACTSTT-10$              ;SET TRANSMIT
036410  001004                                  .WORD   ACTSZE-10$              ;SIZE
036412  001014                                  .WORD   ACTCOP-10$              ;COPY
036414  001024                                  .WORD   ACTNUM-10$              ;NUMERIC VALUE FOR SIZE OR COPY
036416  001116                                  .WORD   ACTOPM-10$              ;QUOTED MESSAGE FROM USER
036420  001612                                  .WORD   ACTSTS-10$              ;STATUS
036422  001136                                  .WORD   ACTEQQ-10$              ;END OF QUOTED MESSAGE FROM USER
036424  001216                                  .WORD   ACTMS0-10$              ;ONES
036426  001224                                  .WORD   ACTMS1-10$              ;ZEROS
036430  001234                                  .WORD   ACTMS2-10$              ;1ALT
036432  001244                                  .WORD   ACTMS3-10$              ;0ALT
036434  001254                                  .WORD   ACTMS4-10$              ;ITEP
036436  001264                                  .WORD   ACTMS5-10$              ;CCITT
036440  001302                                  .WORD   ACTMS6-10$              ;ALPHA
036442  001370                                  .WORD   ACTATV-10$              ;ACTIVE MODE
036444  001400                                  .WORD   ACTPAS-10$              ;PASSIVE MODE
036446  001420                                  .WORD   ACTREC-10$              ;RECEIVE MODE
036450  001426                                  .WORD   ACTLIS-10$              ;LISTEN MODE
036452  001436                                  .WORD   ACTDLL-10$              ;DOWNLINE LOAD
036454  001446                                  .WORD   ACTTRA-10$              ;TRANSMIT MODE
036456  001456                                  .WORD   ACTTAL-10$              ;TALK MODE
036460  001504                                  .WORD   ACTNO-10$               ;NO
036462  001514                                  .WORD   ACTECH-10$              ;ECHO
036464  001620                                  .WORD   ACTCRC-10$              ;SET CRC BIT
036466  001634                                  .WORD   ACTPRO-10$              ;SET PROTOCOL BIT
036470  001666                                  .WORD   ACTRPS-10$              ;STATUS
036472  001676                                  .WORD   ACTMOP-10$              ;REMOTE STATION IN MAINTENANCE LOOP MODE
036474  001706                                  .WORD   ACTTLP-10$              ;INTERNALTTL
036476  001716                                  .WORD   ACTCLP-10$              ;CABLE LOOP
036500  001726                                  .WORD   ACTLLP-10$              ;LOCAL MODEM LOOP
036502  001736                                  .WORD   ACTRLP-10$              ;REMOTE MODEM LOOP
036504  000142                                  .WORD   ACTNUF-10$              ;MORE COMMAND LINE NEEDED
036506  001174                                  .WORD   ACTBCR-10$              ;BAD CHARACTER IN OPERATOR MESSAGE
036510  000712                                  .WORD   ACTDMS-10$              ;DUMP MEMORY START ADDRESS
036512  000742                                  .WORD   ACTDME-10$              ;DUMP MEMORY END ADDRESS
```

```
 5186  036514  000734
 5187  036516  000246
 5188  036520  001626
 5189  036522  000236
 5190  036524  001326
 5191
```

```
.WORD   ACTDMQ-10$      ;DUMP WORD
.WORD   ACTPRT-10$      ;PRINT
.WORD   ACTMOS-10$      ;MODEM ACTION REV B BY EC
.WORD   ACTEXT-10$      ;EXIT ACTION  REV B BY EC
.WORD   ACTSEX-10$      ;SET E=T ACTION REV B BY EC NPI
```

```
5192
5193  036526  112737  177777  003560  ACTNUF: MOVB    #-1,P$NNUF      ;SET FLAG TO SAY NEED MORE OF COMMAND
5194  036534  000207                  ACTNUL: RTS     PC              ;RETURN TO PARSER
5195
5196  036536  012737  000001  003202  ACTCLR: MOV     #CLEAR,KEYWD1   ;SET LOC TO SAY A CLEAR WAS TYPED
5197  036544  000207                          RTS     PC
5198
5199  036546  012737  000002  003202  ACTSHO: MOV     #SHOW,KEYWD1    ;SET LOC. TO SAY A SHOW WAS TYPED
5200  036554  000207                          RTS     PC
5201
5202  036556  012702  003210          ACTHLP: MOV     #HLPTAB,R2      ;SETUP R2 AS A POINTER TO HELP MSG TABLE
5203  036562                          1$:     PRINTF  #HLPF,(R2)+     ;PRINT HELP INFORMATION MESSAGES
5204  036562  012246                                                          MOV     (R2)+,-(SP)
5205  036564  012746  013231                                                  MOV     #HLPF,-(SP)
5206  036570  012746  000002                                                  MOV     #2,-(SP)
5207  036574  010600                                                          MOV     SP,R0
5208  036576  104417                                                          TRAP    C$PNTF
5209  036600  062706  000006                                                  ADD     #6,SP
5210  036604  020227  003230                  CMP     R2,#HLPEND      ;SEE IF ALL INFO PRINTED YET
5211  036610  001364                          BNE     1$              ;IF NO KEEP PRINTING
5212  036612  012737  000005  003202          MOV     #HLP,KEYWD1     ;SET LOC. TO SAY A HELP WAS TYPED
5213  036620  000207                          RTS     PC
5214  036622  012737  000057  003202  ACTEXT: MOV     #EXIT,KEYWD1    ;EXIT COMMAND
5215  036630  000207                          RTS     PC
5216  036632  012737  000055  003202  ACTPRT: MOV     #PRNT,KEYWD1    ;SET LOC. TO SAY A HELP WAS TYPED
5217  036640  004737  027046                  JSR     PC,REPORT       ;CALL ROUTINE TO PRINT EVENT LOG AND BASE TABLE
5218  036644  000207                          RTS     PC
5219
5220  036646  012737  000004  003202  ACTRUN: MOV     #RUN,KEYWD1     ;SET RUN FLAG
5221  036654  112737  177777  003560          MOVB    #-1,P$NNUF      ;SET FLAG TO SAY NEED MORE OF COMMAND
5222  036662  012737  000001  007500          MOV     #1,RPASS        ;SET DEFAULT RUN "PASS" TO 1
5223  036670  000207                          RTS     PC
5224
5225  036672  012701  006562          ACTCSE: MOV     #PTRTAB,R1
5226  036676  012702  000017                  MOV     #MSGLIM,R2
5227  036702  006302                          ASL     R2
5228  036704  006302                          ASL     R2
5229  036706  010137  007336                  MOV     R1,CMPPTR
5230  036712  060237  007336                  ADD     R2,CMPPTR       ;INIT COMPARE MESSAGE POINTER
5231  036716  013701  007336                  MOV     CMPPTR,R1
5232
5233  036722  013702  007340                  MOV     CMPTOT,R2
5234  036726  105037  003560                  CLRB    P$NNUF          ;FLAG THAT HAVE VALID COMMAND AT THIS PT.
5235  036732  023727  003202  000002          CMP     KEYWD1,#SHOW    ;SEE IF A CLEAR OR SHOW WAS TYPED
5236  036740  001500                          BEQ     ACTSHW          ;BR IF A SHOW WAS TYPED
5237  036742  012737  000001  007340          MOV     #1,CMPTOT       ;CLEAR COMPARE MESSAGE COUNT, CHAR. COUNT
5238  036750  005037  007342                  CLR     CTOTCC          ; AND RESET POINTER
5239
5240  036754  012701  006562                  MOV     #PTRTAB,R1
5241  036760  012702  000017                  MOV     #MSGLIM,R2
5242  036764  006302                          ASL     R2
5243  036766  006302                          ASL     R2
5244  036770  010137  007336                  MOV     R1,CMPPTR
5245  036774  060237  007336                  ADD     R2,CMPPTR       ;INIT COMPARE MESSAGE POINTER
5246  037000  013737  007336  007416          MOV     CMPPTR,CPTR     ;SET UP TO FILL IN DEFAULT MESSAGE
5247  037006  012701  005562                  MOV     #CMPBUF,R1
```

```
5248  037012  010137  007344            MOV     R1,CCURAD
5249  037016  000431                    BR      ACTCLB
5250
5251  037020  012701  006562    ACTCST: MOV     #PTRTAB,R1
5252  037024  013702  007354            MOV     TXMTOT,R2
5253  037030  105037  003560            CLRB    P$NNUF          ;FLAG THAT HAVE VALID COMMAND AT THIS PT.
5254  037034  023727  003202  000002    CMP     KEYWD1,#SHOW    ;SEE IF A CLEAR OR SHOW WAS TYPED
5255  037042  001437                    BEQ     ACTSHW          ;BR IF A SHOW WAS TYPED
5256  037044  012737  000001  007354    MOV     #1,TXMTOT       ;CLEAR TRANSMIT MESSAGE COUNT, CHAR. COUNT
5257  037052  005037  007356            CLR     TTOTCC          ; AND RESET POINTER
5258  037056  012737  006562  007334    MOV     #PTRTAB,TXPTR
5259  037064  013737  007334  007416    MOV     TXPTR,CPTR
5260  037072  012701  003562            MOV     #TXBUF,R1
5261  037076  010137  007360            MOV     R1,TCURAD
5262
5263  037102  012702  001000    ACTCLB: MOV     #BUFLIM,R2
5264  037106  010137  007420            MOV     R1,CURADD       ;SET UP TO PUT DEFAULT MSG IN LIST AFTER 033'S
5265  037112  012737  000005  007410    MOV     #5,MSGTYP
5266  037120  013737  002162  007412    MOV     MSG5C,CURCC
5267  037126  105021            1$:     CLRB    (R1)+           ;FILL EXPT OR TRAN BUFFER WITH 0'S IF A CLEAR
5268  037130  005302                    DEC     R2              ;DO 'BUFLIM" NUMBER OF BYTE LOCATIONS
5269  037132  001375                    BNE     1$
5270  037134  004737  032010            JSR     PC,BLDBUF       ;"CLEAR" REALLY MEANS TO PUT DEFAULT MSG IN
5271  037140  000207                    RTS     PC              ;WHEN DONE, RETURN TO PARSER
5272
5273
5274  037142  012705  003504    ACTSHW: MOV     #SHTAB,R5
5275  037146  122571  000000    5$:     CMPB    (R5)+,@(R1)     ;LOOK AT FIRST BYTE OF MSG TO DECIPHER TYPE
5276  037152  001404                    BEQ     6$
5277  037154  020527  003513            CMP     R5,#SHTEND      ;SEE IF LOOKED AT ALL OF DEFAULTS YET
5278  037160  001372                    BNE     5$
5279  037162  005205                    INC     R5              ;MUST BE OPR. SPEC'D THEN
5280  037164  162705  003505    6$:     SUB     #SHTAB+1,R5
5281  037170  006305                    ASL     R5
5282  037172  016137  000002  007426    MOV     2(R1),TEMP
5283  037200                            PRINTF  #SHMSG,SHTYTB(R5),TEMP  ;PRINT MSG SIZE & TYPE
5284  037200  013746  007426                                                    MOV     TEMP,-(SP)
5285  037204  016546  003464                                                    MOV     SHTYTB(R5),-(SP)
5286  037210  012746  014434                                                    MOV     #SHMSG,-(SP)
5287  037214  012746  000003                                                    MOV     #3,-(SP)
5288  037220  010600                                                            MOV     SP,R0
5289  037222  104417                                                            TRAP    C$PNTF
5290  037224  062706  000010                                                    ADD     #10,SP
5291  037230  062701  000004            ADD     #4,R1           ;BUMP R1 TO NEXT SET OF POINTERS
5292  037234  005302                    DEC     R2
5293  037236  001341                    BNE     ACTSHW
5294  037240  013737  007470  010654    MOV     MODTYP,DEV1
5295  037246  013737  007472  010656    MOV     MLTYP,DEV2
5296  037254  013737  007500  010660    MOV     RPASS,DEV3
5297  037262  013737  007476  010662    MOV     PARAM,DEV4
5298  037270  004737  032340            JSR     PC,SHWOP        ;SHOW THE OPERATOR THE CURRENT MODE..... ALSO
5299  037274  000207                    RTS     PC
5300
5301  037276  013737  003554  007402    ACTDMS: MOV     P$NUM,STADD     ;SETUP STARTING ADDRESS FOR DUMP
5302  037304  005037  007406            CLR     BYTBIT          ;SET DEFAULT OF WORD DUMP
5303  037310  012737  000052  003202    MOV     #DMPS,KEYWD1    ;FLAG THAT A DUMP WAS TYPED
```

```
 5304  037316  000403                          BR      ACTDME
 5305
 5306  037320  012737  177777  007406  ACTDMQ: MOV     #-1,BYTBIT      ;SET DUMP FLAG TO "DUMP-WORD"
 5307  037326  013737  003554  007404  ACTDME: MOV     P$NUM,ENADD     ;SETUP END ADDRESS FOR DUMP (=START IF NO "EEE"
 5308  037334  105037  003560          ACTDMX: CLRB    P$NNUF          ;CLEAR NOT-ENOUGH FLAG, "DUMP N-N/B" IS VALID
 5309  037340  000207                          RTS     PC
 5310
```

```


037342  012737  000010  003202  ACTSTE: MOV     #SETEXP,KEYWD1
037350  000403                          BR      ACTSTX

037352  012737  000011  003202  ACTSTT: MOV     #SETTRN,KEYWD1
037360  012737  000001  003206  ACTSTX: MOV     #1,QUALVL       ;SET UP DEFAULT COPY TO 1 (/COPY=0)
037366  000207                          RTS     PC

037370  012737  000012  003204  ACTSZE: MOV     #SIZE,QUALFG
037376  000207                          RTS     PC

037400  012737  000013  003204  ACTCOP: MOV     #QCOPY,QUALFG
037406  000207                          RTS     PC

037410  023727  003204  000012  ACTNUM: CMP     QUALFG,#SIZE    ;SEE IF A SIZE OR COPY TYPED
037416  001023                          BNE     1$              ;BR IF IT WAS A COPY
037420  005737  003554                  TST     P$NUM           ;CHECK TO BE SURE DIDN'T TRY SIZE=0
037424  001014                          BNE     3$              ; BR IF NO
037426                                  PRINTF  #CLISEO
037426  012746  013005                                                                          MOV     #CLISEO,-(SP)
037432  012746  000001                                                                          MOV     #1,-(SP)
037436  010600                                                                                  MOV     SP,R0
037440  104417                                                                                  TRAP    C$PNTF
037442  062706  000004                                                                          ADD     #4,SP
037446  112737  177777  003561          MOVB    #-1,P$GDBD      ;SEE ERROR-IN-CMD FLAG
037454  000411                          BR      2$
037456  013737  003554  007412  3$:     MOV     P$NUM,CURCC     ;IF A SIZE LOAD CURCC WITH BYTE COUNT
037464  000405                          BR      2$
037466  013737  003554  003206  1$:     MOV     P$NUM,QUALVL    ;IF A COPY, LOAD COPY COUNT
037474  005237  003206                  INC     QUALVL          ;INCREMENT SO FIRST DEC MAKES IT REAL #
037500  000522                  2$:     BR      ACTMEX

037502  012737  000007  007410  ACTOPM: MOV     #7,MSGTYP
037510  010437  007426                  MOV     R4,TEMP         ;KEEP TRACK OF START OF QUOTED TEXT
037514  005237  007426                  INC     TEMP            ; SO CAN CALC OPCNT AT END OF QUOTES
037520  000207                          RTS     PC

037522  010402                  ACTEQO: MOV     R4,R2
037524  163702  007426                  SUB     TEMP,R2
037530  010237  007412                  MOV     R2,CURCC        ;CALC BYTE COUNT FOR QUOTED TEXT
037534  010237  002166                  MOV     R2,OPCNT
037540  013701  007426                  MOV     TEMP,R1
037544  012705  002524                  MOV     #OPBUF,R5
037550  112125                  1$:     MOVB    (R1)+,(R5)+     ;COPY QUOTED TEXT TO OPBUF
037552  005302                          DEC     R2
037554  001375                          BNE     1$
037556  000473                          BR      ACTMEX

037560                          ACTBCR: PRINTF  #CLIBCR         ;BAD CHAR. IN OPR. QUOTED STRING
037560  012746  012740                                                                          MOV     #CLIBCR,-(SP)
037564  012746  000001                                                                          MOV     #1,-(SP)
037570  010600                                                                                  MOV     SP,R0
037572  104417                                                                                  TRAP    C$PNTF
037574  062706  000004                                                                          ADD     #4,SP
037600  000207                          RTS     PC
```

```
 5367                                          ;SET THE MESSAGE TYPE AS PER COMMAND LINE
 5368  037602  005037  007410          ACTMS0: CLR     MSGTYP
 5369  037606  000435                  BR      ACTME1
 5370  037610  012737  000001  007410  ACTMS1: MOV     #1,MSGTYP       ;ALL ONES
 5371  037616  000431                  BR      ACTME1
 5372  037620  012737  000002  007410  ACTMS2: MOV     #2,MSGTYP       ;ONES & ZEROS
 5373  037626  000425                  BR      ACTME1
 5374  037630  012737  000003  007410  ACTMS3: MOV     #3,MSGTYP       ;ZEROS & ONES
 5375  037636  000421                  BR      ACTME1
 5376  037640  012737  000004  007410  ACTMS4: MOV     #4,MSGTYP       ;CCITT
 5377  037646  000415                  BR      ACTME1
 5378  037650  012737  000005  007410  ACTMS5: MOV     #5,MSGTYP       ;QUICK FOX
 5379  037656  013737  002162  007412          MOV     MSG5C,CURCC             ;SETUP DEFAULT SIZE FOR THIS TYPE
 5380  037664  000430                  BR      ACTMEX
 5381  037666  012737  000006  007410  ACTMS6: MOV     #6,MSGTYP       ;ALPHA/NUM
 5382  037674  013737  002164  007412          MOV     MSG6C,CURCC             ;SETUP DEFAULT SIZE FOR THIS TYPE
 5383
 5384  037702  012737  000100  007412  ACTME1: MOV     #64.,CURCC              ;SETUP DEFAULT SIZE FOR MSG0-4
 5385  037710  000416                  BR      ACTMEX                  ;GO TO EXIT
 5386
 5387                                          ;REV B BY EC
 5388  037712  022737  000010  003202  ACTSEX: CMP     #SETEXP,KEYWD1          ;DID WE GET HERE FROM "SET E =" COMMAND?
 5389  037720  001404                  BEQ     10$                     ;YES,BRANCH
 5390  037722  112737  177777  003561          MOVB    #-1,P$GDBD              ;SET ERROR FLAG
 5391  037730  000406                  BR      ACTMEX                  ;GO TO EXIT
 5392  037732  004737  032134          10$:    JSR     PC,FACSIMILE            ;GO COPY TRANMIT BUFFER TO EXPECT BUFFER
 5393  037736  012737  000060  003202          MOV     #SETET,KEYWD1           ;SET FLAG TO BE USED IN T1::
 5394  037744  000400                  BR      ACTMEX                  ;GO TO EXIT
 5395
 5396  037746  105037  003560          ACTMEX: CLRB    P$NNUF                  ;CLEAR NOT-ENOUGH FLAG
 5397  037752  000207                  RTS     PC
 5398
```

```
 5399  037754  012737  000003  007470  ACTATV: MOV     #ACT,MODTYP
 5400  037762  000432                  BR      ACTM2X
 5401
 5402  037764  012737  000002  007470  ACTPAS: MOV     #PAS,MODTYP
 5403  037772  105037  003560                  CLRB    P$NNUF                  ;CLEAR NOT-ENOUGH FLAG
 5404  037776  005037  007472                  CLR     MLTYP                   ;CLEAR MAINT LOOP TYPE
 5405  040002  000207                  RTS     PC
 5406
 5407  040004  005037  007470          ACTREC: CLR     MODTYP
 5408  040010  000417                  BR      ACTM2X
 5409
 5410  040012  012737  000006  007470  ACTLIS: MOV     #LIS,MODTYP
 5411  040020  000413                  BR      ACTM2X
 5412
 5413  040022  012737  000004  007470  ACTDLL: MOV     #DOW,MODTYP
 5414  040030  000407                  BR      ACTM2X
 5415
 5416  040032  012737  000001  007470  ACTTRA: MOV     #TRA,MODTYP
 5417  040040  000403                  BR      ACTM2X
 5418
 5419  040042  012737  000005  007470  ACTTAL: MOV     #TAL,MODTYP
 5420
 5421  040050  042737  000004  007476  ACTM2X: BIC     #ECHOB,PARAM            ;DISABLE /ECHO (ALL BUT PASSIVE MODE)
 5422  040056  105037  003560                  CLRB    P$NNUF                  ;CLEAR NOT-ENOUGH FLAG
 5423  040062  005037  007472                  CLR     MLTYP                   ;CLEAR MAINT LOOP TYPE
 5424  040066  000207                  RTS     PC
 5425
```

```
 5426  040070  012737  000036  003204  ACTNO:  MOV     #NO,QUALFG
 5427  040076  000207                  RTS     PC
 5428
 5429  040100  022737  000036  003204  ACTECH: CMP     #NO,QUALFG
 5430  040106  001422                  BEQ     1$
 5431  040110  052737  000004  007476          BIS     #ECHOB,PARAM
 5432  040116  022737  000002  007470          CMP     #PAS,MODTYP             ;BE SURE IN PASSIVE MODE IF
 5433  040124  001416                  BEQ     2$                      ;IF TRYING TO SET /ECHO
 5434  040126                          PRINTF  #CLINPS
 5435  040126  012746  012675                                                          MOV     #CLINPS,-(SP)
 5436  040132  012746  000001                                                          MOV     #1,-(SP)
 5437  040136  010600                                                                  MOV     SP,R0
 5438  040140  104417                                                                  TRAP    C$PNTF
 5439  040142  062706  000004                                                          ADD     #4,SP
 5440  040146  112737  177777  003561          MOVB    #-1,P$GDBD
 5441  040154  042737  000004  007476  1$:     BIC     #ECHOB,PARAM
 5442  040162  005037  003204          2$:     CLR     QUALFG                  ;CLEAR 'NO'' OUT OF QUALIFIER FLAG
 5443  040166  000501                  BR      ACTLXX
 5444
 5445  040170  012701  000002          ACTCHK: MOV     #DATCKB,R1              ;SET DATA CHECK BIT
 5446  040174  000413                  BR      ACTQFG
 5447
 5448  040176  012701  000001          ACTSTS: MOV     #STATB,R1       ;SET THE STATUS BIT
 5449  040202  000410                  BR      ACTQFG
 5450
 5451  040204  012701  000020          ACTCRC: MOV     #CRCB,R1        ;SET THE CRC BIT
 5452  040210  000405                  BR      ACTQFG
 5453
 5454  040212  012701  000010          ACTMOS: MOV     #MOCHK,R1       ;MODEM BIT      ADDED BY EC
 5455  040216  000402                  BR      ACTQFG
 5456
 5457  040220  012701  000040          ACTPRO: MOV     #PROTOB,R1      ;SET THE PROTOCOL BIT
 5458
 5459  040224  050137  007476          ACTQFG: BIS     R1,PARAM
 5460  040230  022737  000036  003204          CMP     #NO,QUALFG
 5461  040236  001002                  BNE     1$
 5462  040240  040137  007476          BIC     R1,PARAM
 5463  040244  005037  003204          1$:     CLR     QUALFG                  ;CLEAR 'NO'' OUT OF QUALIFIER FLAG
 5464  040250  000450                  BR      ACTLXX
 5465
 5466  040252  013737  003554  007500  ACTRPS: MOV     P$NUM,RPASS             ;GET NUMBER OF 'RUN PASSES''
 5467  040260  000444                  BR      ACTLXX
 5468
 5469  040262  012737  000005  007472  ACTMOP: MOV     #5,MLTYP
 5470  040270  000417                  BR      ACTLPX
 5471  040272  012737  000001  007472  ACTTLP: MOV     #1,MLTYP
 5472  040300  000413                  BR      ACTLPX
 5473  040302  012737  000002  007472  ACTCLP: MOV     #2,MLTYP
 5474  040310  000407                  BR      ACTLPX
 5475  040312  012737  000003  007472  ACTLLP: MOV     #3,MLTYP
 5476  040320  000403                  BR      ACTLPX
 5477  040322  012737  000004  007472  ACTRLP: MOV     #4,MLTYP
 5478
 5479  040330  022737  000003  007470  ACTLPX: CMP     #ACT,MODTYP     ;BE SURE IN ACTIVE IF TRYING TO SET LOOP
 5480  040336  001415                  BEQ     ACTLXX          ; BR IF IN ACTIVE
 5481  040340  112737  177777  003561          MOVB    #-1,P$GDBD
```

```
 5482  040346  005037  007472                CLR     MLTYP           ;CLEAR ANY LOOP TYPE THAT MAY HAVE GOT SET
 5483  040352                                PRINTF  #CLIBDL
 5484  040352  012746  012633                                                MOV     #CLIBDL,-(SP)
 5485  040356  012746  000001                                                MOV     #1,-(SP)
 5486  040362  010600                                                        MOV     SP,R0
 5487  040364  104417                                                        TRAP    C$PNTF
 5488  040366  062706  000004                                                ADD     #4,SP
 5489  040372  105037  003560          ACTLXX: CLRB    P$NNUF          ;CLEAR NOT-ENOUGH FLAG
 5490  040376  000207                        RTS     PC
 5491
```

```
5492
5493                                        ;REV B BY EC
5494  040400  005737  007472        GTR9:   TST     MLTYP           ;LOOP MODE ?
5495  040404  001422                        BEQ     10$             ;NO,BRANCH
5496  040406  032737  000002  007476        BIT     #DATCKB,PARAM   ;DATA CHECK ?
5497  040414  001416                        BEQ     10$             ;NO,BRANCH
5498  040416  023737  007340  007354        CMP     CMPTOT,TXMTOT   ;TX & EX EQUAL
5499  040424  001412                        BEQ     10$             ;YES,BRANCH
5500  040426                                PRINTF  #CLIPW          ;PRINT WARNING
5501  040426  012746  013036                                                        MOV     #CLIPW,-(SP)
5502  040432  012746  000001                                                        MOV     #1,-(SP)
5503  040436  010600                                                                MOV     SP,R0
5504  040440  104417                                                                TRAP    C$PNTF
5505  040442  062706  000004                                                        ADD     #4,SP
5506  040446  000137  035354                JMP     GETCL           ;TRY AGAIN
5507
5508                                        ;RX ALLOCATE CODE
5509  040452  012701  006562        10$:    MOV     #PTRTAB,R1      ;INIT TRANSMIT MESSAGE POINTER
5510  040456  010137  007334                MOV     R1,TXPTR
5511  040462  012702  000017                MOV     #MSGLIM,R2
5512  040466  006302                        ASL     R2
5513  040470  006302                        ASL     R2
5514  040472  010137  007336                MOV     R1,CMPPTR
5515  040476  060237  007336                ADD     R2,CMPPTR       ;INIT COMPARE MESSAGE POINTER
5516  040502  013701  007336                MOV     CMPPTR,R1
5517  040506  012702  000017                MOV     #MSGLIM,R2
5518  040512  006302                        ASL     R2
5519  040514  006302                        ASL     R2
5520  040516  010137  007332                MOV     R1,RXPTR
5521  040522  060237  007332                ADD     R2,RXPTR        ;INIT RECEIVE MESSAGE POINTER
5522
5523  040526  013737  007340  007370        MOV     CMPTOT,RXMTOT   ;MAKE COMPARE AND RX MESSAGE COUNTS EQUAL
5524
5525
5526  040534  005037  007502        GTREX:  CLR     FLAG            ;CLEAR FLAG
5527  040540  005037  007374                CLR     NOBUF           ;CLEAR NO BUFFER COUNTER
5528  040544  005037  007376                CLR     PSCNT           ;CLEAR PASS COUNT
5529  040550  005037  007400                CLR     ERRCNT          ;CLEAR ERROR COUNT
5530  040554  005037  007372                CLR     LNCNT           ;CLEAR LINE COUNTER
5531
5532  040560  004737  026452                JSR     PC,LOGDVI       ;LOG ABOUT TO INIT DEVICE
5533  040564  004737  043364                JSR     PC,DVINIT       ;INIT DEVICE
5534
5535  040570  012737  001000  007412  GTRX2:  MOV     #BUFLIM,CURCC   ;SET CHAR COUNT TO 'BUFLIM'' NO. OF BYTES
5536  040576  012737  004562  007420        MOV     #RXBUF,CURADD   ;SET UP RX BUFFER AS CURRRENT ADD.
5537  040604  013737  007332  007416        MOV     RXPTR,CPTR
5538  040612  012737  000010  007410        MOV     #10,MSGTYP      ;SET UP FOR 33 TO FILL RX BUFFERS
5539  040620  004737  032010                JSR     PC,BLDBUF       ;CLEAR RX BUFFER
5540  040624  013702  007470                MOV     MODTYP,R2
5541  040630  006302                        ASL     R2
5542  040632  000172  007504                JMP     @MODE(R2)       ;MODE DISPATCH
5543
```

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

040636                                  RXONLY:
040636  013737  007332  007414          RXON2:  MOV     RXPTR,CPTRR
040644  013737  007370  007366                  MOV     RXMTOT,DVRCT    ;SET UP MESSAGE COUNT
040652  052737  000104  007502                  BIS     #QRX+#ERX,FLAG  ;SET UP RX QUE
040660  005037  007416                          CLR     CPTR            ;CLEAR THE TX POINTER
040664  000137  041026                          JMP     ALLTR           ;GO RX.
```

```
 5567                                   .SBTTL          TRANSMIT MODE SECTION
 5568
 5569                                   ;++
 5570                                   ; FUNCTIONAL DESCRIPTION:
 5571                                   ;       TRANSMIT-ONLY (OR ONE-WAY-OUT) ROUTINE
 5572                                   ;       IN THIS MODE OF TESTING A LIST OF MESSAGES IS TRANSMITTED WITHOUT
 5573                                   ;       EXPECTING ANY DATA TO BE RECEIVED.  A REPETITION COUNT CAN BE
 5574                                   ;       SPECIFIED TO REPETITIVELY TRANSMIT THE LIST.
 5575                                   ;
 5576                                   ; SUBORDINATE ROUTINES USED:
 5577                                   ;               "ALLTR"
 5578                                   ;
 5579                                   ; CALLING SEQUENCE:
 5580                                   ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
 5581                                   ;--
 5582
 5583  040670  042737  000002  007476  TXONLY: BIC     #DATCKB,PARAM   ;SET NOCHECK
 5584  040676  013737  007334  007416  TXON2:  MOV     TXPTR,CPTR
 5585  040704  013737  007354  007352           MOV     TXMTOT,DVTCT    ;COPY COUNTER FOR THIS PASS
 5586  040712  052737  000210  007502           BIS     #QTX+#ETX,FLAG  ;SET THE QUE TX FLAG
 5587  040720  005037  007414                   CLR     CPTRR           ;CLEAR RX POINTER
 5588  040724  000137  041026                   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
                                ;--

040730                          PLCK:
040730  013737  007354  007352 PLCK2:  MOV     TXMTOT,DVTCT    ;SET UP  THE TRANSMIT COUNT
040736  013737  007334  007416         MOV     TXPTR,CPTR      ;SET UP CPTR TO TRANSMIT POINTER
040744  013737  007332  007414 PLCK3:  MOV     RXPTR,CPTRR     ;SET UP CPTRR TO REC POINTER
040752  052737  000104  007502         BIS     #QRX+#ERX,FLAG  ;SET UP Q AND EXPECT RX
040760  000137  041026                 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
                                ;--

040764  013737  007354  007352  ALCK:   MOV     TXMTOT,DVTCT
040772  013737  007334  007416          MOV     TXPTR,CPTR      ;SET UP TX COUNTS
041000  013737  007370  007366          MOV     RXMTOT,DVRCT    ;SET UP COUNTS
041006  013737  007332  007414          MOV     RXPTR,CPTRR
041014  052737  000314  007502          BIS     #QRX+#QTX+#ETX+#ERX,FLAG
041022  000137  041026                  JMP     ALLTR
```

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

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

                                ALLTR:
041026  032737  000004  007502  ALCK5:  BIT     #QRX,FLAG
041034  001420                  BEQ     ALCK1           ;IF NOT RX GO TO TX'S
041036  013702  007414          MOV     CPTRR,R2
041042  011237  007432          MOV     (R2),TEMP2
041046  012237  007362          MOV     (R2)+,DVRXA
041052  011237  007434          MOV     (R2),TEMP3
041056  011237  007364          MOV     (R2),DVRCC
041062  010237  007414          MOV     R2,CPTRR
041066  004737  044104          JSR     PC,DVRXQ        ;GO QUE DEVICE
041072  004737  026406          JSR     PC,LOGRXQ       ;LOG REC QUED
041076  032737  000010  007502  ALCK1:  BIT     #QTX,FLAG
```

```
 5698  041104  001416                          BEQ     ALCK2           ;IF NO TX'S GO TO 2
 5699  041106  013702  007416                  MOV     CPTR,R2
 5700  041112  011237  007432                  MOV     (R2),TEMP2
 5701  041116  012237  007346                  MOV     (R2)+,DVTXA
 5702  041122  011237  007434                  MOV     (R2),TEMP3
 5703  041126  012237  007350                  MOV     (R2)+,DVTCC
 5704  041132  010237  007416                  MOV     R2,CPTR
 5705  041136  004737  026352                  JSR     PC,LOGTXQ
 5706
 5707  041142  004737  044164          ALCK2:  JSR     PC,DVTXRX       ;GO TO TX AND RX SUB ROUT.
 5708
 5709  041146  032737  000004  007502          BIT     #QRX,FLAG       ;CHECK FOR REC. MSG.
 5710  041154  001514                          BEQ     ALCK3
 5711  041156  013737  007362  007432          MOV     DVRXA,TEMP2
 5712  041164  013737  007364  007434          MOV     DVRCC,TEMP3
 5713  041172  004737  026424                  JSR     PC,LOGRXC       ;LOG REC COMPLETE
 5714  041176  032737  000004  007476  UPTABL: BIT     #ECHOB,PARAM    ;IS THIS ECHO MODE(PASSIVE)
 5715  041204  001406                          BEQ     UPTA4           ;IF NOT GO TO 4
 5716  041206  013702  007416                  MOV     CPTR,R2         ;ELSE SET R2 TO PRESENT TX TABL
 5717  041212  013722  007432                  MOV     TEMP2,(R2)+     ;STORE OFF RX ADD
 5718  041216  013712  007434                  MOV     TEMP3,(R2)      ;AND CC
 5719  041222  032737  000002  007476  UPTA4:  BIT     #DATCKB,PARAM   ;DATA CHECK?
 5720  041230  001015                          BNE     UPTA1           ;YES,BRANCH
 5721  041232  012737  000001  007366          MOV     #01,DVRCT       ;ELSE SET DVRCT TO A 1
 5722  041240  013737  007332  007414          MOV     RXPTR,CPTRR     ;RESET POINTER
 5723  041246  022737  000003  007470          CMP     #ACT,MODTYP     ;IS THIS ACTIVE
 5724  041254  001002                          BNE     UPTA3
 5725  041256  005237  007366                  INC     DVRCT           ;IF YES BUMP COUNT
 5726  041262  000424                  UPTA3:  BR      UPTEX
 5727  041264  013702  007414          UPTA1:  MOV     CPTRR,R2
 5728  041270  011237  007426                  MOV     (R2),TEMP       ;LOAD TEMP WITH PREV. COUNT
 5729  041274  163737  007434  007426          SUB     TEMP3,TEMP      ;LOAD TEMP WITH PREV.COUNT-CURRENT
 5730  041302  013722  007434                  MOV     TEMP3,(R2)+
 5731  041306  063737  007434  007432          ADD     TEMP3,TEMP2
 5732  041314  013722  007432                  MOV     TEMP2,(R2)+     ;STORE OF NEW ADD
 5733  041320  013712  007426                  MOV     TEMP,(R2)       ;AND NEW CC
 5734  041324  162702  000002                  SUB     #2,R2           ;PUT POINTER BACK TO ADDR.
 5735  041330  010237  007414                  MOV     R2,CPTRR        ;AND RESTORE IT.
 5736  041334                          UPTEX:
 5737  041334  022737  000002  007470          CMP     #PAS,MODTYP
 5738  041342  001007                          BNE     ALCK2A          ;IF NOT PASSIVE LOOP THEN GO TO 2A
 5739  041344  042737  000104  007502          BIC     #QRX+#ERX,FLAG  ;CLEAR BOTH EXPECTED AND COMPLETED FLAGS
 5740  041352  052737  000210  007502          BIS     #QTX+#ETX,FLAG  ;SET THE TX FLAGS
 5741  041360  000646                          BR      ALCK1
 5742
 5743  041362  005337  007366          ALCK2A: DEC     DVRCT           ;DEC REC COUNT
 5744  041366  005737  007366                  TST     DVRCT           ;IS IT ALL DONE
 5745  041372  001005                          BNE     ALCK3           ;NO. GO CHECK TX
 5746  041374  042737  000004  007502          BIC     #QRX,FLAG       ;CLEAR THE RX FLAG
 5747  041402  005037  007414                  CLR     CPTRR           ;YES. CLEAR POINTER
 5748  041406  032737  000010  007502  ALCK3:  BIT     #QTX,FLAG       ;IS IT TX
 5749  041414  001447                          BEQ     ALCK4           ;IF NOT TX THEN GO BACK
 5750  041416  013737  007346  007432          MOV     DVTXA,TEMP2
 5751  041424  013737  007350  007434          MOV     DVTCC,TEMP3     ;LOG TX COMPLETED
 5752  041432  004737  026370                  JSR     PC,LOGTXC
 5753  041436  005337  007352                  DEC     DVTCT           ;DEC TX COUNT
```

```
 5754  041442  022737  000002  007470            CMP     #PAS,MODTYP
 5755  041450  001013                            BNE     ALCK3A          ;IF NOT PASSIVE MODE GO TO 3A
 5756  041452  042737  000210  007502            BIC     #QTX+ETX,FLAG   ;CLEAR THE TX FLAGS
 5757  041460  052737  000104  007502            BIS     #QRX+ERX,FLAG   ;AND SET THE RX FLAGS
 5758  041466  005737  007352                    TST     DVTCT
 5759  041472  001005                            BNE     ALCK3C          ;IF MORE RX'S DO IT
 5760  041474  000137  041554                    JMP     CMPSR           ; ELSE COMPARE
 5761  041500  005737  007352          ALCK3A:   TST     DVTCT           ;IS IT ALL DONE
 5762  041504  001402                            BEQ     ALCK3B          ;IF NOT GO BACK TO 5
 5763  041506  000137  041026          ALCK3C:   JMP     ALCK5
 5764  041512  005037  007416          ALCK3B:   CLR     CPTR            ;IF SO CLEAR POINTER
 5765  041516  042737  000010  007502            BIC     #QTX,FLAG       ;CLEAR TX FLAG
 5766  041524  032737  000002  007476            BIT     #DATCKB,PARAM   ;IS IT DAT CK
 5767  041532  001403                            BEQ     ALCK4A          ;IF NOT THEN END WO CKING RX.
 5768  041534  005737  007414          ALCK4:    TST     CPTRR
 5769
 5770  041540  001362                            BNE     ALCK3C          ;IF SOME RX''S LEFT GO BACK
 5771  041542  005737  007416          ALCK4A:   TST     CPTR
 5772  041546  001402                            BEQ     ALCK4B          ;BRANCH IF ANY TX'S LEFT
 5773  041550  000137  041142                    JMP     ALCK2
 5774  041554                          ALCK4B:
 5775
 5776
 5777
```

```
5778                                    .SBTTL          DATA COMPARISON CODE
5779
5780
5781                                    ;++
5782                                    ; FUNCTIONAL DESCRIPTION:
5783                                    ;
5784                                    ;       CMPSR - COMPARE CODE
5785                                    ;       THIS CODE COMPARES THE RECEIVED DATA AGAINST THE
5786                                    ;       EXPECTED AND FILLS THE EVENT LOG WITH 1 OF 3 MSGS.
5787                                    ;
5788                                    ;       NOTE: IF NO DATA CHECKING SKIP THIS CODE
5789                                    ;
5790                                    ;       1.) A DATA COMPARISON ENTRY WHICH REPORTS THE NUMBER
5791                                    ;           OF COMPARISON ERRORS FOUND.
5792                                    ;       2.) A DATA COMPARISON ENTRY WHICH REPORTS DIFFERENCES
5793                                    ;           IN REC LENGTH TO COMPARE LENGTH.
5794                                    ;       3.) A DATA COMPARISON STARTED ENTRY WHICH REPORTS ADDRESS
5795                                    ;           OF RECEIVE BUFFER AND BYTE COUNT.
5796                                    ;       THIS CODE ALSO REPORTS SOFT ERRORS FOR DATA COMPARISON
5797                                    ;       (THE FIRST 5 ONLY),LENGTH ERROR,AND TOTAL NUMBER OF ERRORS
5798                                    ;
5799                                    ;
5800                                    ; SUBORDINATE ROUTINES USED:
5801                                    ;
5802                                    ;               "LOGCMP" - SEE ITEM 3 ABOVE
5803                                    ;               "LOGCML" - SEE ITEM 2 ABOVE
5804                                    ;               "LOGCMD" - SEE ITEM 1 ABOVE
5805                                    ;
5806                                    ; CALLING SEQUENCE:
5807                                    ;       JMP     CMPSR           ;JUMP TO DATA COMPARISON CODE
5808                                    ;--
5809
5810  041554  032737  000002  007476  CMPSR:  BIT     #DATCKB,PARAM   ;IS DATA CHECKING TO BE DONE
5811  041562  001522                          BEQ     CMPSEX          ;IF NOT THEN EXIT
5812  041564  013737  007332  007416          MOV     RXPTR,CPTR      ;PUT START OF RX POINTERS TO CPTR
5813  041572  013737  007336  007414          MOV     CMPPTR,CPTRR    ; AND START OF COMPARE POINTS TO CPTRR
5814  041600  013737  007370  007366          MOV     RXMTOT,DVRCT
5815
5816  041606                          CMPS3:
5817  041606  013702  007416                  MOV     CPTR,R2         ;MOVE CURRET RX PT.TO R2
5818  041612  011237  007432                  MOV     (R2),TEMP2      ;MOVE RX ADD TO EVENT LOG
5819  041616  012201                          MOV     (R2)+,R1        ;SET R1 TO START ADD OF RX
5820  041620  012237  007434                  MOV     (R2)+,TEMP3     ;SET CHAR  COUNT TO EVENT LOG
5821  041624  010237  007416                  MOV     R2,CPTR         ;RESTORE RX POINT
5822
5823  041630  013702  007414                  MOV     CPTRR,R2        ;PUT R2 AT COMPARE TABLE
5824  041634  012203                          MOV     (R2)+,R3        ;SET R3 TO COMPARE ADD
5825  041636  012204                          MOV     (R2)+,R4        ;SET R4 TO COMP CC
5826  041640  010237  007414                  MOV     R2,CPTRR        ;RESTORE POINTER
5827  041644  010437  007436                  MOV     R4,TEMP4
5828  041650  004737  026520                  JSR     PC,LOGCMP       ;LOG COMPARE START.
5829
5830  041654  020437  007434                  CMP     R4,TEMP3        ;IS COMPARE COUNT = TO RX COUNT
5831  041660  001410                          BEQ     CMPS7           ;IF SO GO TO 7
5832  041662  005237  007400                  INC     ERRCNT
5833  041666                                  ERRSOFT 1,EDDLE,ERR10   ;PRINT ERROR
```

```
                                                                                    TRAP    C$ERSOFT
                                                                                    .WORD   1
                                                                                    .WORD   EDDLE
                                                                                    .WORD   ERR10
                        JSR     PC,LOGCML       ;LOG LENGTH ERROR

CMPS7:  CLR     TEMP4           ;CLEAR BAD BYTE COUNTER
        MOV     #1,OFSET        ;SET OFSET BYTE COUNT TO 1
CMPS1:  CMPB    (R1)+,(R3)+     ;COMPARE RX WITH EXPETED
        BEQ     CMPS6           ;IF EQUAL THEN GO TO 6

CMPS2:  INC     TEMP4           ;INC BAD COUNT
        CMP     TEMP4,#5        ;IS IT MORE THEN 5
        BHI     CMPS6           ;IF SO GO  FOR MORE
        MOVB    -(R3),GOOD      ;STORE GOOD BYTE FOR ERROR
        MOVB    -(R1),BAD       ;STORE BAD BYTE FOR ERROR
        INC     ERRCNT
        ERRSOFT 2,EDDDE,ERR1    ;REPORT COMPARISON FAILURE TO OPR.
                                                                                    TRAP    C$ERSOFT
                                                                                    .WORD   2
                                                                                    .WORD   EDDDE
                                                                                    .WORD   ERR1
        INC     R1
        INC     R3
CMPS6:  INC     OFSET           ;INC OFFSET
        DEC     R4              ;ELSE DEC CHAR COUNT AND SEE IF 0
        BNE     CMPS1           ;IF NOT GO BACK
        TST     TEMP4           ;SEE IF ANY CMP ERRS FOR THIS MSG
        BEQ     CMPS5A          ;BR IF NONE
        INC     ERRCNT
        ERRSOFT 3,EDDDE,ERR2    ;REPORT # OF MISMATCHES FOR MESSAGE
                                                                                    TRAP    C$ERSOFT
                                                                                    .WORD   3
                                                                                    .WORD   EDDDE
                                                                                    .WORD   ERR2
CMPS5:  JSR     PC,LOGCMD       ;LOG DATA ERROR IN COMPARE
CMPS5A:
        DEC     DVRCT
        BNE     CMPS3           ;IF NOT ALL DONE GO BACK
```

```
                                .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
                                ;------

042030  005237  007376          CMPSEX: INC     PSCNT           ;BUMP PASS COUNT

042034  013737  007374  007436          MOV     NOBUF,TEMP4
042042  013737  007376  007432          MOV     PSCNT,TEMP2
042050  013737  007400  007434          MOV     ERRCNT,TEMP3
042056  004737  026572                  JSR     PC,LOGEOP       ;LOG END OF PASS
042062  022737  177777  007500  5$:     CMP     #-1,RPASS       ;SEE IF RPASS=-1
042070  001403                          BEQ     1$              ;IF IT IS DON'T DECRMNT, LOOP FOREVER
042072  005337  007500                  DEC     RPASS           ;DEC PASS COUNT
042076  001402                          BEQ     2$              ;IF DONE EXIT TEST
042100  000137  040570          1$:     JMP     GTRX2           ;ELSE GO BACK AND DISPATCH
042104  004737  045774          2$:     JSR     PC,DVBTUP       ;GO UPDATE BASE TABLE
042110  000137  035270                  JMP     GTRA5           ;WHEN RPASS=0 GO BACK TO "DCLT>"
```

```
                                .SBTTL          DOWN-LINE-LOAD SECTION

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       DOWN-LINE-LOAD SECTION
                                ;       IN THIS MODE OF TESTING THE "HOST" OR ORIGINATING STATION
                                ;       REQUESTS THE "SATELLITE" OR BOOT STATION TO ENTER MOP MODE.
                                ;       THE BOOT STATION THEN SENDS A  "REQUEST PROGRAM MESSAGE".
                                ;       THE "HOST" THEN SENDS A "MEMORY LOAD WITH TRANSFER ADDRESS"
                                ;       THAT CONTAINS IMAGE DATA TO BE LOADED BY THE BOOT STATION'S
                                ;       M9312 STARTING AT LOC. 0.  THIS IMAGE DATA WILL CONTAIN A
                                ;       PROGRAM THAT WILL PRINT A MSG THAT DOWN-LINE-LOAD WAS SUCESSFUL.
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;               "DLTXRX" - SPECIAL TX RX ROUTINE FOR DLL
                                ;               "DVRXQ"  - QUE RX BUFFER SPACE TO DEVICE
                                ;               "LOGRXQ" - LOG RX SPACE QUED TO EVENT LOG
                                ;               "LOGTXQ" - LOG TX BUFFER QUED TO EVENT LOG
                                ;               "DVTXRX" - QUE TX BUFFER AND WAIT FOR RX OR TX TO COMPLETE
                                ;               "LOGTXC" - LOG TX COMPLETED TO EVENT LOG
                                ;               "LOGRXC" - LOG RX COMPLETED TO EVENT LOG
                                ;
                                ; CALLING SEQUENCE:
                                ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
                                ;--

042114                          DLL:    GMANID  DLLQ1,TEMP3,0,377,0,377,NO      ;GET PASSWORD
042114  104443                                                                         TRAP    C$GMAN
042116  000406                                                                         BR      10001$
042120  007434                                                                         .WORD   TEMP3
042122  000022                                                                         .WORD   T$CODE
042124  013126                                                                         .WORD   DLLQ1
042126  000377                                                                         .WORD   377
042130  000000                                                                         .WORD   T$LOLIM
042132  000377                                                                         .WORD   T$HILIM
042134                                                                          10001$:
042134  113737  007434  002650              MOVB    TEMP3,PASS1                     ;PUT PASSWORD IN MESSAGE
042142  113737  007434  002651              MOVB    TEMP3,PASS2                     ;PASSWORD IS DUPLICATE
042150  113737  007434  002652              MOVB    TEMP3,PASS3                     ;;HERE
042156  113737  007434  002653              MOVB    TEMP3,PASS4                     ;;;AND HERE.
042164  052737  000100  007502              BIS     #ERX,FLAG       ;SET EXPECTED TO RX
042172  042737  000002  007476              BIC     #DATCKB,PARAM   ;CLEAR NOCHECK
042200  012737  002647  007420              MOV     #DLLM1,CURADD   ;SET THE DOWN LINE LOAD MSG TO #1
042206  013737  002172  007412              MOV     DLLM1C,CURCC    ;SET THE CC
042214  004737  042306                      JSR     PC,DLTXRX       ;GO TO THE DOWN LINE TX RX ROUTINE

                                ;RETURN WHEN TX AND RX ARE COMPLETED

042220  012737  002654  007420              MOV     #DLLM2,CURADD   ;SET THE DOWN LINE LOAD MSG TO #2
042226  013737  002174  007412              MOV     DLLM2C,CURCC    ;SET CC
042234  042737  000400  007502              BIC     #DLLGA,FLAG     ;CLEAR THE GO AHEAD FLAG
042242  004737  042306                      JSR     PC,DLTXRX       ;GO TO THE DOWN LINE TX RX ROUTINE

                                ; RETURN WHEN TX AND RX ARE COMPLETED
042246                          DLLPRI:
```

```
042246                                          PRINTF  #DLLCM
042246  012746  015034                                                                  MOV     #DLLCM,-(SP)
042252  012746  000001                                                                  MOV     #1,-(SP)
042256  010600                                                                          MOV     SP,R0
042260  104417                                                                          TRAP    C$PNTF
042262  062706  000004                                                                  ADD     #4,SP
042266  000137  035270                          JMP     GTRA5

042272                                  DLLEA:
042272                                          ERRHRD  20,DLLAB,ERR14
042272  104456                                                                          TRAP    C$ERHRD
042274  000024                                                                          .WORD   20
042276  025214                                                                          .WORD   DLLAB
042300  026142                                                                          .WORD   ERR14

042302  000137  035270                          JMP     GTRA5           ;PRINT ABORT AND EXIT



042306                                  DLTXRX:
042306  052737  000004  007502                  BIS     #QRX,FLAG       ;SET THE QUE RX FLAG
042314  012737  004562  007362                  MOV     #RXBUF,DVRXA    ;SET THE DEVICE RX BUFFER TO RXBUF
042322  012737  004562  007432                  MOV     #RXBUF,TEMP2    ;SET UP FOR LOG
042330  012737  000400  007364                  MOV     #256.,DVRCC     ;SET UP FOR CC OF 256
042336  012737  000400  007434                  MOV     #256.,TEMP3     ;SET UP FOR LOG
042344  004737  044104                          JSR     PC,DVRXQ        ; GO QUE RX
042350  004737  026406                          JSR     PC,LOGRXQ       ;AND LOG IT...

042354  013737  007420  007346                  MOV     CURADD,DVTXA    ;SET UP FOR TX
042362  013737  007420  007432                  MOV     CURADD,TEMP2    ;AND LOG
042370  013737  007412  007350                  MOV     CURCC,DVTCC     ;SE UP FOR TX COUNT
042376  013737  007412  007434                  MOV     CURCC,TEMP3     ;AND LOG IT
042404  004737  026352                          JSR     PC,LOGTXQ       ;LOG THE TX QUEUED
042410  052737  000210  007502                  BIS     #QTX+#ETX,FLAG  ;SET UP TO QUE AND EXPECTED
042416  004737  044164                  DLLE2:  JSR     PC,DVTXRX       ;GO TO DEVICE ROUTINE
042422  032737  000400  007502                  BIT     #DLLGA,FLAG     ;TEST FOR GO AHEAD BIT
042430  001047                                  BNE     DLLE1           ;IF SET GO TO ONE
042432  032737  000010  007502                  BIT     #QTX,FLAG       ;ELSE CHECK FOR TX DONE
042440  001020                                  BNE     DLLE6           ;IF DONE THEN BRANCH
                                                                        ;ELSE ERROR
042442  012737  025521  007442                  MOV     #TXNC,CONOTM
042450  013737  004562  007434          DLLE7:  MOV     RXBUF,TEMP3
042456  013737  003562  007436                  MOV     TXBUF,TEMP4
042464  012737  025214  007432                  MOV     #DLLAB,TEMP2
042472  004737  026434                          JSR     PC,LGDVE        ;LOG ERROR
042476  000137  042272                          JMP     DLLEA           ;ABORT TEST

042502  013737  007346  007432          DLLE6:  MOV     DVTXA,TEMP2
042510  013737  007350  007434                  MOV     DVTCC,TEMP3
042516  004737  026370                          JSR     PC,LOGTXC       ;LOG TX DONE
042522  042737  000210  007502                  BIC     #QTX+#ETX,FLAG  ;CLEAR QUE AND EXPECTED
042530  052737  000400  007502                  BIS     #DLLGA,FLAG     ;SET THE GO AHEAD BIT
042536  023737  002174  007350                  CMP     DLLM2C,DVTCC
042544  001472                                  BEQ     DLLE5           ;EXIT IF SECOND MSG.
042546  000723                                  BR      DLLE2           ;AND GO BACK TO 2
042550  032737  000004  007502          DLLE1:  BIT     #QRX,FLAG       ;IS THE A RX COMPLETED
```

```
6017  042556  001004                        BNE     DLLE8           ;IF SO GO TO 8
6018  042560  012737  025541  007442        MOV     #RXNC,CONOTM    ;ELSE SET UP ERROR AND ABORT.
6019  042566  000730                        BR      DLLE7
6020  042570  013737  007362  007432  DLLE8: MOV     DVRXA,TEMP2
6021  042576  013737  007364  007434        MOV     DVRCC,TEMP3
6022  042604  004737  026424                JSR     PC,LOGRXC       ;LOG RECEIVE COMPLETE
6023  042610  122737  000010  004562        CMPB    #10,RXBUF       ;CHECK FOR FIRST WORD OF RX
6024                                                                ;SEC BOOT MSG.
6025  042616  001404                        BEQ     DLLE3
6026  042620  012737  025561  007442  DLLE4: MOV     #RXM1,CONOTM    ;SET UP MESG AND ABORT
6027  042626  000710                        BR      DLLE7           ;ABORT TEST
6028
6029  042630  122737  000001  004564  DLLE3: CMPB    #1,RXBUF+2      ;IS SECOND WORD 1?
6030  042636  001404                        BEQ     DLLE5A          ;YES,BRANCH
6031  042640  012737  025604  007442        MOV     #RXM2,CONOTM
6032  042646  000700                        BR      DLLE7           ;SET UP MESSAGE AND ABORT
6033
6034                                ;;PRINT ID OF DEVICE REQUESTING LOAD    REV B BY EC
6035  042650  012737  020273  007426  DLLE5A: MOV    #UNKM,TEMP      ;SET UP FOR UNKNOWN DEVICE
6036  042656  113703  004563                MOVB    RXBUF+1,R3      ;GET DEVTYPE FROM MESSAGE
6037  042662  120327  000042                CMPB    R3,#34.         ;OUT OF LEGAL RANGE ?
6038  042666  101006                        BHI     DLLE5B          ;YES,BRANCH
6039  042670  132703  000001                BITB    #1,R3           ;ODD ?
6040  042674  001003                        BNE     DLLE5B          ;YES,BRANCH
6041  042676  016337  010544  007426        MOV     DLLIND(R3),TEMP ;GET ASCIZ MESSAGE FROM TABLE
6042
6043  042704                          DLLE5B: PRINTF  #SECRM,TEMP,R3  ;PRINT ID MESSAGE
6044  042704  010346                                                        MOV     R3,-(SP)
6045  042706  013746  007426                                               MOV     TEMP,-(SP)
6046  042712  012746  020135                                               MOV     #SECRM,-(SP)
6047  042716  012746  000003                                               MOV     #3,-(SP)
6048  042722  010600                                                        MOV     SP,R0
6049  042724  104417                                                        TRAP    C$PNTF
6050  042726  062706  000010                                               ADD     #10,SP
6051  042732  000207                  DLLE5:  RTS     PC              ;RETURN TO CALLER
6052
6053
6054
```

```
                                .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
                                ;--

042734                          TALCK:
042734  042737  000002  007476          BIC     #DATCKB,PARAM   ;SET NOCHECK
042742  012702  002524                  MOV     #OPBUF,R2
042746  012722  177777          1$:     MOV     #-1,(R2)+       ;CLEAR OUT OPBUFFER FIRST
042752  022702  002646                  CMP     #OPEND,R2
042756  001373                          BNE     1$
042760                                  GMANID  OPRMM,OPBUF,A,0,1,72.,NO        ;GET TALK MESSAGE
042760  104443                                                                          TRAP    C$GMAN
042762  000406                                                                          BR      10002$
042764  002524                                                                          .WORD   OPBUF
042766  000142                                                                          .WORD   T$CODE
042770  014770                                                                          .WORD   OPRMM
042772  000000                                                                          .WORD   0
042774  000001                                                                          .WORD   T$LOLIM
042776  000110                                                                          .WORD   T$HILIM
043000                                                                          10002$:
043000  005002                          CLR     R2              ;NOW GET CHAR COUNT
043002  122762  000377  002524  2$:     CMPB    #377,OPBUF(R2)
043010  001402                          BEQ     3$
043012  005202                          INC     R2
043014  000772                          BR      2$
043016  010237  002166          3$:     MOV     R2,OPCNT

043022  012737  002524  007346          MOV     #OPBUF,DVTXA    ;SET UP TX ADDR.
043030  012737  002524  007432          MOV     #OPBUF,TEMP2
043036  013737  002166  007434          MOV     OPCNT,TEMP3
043044  013737  002166  007350          MOV     OPCNT,DVTCC     ;SET UP TX CC
043052  004737  026352                  JSR     PC,LOGTXQ
043056  052737  000210  007502          BIS     #QTX+#ETX,FLAG  ;SET UP FLAGS
043064  005037  007414                  CLR     CPTRR           ;CLEAR RX POINTER

043070  004737  044164                  JSR     PC,DVTXRX

043074  013737  007346  007432          MOV     DVTXA,TEMP2
043102  013737  007350  007434          MOV     DVTCC,TEMP3
043110  004737  026370                  JSR     PC,LOGTXC
043114  022737  054105  002524          CMP     #"EX,OPBUF      ;CHECK FOR EXIT
```

```
 6111  043122  001304                    BNE     TALCK
 6112  043124  022737  052111  002526    CMP     #''IT,OPBUF+2
 6113  043132  001300                    BNE     TALCK
 6114  043134  042737  000210  007502    BIC     #QTX+#ETX,FLAG  ;CLEAR THE TX BITS
 6115  043142  012737  000006  007470    MOV     #LIS,MODTYP     ;CHANGE TO LISTEN MODE
 6116  043150  000137  040570            JMP     GTRX2           ;AND GO BACK TO DISPATCH
```

```
 CZCLKC0 DMR,DMC-11 DATA COMM. LINK TEST        MACY11 30A(1052)  23-MAR-82  16:45  PAGE 149
CZCLKC.P11    19-MAR-82 18:32                   LISTEN MODE SECTION

                                        .SBTTL          LISTEN MODE SECTION

                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       LISTEN MODE SECTION
                                        ;       IN THIS 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.
                                        ;
                                        ; SUBORDINATE ROUTINES USED:
                                        ;
                                        ;               "DVRXQ" - QUE RECEIVE BUFFER SPACE TO DEVICE
                                        ;              "LOGRXQ" - LOG RECEIVE BUFFER QUED TO EVENT LOG
                                        ;              "DVTXRX" - WAIT FOR RX TO COMPLETE
                                        ;              "LOGRXC" - LOG RX COMPLETE TO EVENT LOG
                                        ;
                                        ; CALLING SEQUENCE:
                                        ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
                                        ;--

043154  042737  000002  007476  LISCK:  BIC     #DATCKB,PARAM   ;CLEAR CHECK BIT
043162                                  PRINTF  #LISP           ;PRINT PROMPT FOR OPR.
043162  012746  014757                                                  MOV     #LISP,-(SP)
043166  012746  000001                                                  MOV     #1,-(SP)
043172  010600                                                          MOV     SP,R0
043174  104417                                                          TRAP    C$PNTF
043176  062706  000004                                                  ADD     #4,SP
043202  012737  002524  007362  LISCKA: MOV     #OPBUF,DVRXA    ;SET DEVICE UP TO REC AT OPBUF
043210  012737  002524  007432          MOV     #OPBUF,TEMP2
043216  012737  000122  007364          MOV     #82.,DVRCC      ;SET UP CHAR COUNT TO 82.
043224  012737  000122  007434          MOV     #82.,TEMP3
043232  052737  000104  007502          BIS     #QRX+#ERX,FLAG  ;SET UP FLAG
043240  005037  007416                  CLR     CPTR            ;CLEAR THE TX.

043244  004737  044104                  JSR     PC,DVRXQ        ;QUE RX
043250  004737  026406                  JSR     PC,LOGRXQ

043254  004737  044164                  JSR     PC,DVTXRX       ;GO TO DEVICE RX. SUBROUTINE

043260  013737  007362  007432          MOV     DVRXA,TEMP2
043266  013737  007364  007434          MOV     DVRCC,TEMP3     ;SET UP ADDR.AND CC.
043274  004737  026424                  JSR     PC,LOGRXC       ;LOG COMPLETED
043300  063737  007362  007364          ADD     DVRXA,DVRCC
043306  105077  144052                  CLRB    @DVRCC
043312                                  PRINTF  #OPBFPT
043312  012746  002520                                                  MOV     #OPBFPT,-(SP)
043316  012746  000001                                                  MOV     #1,-(SP)
043322  010600                                                          MOV     SP,R0
043324  104417                                                          TRAP    C$PNTF
043326  062706  000004                                                  ADD     #4,SP
043332  022737  054105  002524          CMP     #'EX,OPBUF      ;COMPARE FOR EX OF "EXIT"
043340  001320                          BNE     LISCKA          ;IF NOT EXIT THEN GO BACK
043342  022737  052111  002526          CMP     #'IT,OPBUF+2    ;IF FIRST HALF OK CHECK NEXT PART
043350  001314                          BNE     LISCKA          ;IF NOT EXIT THE GO BACK
043352  012737  000005  007470          MOV     #TAL,MODTYP     ;CHANGE MODE TO TALK
043360  000137  040570                  JMP     GTRX2           ;RETURN TO DISPATCHER
```

```
 CZCLKCO DMR,DMC-11 DATA COMM. LINK TEST          MACY11 30A(1052)  23-MAR-82  16:45  PAGE 150
CZCLKC.P11    19-MAR-82 18:32                    LISTEN MODE SECTION

  6173
  6174
```

```
                                .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. (I.E. FULL/HALF DUPLEX BAUD RATE, MAINT MODE.)
                                ;
                                ; INPUTS:       "FHDPLX" INDICATES IF MODE IS FULL OR HALF DUPLEX. (1=FULL)
                                ;               ADDRESS POINTERS (SEL0,...) ALREADY POINT TO DEVICE'S REG.S
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;                "LGDVE" - LOG DEVICE ERROR TO EVENT LOG
                                ;               "TOORIO" - TIME OUT OR INPUT INTERRUPT OR OUTPUT INTERRUPT
                                ;                "CLRAW" - CLEAR RQI AND WAIT FOR RDI TO GO AWAY
                                ;
                                ; CALLING SEQUENCE:
                                ;                       JSR     PC,DVINIT
                                ;--

043364                          DVINIT:
                                        ;MASTER CLEAR DEVICE

043364  012737  000100  007542          MOV     #100,TIMER1     ;SET UP TIMER 1 FOR 100(OCTAL) TICKS
043372  005077  147002                  CLR     @SEL6
043376  005077  146772                  CLR     @SEL4
043402  012777  040000  146754          MOV     #MCLR,@SEL0     ;DO A MASTER CLEAR

043410  022737  000004  012412          CMP     #DMRC6,OPTYP    ;IS THIS A 8206
043416  001003                          BNE     DVIN6           ;IF NOT GO TO 6
043420  112777  000200  146740          MOVB    #200,@BSEL1     ;SET RUN FOR 8206
043426  022737  000006  012412  DVIN6:  CMP     #DMR6,OPTYP     ;IS THIS AN 8206 DMR
043434  001003                          BNE     DVIN2           ;IF NOT GO TO 2
043436  112777  000200  146722          MOVB    #200,@BSEL1     ;SET RUN BIT FOR 8206

043444  005777  146714          DVIN2:  TST     @SEL0           ;IS RUN BIT SET
043450  100426                          BMI     DVIN1           ;IF YES GO TO 1 ELSE...
043452                                  BREAK                                           TRAP    C$BRK
043452  104422
043454  005737  007542                  TST     TIMER1          ;SEE IF TIME HAS EXPIRED
043460  001371                          BNE     DVIN2           ;IF NOT GO BACK AND CHECK
                                                                ;AGAIN ELSE...PRINT ERROR
043462  012737  024424  007432          MOV     #DVEM3,TEMP2
043470  017737  146670  007434          MOV     @SEL0,TEMP3
043476  017737  146666  007436          MOV     @SEL2,TEMP4     ;LOAD UP ERRM. AND REG OUTPUTS
043504  004737  026434                  JSR     PC,LGDVE        ;LOG TIME OUT WAITING FOR RUN
043510  005237  007400                  INC     ERRCNT
043514                                  ERRSOFT 11,DVEM3,ERR13
```

```
                                                                                TRAP    C$ERSOFT
                                                                                .WORD   11
                                                                                .WORD   DVEM3
                                                                                .WORD   ERR13

043524  000717                          BR DVINIT               ;GO BACK AND TRY MSTR CLR AGAIN IF ERROR

043526                          DVIN1:

                                        ; DO BASE IN COMMAND

043526  042737  000003  007502          BIC     #3,FLAG         ;CLEAR INPUT AND OUTPUT INT FLAGS
043534  112777  000143  146622          MOVB    #143,@BSEL0     ;SET UP BASE IN INT EN
043542  004737  045012                  JSR     PC,TOORIO       ;GO WAIT FOR INTERRUPT OR TIME OUT
043546  012777  017370  146620          MOV     #BASE,@SEL4

043554  012777  000000  146616          MOV     #0,@SEL6        ;SET UP SEL 6
043562  023727  012412  000006          CMP     OPTYP,#6        ;IS THIS DMR MODE
043570  002403                          BLT     DVIN7           ;IF NOT GO TO 7
043572  012777  000522  146600          MOV     #522,@SEL6      ;SET DMR MODE
043600  052777  000100  146562  DVIN7:  BIS     #IEO,@SEL2      ;SET IEO
043606  042777  004000  146550          BIC     #LULOOP,@SEL0   ;CLEAR LU LOOP
043614  022737  000001  007472          CMP     #TTL,MLTYP      ;IS TTL SELECTED
043622  001003                          BNE     DVIN3           ; IF NOT GO TO 3
043624  052777  004000  146532          BIS     #LULOOP,@SEL0   ;ELSE SET LU LOOP
043632  004737  044704          DVIN3:  JSR     PC,CLRAW

                                        ; DO WRITE MODEM IF DMR MODE

043636  023727  012412  000006          CMP     OPTYP,#6        ;IS THIS DMR MODE
043644  002437                          BLT     DVIN8           ;IF NOT GO TO 8
043646  112777  000145  146510          MOVB    #145,@BSEL0     ;SET UP WRITE MODEM
043654  004737  045012                  JSR     PC,TOORIO       ;GO TO  WAIT FOR INT
043660  042777  000014  146512          BIC     #BIT2+#BIT3,@SEL6               ;CLEAR BSEL6 AND 7
043666  022737  000004  007472          CMP     #MODREM,MLTYP   ;IS THIS REMOTE LOOP
043674  001003                          BNE     DVIN9           ;IF NOT GO TO 9
043676  052777  000004  146474          BIS     #BIT2,@BSEL6    ;SET THE BIT
043704  022737  000003  007472  DVIN9:  CMP     #MODLOC,MLTYP   ;IS IT MODEM LOCAL
043712  001003                          BNE     DVIN10          ;IF NOT EXIT
043714  052777  000010  146456          BIS     #BIT3,@BSEL6    ;SET MODEM LOCAL
043722  004737  044704          DVIN10: JSR     PC,CLRAW        ;CLEAR RDI AND WAIT

                                        ; ENABLE EXTENDED ERROR IF DMR MODE


043726  112777  000146  146430          MOVB    #146,@BSEL0     ;SET UP FOR ENABLE
043734  004737  045012                  JSR     PC,TOORIO
043740  004737  044704                  JSR     PC,CLRAW        ;CLEAR RDI AND WAIT

                                        ; DO CONTROL IN COMMAND

043744  112777  000141  146412  DVIN8:  MOVB    #141,@BSEL0     ;SET UP CONTROL IN
043752  004737  045012                  JSR     PC,TOORIO       ;WAIT FOR INT OR TIME OUT
043756  005077  146416                  CLR     @SEL6           ;CLEAR HALF/DUP
043762  022737  000004  007470          CMP     #DOW,MODTYP     ;IS THIS DOWN LINE LOAD?
043770  001004                          BNE     DVIN5           ; BR IF NOT
```

```
 6287  043772  052777  002400  146400          BIS     #MAINTB+HALFDB,@SEL6    ;IF SO SET MAINT MODE  BIT
 6288  044000  000406                          BR      DVIN4                   ; AND FORCE HALF DUPLEX
 6289
 6290  044002  005737  007474          DVIN5:  TST     FHDPLX                  ;IS THIS A HALF/DUP
 6291  044006  001003                          BNE     DVIN4                   ;IF NOT GO TO 4
 6292  044010  052777  002000  146362          BIS     #HALFDB,@SEL6           ;ELSE SET HALF/DUP
 6293
 6294  044016  017737  146356  007444  DVIN4:  MOV     @SEL6,CONTIN            ;SET UP CONTROL IN FOR MODS
 6295  044024  004737  044704                  JSR     PC,CLRAW                ;GO CLEAR RQI AND WAIT
 6296                                                                          ;FOR RDI TO GO AWAY.
 6297  044030  023727  012412  000006          CMP     OPTYP,#6                ;IS THIS DMR ?
 6298  044036  002403                          BLT     DVINEX                  ;NO,EXIT
 6299  044040  052737  001000  007502          BIS     #DMRRUN,FLAG            ;SET DMRRUN OUTPUT EXPECTED BIT
 6300  044046  000207                  DVINEX: RTS     PC                      ;RETURN TO CALLER
 6301
 6302
 6303
 6304
 6305
```

```
                                .SBTTL                  DEVICE GET MODEM STATUS SUBROUTINE
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       "DVMODS"   GET MODEM STATUS
                                ;
                                ; IMPLICIT INPUTS:
                                ;       THE BIT POSITION AND AVAIABLITY OF THE MODEM SIGNALS CTS,DSR,...RI,..
                                ;       IN THE DEPENDENT PORITION OF THE GLOBAL EQUATES SECTION.
                                ;
                                ; OUTPUTS:
                                ;       CURRENT MODEM SIGNAL VALUES IN "MODS"
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;               "TOORIO" - TIME OUT OR INPUT INTERRUPT OR OUTPUT INTERRUPT
                                ;               "CLRAW" - CLEAR RQI AND WAIT FOR RDI TO CLEAR
                                ;
                                ; CALLING SEQUENCE:
                                ;       JSR     PC,DVMODS
                                ;--
044050  112777  000141  146306  DVMODS: MOVB    #141,@BSEL0     ;SET UP CONTORL IN
044056  004737  045012          JSR     PC,TOORIO       ;GO TIME OUT CHECK
044062  017737  146306  010456  MOV     @SEL4,MODS      ;SET UP MODEM STATUS
044070  013777  007444  146302  MOV     CONTIN,@SEL6    ;SET UP OLD CONTORL IN
044076  004737  044704          JSR     PC,CLRAW
044102  000207                  RTS     PC              ;RETURN TO CALLER
```

```
                                .SBTTL                  DEVICE QUEUE RECEIVE SPACE SUBROUTINE

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       DVRXQ - THIS SUB ROUTINE QUES THE REC 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
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;               "TOORIO" - TIME OUT OR OUTPUT INTERRUPT OR INTPUT INTERRUPT
                                ;               "CLRAW"  - CLEAR RQI AND WAIT FOR RDI TO CLEAR
                                ;
                                ; CALLING SEQUENCE:
                                ;       JSR     PC,DVRXQ
                                ;--

044104                          DVRXQ:
044104  032737  000004  007502          BIT     #QRX,FLAG
044112  001423                         BEQ     DVREX           ;IF NOT RX THEN EXIT
                                                                ;ELSE QUE RX
044114  042737  000004  007502          BIC     #QRX,FLAG       ;CLEAR FLAG FOR RX
044122  112777  000144  146234          MOVB    #144,@BSEL0
044130  004737  045012                  JSR     PC,TOORIO       ;GO CHECK FOR IN OR OUT
044134  017737  146234  010456          MOV     @SEL4,MODS      ;SET UP NEW MOD STATUS
044142  013777  007362  146224          MOV     DVRXA,@SEL4
044150  013777  007364  146222          MOV     DVRCC,@SEL6     ;LOAD CC AND ADDR
044156  004737  044704                  JSR     PC,CLRAW        ;CLEAR AND WAIT
044162  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 BACC OUTPUT INTERRUPT IS RECIEVED BEFORE
                                ;       60 SECONDS. AFTER REPORTING ERROR TIMER IS RE STARTED
                                ;       AND DEVICE WILL CONTINUE TO WAIT FOR INTERRUPT. CODE
                                ;       ALSO REPORTS ERROR IF INPUT INTERRUPT OCCURS WHEN
                                ;       EXPECTING OUTPUT INTERRUPT;WHEN RX BACC OCCURS WHEN
                                ;       EXPECTING TX,AND WHEN TX INT. OCCURS WHEN EXPECTING
                                ;       RECIEVE.
                                ;
                                ;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:
                                ;
                                ;               ''TOORIO'' - TIME OUT OR OUTPUT INTERRUPT OR INTPUT INTERRUPT
                                ;               ''CLRAW''  - CLEAR RQI AND WAIT FOR RDI TO CLEAR
                                ;               ''LGDVE''  - LOG DEVICE ERROR TO EVENT LOG
                                ;               ''OUTHDL'' - OUTPUT INTERRUPT HANDLER CODE
                                ;
                                ; CALLING SEQUENCE:
                                ;       JSR     PC,DVTXRX
                                ;--
                                
 044164  032737  000010  007502  DVTXRX: BIT     #QTX,FLAG       ;ANY TX TO QUE
 044172  001423                  BEQ     DVTR3           ;IF NOT GO WAIT FOR OUPUT
 044174  042737  000010  007502          BIC     #QTX,FLAG       ;CLEAR FLAG
 044202  112777  000140  146154          MOVB    #140,@BSEL0
 044210  004737  045012                  JSR     PC,TOORIO       ;GO CHECK FOR IN OR OUT
 044214  017737  146154  010456          MOV     @SEL4,MODS      ;PUT IN NEW MOD STAT
 044222  013777  007346  146144          MOV     DVTXA,@SEL4
 044230  013777  007350  146142          MOV     DVTCC,@SEL6
 044236  004737  044704                  JSR     PC,CLRAW        ;CLEAR RQUI ANDWAIT
 044242                          DVTR3:
 044242  012737  000074  007546          MOV     #60.,TIMERS     ;SET TIMER FOR 60 SECS
 044250  032737  000060  007502  TOINOT: BIT     #CRX+#CTX,FLAG  ;IS IT TX OR RX COMP ALREADY?
```

```
6434  044256  001071                          BNE     DVTR4           ;IS SO EXIT
6435
6436  044260  005737  007546                  TST     TIMERS          ;IS TIMER EXPIRED
6437  044264  001022                          BNE     TOIN1
6438  044266  012737  024510  007432          MOV     #DVEM4,TEMP2
6439  044274  017737  146064  007434          MOV     @SEL0,TEMP3
6440  044302  017737  146062  007436          MOV     @SEL2,TEMP4
6441  044310  004737  026434                  JSR     PC,LGDVE
6442  044314  005237  007400                  INC     ERRCNT
6443  044320                                  ERRSOFT 12,DVEM4,ERR13
6444  044320  104457                                                          TRAP    C$ERSOFT
6445  044322  000014                                                          .WORD   12
6446  044324  024510                                                          .WORD   DVEM4
6447  044326  026110                                                          .WORD   ERR13
6448  044330  000744                          BR      DVTR3           ;RETURN TO CHECK TIMER
6449
6450
6451  044332                          TOIN1:  BREAK
6452  044332  104422                                                          TRAP    C$BRK
6453  044334  032737  000001  007502          BIT     #ININT,FLAG     ;IS IT INPUT INTERRUPT
6454  044342  001425                          BEQ     TOIN2           ;IF SO LOG ERROR
6455
6456  044344  012737  024602  007432          MOV     #DVEM5,TEMP2
6457  044352  017737  146006  007434          MOV     @SEL0,TEMP3
6458  044360  017737  146004  007436          MOV     @SEL2,TEMP4
6459  044366  004737  026434                  JSR     PC,LGDVE
6460  044372  042737  000001  007502          BIC     #ININT,FLAG     ;CLEAR BIT
6461  044400  005237  007400                  INC     ERRCNT
6462  044404                                  ERRSOFT 13,DVEM5,ERR13
6463  044404  104457                                                          TRAP    C$ERSOFT
6464  044406  000015                                                          .WORD   13
6465  044410  024602                                                          .WORD   DVEM5
6466  044412  026110                                                          .WORD   ERR13
6467  044414  000715                          BR      TOINOT
6468
6469  044416  032737  000002  007502  TOIN2:  BIT     #OTINT,FLAG
6470  044424  001711                          BEQ     TOINOT          ;IF NOT OUTPUT GO BACK AND
6471                                                                  ;CHECK TIMER AGAIN
6472  044426  004737  045134                  JSR     PC,OUTHDL       ;ELSE HANDLE OUTPUT AND RETURN
6473  044432  032737  000060  007502          BIT     #CTX+#CRX,FLAG  ;IS IT TX OR RX
6474  044440  001703                          BEQ     TOINOT          ;IF NOT GO BACK AND TRY AGAIN
6475  044442  032737  000020  007502  DVTR4:  BIT     #CTX,FLAG       ;IS IT TX
6476  044450  001440                          BEQ     DVTR5           ;IF NOT TRY RX
6477  044452  032737  000200  007502          BIT     #ETX,FLAG       ;IF SO SHOULD IT BE
6478  044460  001020                          BNE     DVTR4A          ;IF IT SHOULD GO TO 4A
6479  044462  012737  025125  007432          MOV     #DVEM9,TEMP2    ;ELSE LOG ERROR
6480  044470  013737  045762  007434          MOV     TSEL4,TEMP3
6481  044476  013737  045764  007436          MOV     TSEL6,TEMP4
6482  044504  004737  026434                  JSR     PC,LGDVE        ;
6483  044510                                  ERRSOFT 14,DVEM9,ERR13  ;REPORT ERROR
6484  044510  104457                                                          TRAP    C$ERSOFT
6485  044512  000016                                                          .WORD   14
6486  044514  025125                                                          .WORD   DVEM9
6487  044516  026110                                                          .WORD   ERR13
6488
6489  044520  000411                          BR      DVTR4B          ;THEN CLEAR COMPL.FLAG
```

```
 6490  044522  013737  045762  007346  DVTR4A: MOV     TSEL4,DVTXA
 6491  044530  013737  045764  007350          MOV     TSEL6,DVTCC
 6492  044536  052737  000010  007502          BIS     #QTX,FLAG       ;AND SET TX COMPL FLAG
 6493  044544  042737  000020  007502  DVTR4B: BIC     #CTX,FLAG       ;ELSE CLEAR FLAG
 6494  044552  032737  000040  007502  DVTR5:  BIT     #CRX,FLAG       ;IS IT RX TOO?
 6495  044560  001440                          BEQ     DVTREX          ;IF NOT THEN EXIT.
 6496  044562  032737  000100  007502          BIT     #ERX,FLAG       ;TEST IS THIS SUPPOSED TO BE RX
 6497  044570  001020                          BNE     DVTR5A          ;IF YES PROCESS AS SUCH
 6498  044572  012737  025036  007432          MOV     #DVEM8,TEMP2
 6499  044600  013737  045766  007434          MOV     RSEL4,TEMP3
 6500  044606  013737  045770  007436          MOV     RSEL6,TEMP4     ;ELSE
 6501  044614  004737  026434                  JSR     PC,LGDVE        ;LOG ERROR
 6502  044620                                  ERRSOFT 15,DVEM8,ERR13
 6503  044620  104457                                                  TRAP    C$ERSOFT
 6504  044622  000017                                                  .WORD   15
 6505  044624  025036                                                  .WORD   DVEM8
 6506  044626  026110                                                  .WORD   ERR13
 6507
 6508  044630  000411                          BR      DVTRX1          ;AND EXIT
 6509  044632  013737  045766  007362  DVTR5A: MOV     RSEL4,DVRXA
 6510  044640  013737  045770  007364          MOV     RSEL6,DVRCC
 6511  044646  052737  000004  007502          BIS     #QRX,FLAG
 6512  044654  042737  000040  007502  DVTRX1: BIC     #CRX,FLAG       ;CLEAR FLAG FOR RX DONE
 6513  044662  000207                  DVTREX: RTS     PC              ;AND EXIT
 6514
```

```
                                ; DEVICE DEPENDENT SUBROUTINES
                                
                                
                                .SBTTL                  DEVICE INTERRUPT SERVICE ROUTINES
                                
                                
044664                                  BGNSRV  DVINS
044664                                                                          DVINS::
044664  052737  000001  007502          BIS     #ININT,FLAG
                                        ENDSRV
044672                                                                          L10021:
044672                                                                                  RTI
044672  000002
                                
044674                                  BGNSRV  DVOUTS
044674                                                                          DVOUTS::
044674  052737  000002  007502          BIS     #OTINT,FLAG
                                        ENDSRV
044702                                                                          L10022:
044702                                                                                  RTI
044702  000002
                                
                                
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       CLRAW - CLEAR RQI AND WAIT FOR RDI TO GO AWAY
                                ;       THIS CODE CLEARS THE INPUT REQEST BIT(RQI) SETS A
                                ;       TIMER UP TO TIME 50(OCTAL) TICKS AND MAKES SURE
                                ;       RDI CLEARS BEFORE TIMER EXPIRES. IF TIMER EXPIRERS
                                ;       CODE REPORTS ERROR AND SETS UP TIMER AND WAITS AGAIN.
                                ;
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;               "LGDVE" - LOG DEVICE ERROR (TIME OUT)
                                ;
                                ;
                                ; CALLING SEQUENCE:
                                ;       JSR     PC,CLRAW
                                ;--
                                
044704  011637  007462          CLRAW:  MOV     (SP),PCADD      ;SAVE PC OF CALLING ROUTINE
044710  042777  000040  145446          BIC     #RQI,@SEL0
044716  012737  000050  007542  CLRA3:  MOV     #50,TIMER1      ;SET UP TIMER FOR 50(OCTAL) TICKS
044724  005737  007542          CLRA1:  TST     TIMER1
044730  001406                          BEQ     CLRA2           ;IF TIMER EXPIRED ERROR
044732                                  BREAK
044732  104422                                                                  TRAP    C$BRK
044734  032777  000200  145422          BIT     #RDI,@SEL0      ;IS RDI CLEAR
044742  001370                          BNE     CLRA1           ;IF NOT GO CHECK TIMER
                                                                ; ELSE
044744  000207                          RTS     PC              ;RETURN TO CALLER
044746  012737  024252  007432  CLRA2:  MOV     #DVEM0,TEMP2
044754  017737  145404  007434          MOV     @SEL0,TEMP3
044762  017737  145402  007436          MOV     @SEL2,TEMP4
044770  004737  026434                  JSR     PC,LGDVE        ;LOG DEVEICE EVENT 0
044774  005237  007400                  INC     ERRCNT
```

```
6571  045000
6572  045000  104457  ERRSOF  16,DVEM0,ERR9   ;WHILE WAITING FOR RDI
6573  045002  000020                                  TRAP    C$ERSOFT
6574  045004  024252                                  .WORD   16
6575  045006  026032                                  .WORD   DVEM0
6576  045010  000742                                  .WORD   ERR9
                      BR      CLRA3           ;RESET TIMER AND CONTINUE
```

```
                                .SBTTL                  TIME OUT OR INPUT INT. OR OUTPUT INT.


                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       TOORIO - TIME OUT OR INPUT INTERRUPT OR OUTPUT INTERRUPT
                                ;       THIS ROUTINE SETS UP A TIMER FOR 100 (OCTAL) TICKS
                                ;       THEN CHECKS FOR TIME OUT,OR INPUT INTERRUPT,OR OUTPUT
                                ;       INTERRUPT. IF TIME OUT OCCURS IT REPORTS ERROR AND
                                ;       RESTARTS TIMER. IF INPUT INTERRUPT OCCURS RETURN TO CALLER
                                ;       IF OUTPUT INTERRPUT OCCURS LOG IT AND CONTINUE WAITING FOR
                                ;       INPUT INTERRUPT.
                                ;
                                ; USE OF FLAGS:
                                ;               ''OTINT'' - SET BY OUTPUT INT ROUTINE
                                ;               ''ININT'' - SET BY INPUT INT. ROUTINE
                                ;                       CLEARED BY THIS ROUTINTE.
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;               ''OUTHDL'' - OUTPUT INTERRUPT HANDLER
                                ;
                                ; CALLING SEQUENCE:
                                ;       JSR     PC,TOORIO
                                ;--
                                TOORIO: MOV     (SP),PCADD      ;SAVE ADDR. OF CALLING ROUTINE
                                        MOV     #100,TIMER1     ;SET UP TIMER
                                TOOR3:  TST     TIMER1          ;IS TIME EXPIRED
                                        BNE     TOOR1           ;IF NOT CONTINUE
                                                                ;IF YES ERROR
                                        MOV     #DVEM1,TEMP2
                                        MOV     @SEL2,TEMP4
                                        MOV     @SEL0,TEMP3
                                        JSR     PC,LGDVE
                                        INC     ERRCNT
                                        ERRSOFT 17,DVEM1,ERR9
                                                                                TRAP    C$ERSOFT
                                                                                .WORD   17
                                                                                .WORD   DVEM1
                                                                                .WORD   ERR9
                                        BR      TOORIO

                                TOOR1:  BREAK
                                                                                TRAP    C$BRK
                                        BIT     #OTINT,FLAG     ;IS THERE AN OUTPUT
                                                                ;PENDING
                                        BEQ     TOOR2           ;IF NOT GO TO 2
                                                                ;ELSE GO HANDL IT
                                        JSR     PC,OUTHDL       ;
                                TOOR2:  BIT     #ININT,FLAG     ;IS THERE AN INPUT PENDING
                                        BEQ     TOOR3           ;IF NOT GO BACK TO TIMER CK.
                                        BIC     #ININT,FLAG     ;ELSE CLEAR THE INPUT PEND FLAG
                                        RTS     PC              ;AND RETURN TO CALLER
```

```
                                .SBTTL                  OUTPUT INTERRUPT HANDLER

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       OUTHDL - OUTPUT INTERRUPT HANDLER
                                ;       THIS ROUTINE IS CALLED WHEN AN OUTPUT INTERRUPT HAS SET
                                ;       THE ''OTINT'' BIT IN THE ''FLAG'' WORD. IT CHECKS FOR
                                ;       AN RDO SIGNAL IF NO RDO THEN REPORT ILLEGAL INTERRUPT.
                                ;       THEN IT CHECKS FOR BACC OUT IF NOT BACC OUT REPORT THE
                                ;       TYPE OF OUTPUT ERROR. IF BACC OUT FIND IF RX OR TX
                                ;       IF RX SET CRX BIT AND MOVE ADDR AND BYTE COUNT TO RSEL4
                                ;       AND RSEL6. IF TX SET CTX BIT AND MOVE ADDR AND BYTE COUNT
                                ;       TO TSEL4 AND TSEL6. CLEAR OTINT FLAG AND RETURN TO CALLER.
                                ;
                                ; USE OF FLAGS:
                                ;               ''OTINT'' - SET BY OUPUT ROUTINE
                                ;                           CLEARED BY THIS ROUTINE
                                ;               ''DMRRUN'' - SET BY DVINIT ROUTINE IF THIS IS DMR
                                ;                            CHECKED AND CLEARED BY THIS ROUTINE.
                                ;               ''CTX''    - SET IF TRANSMIT COMLETED
                                ;               ''CRX''    - SET IF RECIEVE COMPLETED
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;               ''LGDVE'' -LOG DEVICE ERRORS TO EVENT LOG
                                ;
                                ; CALLING SEQUENCE
                                ;               JSR     PC,OUTHDL
                                ;
                                ;--
045134  011637  007462          OUTHDL: MOV     (SP),PCADD      ;SAVE ADDR. OF CALLING ROUTINE
045140  042737  000002  007502          BIC     #OTINT,FLAG
045146  032777  000200  145214          BIT     #RDO,@SEL2      ;CLEAR PEND FLAG AND CHK FOR RDO
045154  001023                          BNE     OUTH1           ;IF RDO OK ...ELSE LOG ERROR
045156  012737  024674  007432          MOV     #DVEM6,TEMP2
045164  017737  145200  007434          MOV     @SEL2,TEMP3
045172  017737  145202  007436          MOV     @SEL6,TEMP4
045200  004737  026434                  JSR     PC,LGDVE        ;GO LOG ERROR
045204  005237  007400                  INC     ERRCNT
045210                                  ERRSOFT 18,DVEM6,ERR9
045210  104457                                                          TRAP    C$ERSOFT
045212  000022                                                          .WORD   18
045214  024674                                                          .WORD   DVEM6
045216  026032                                                          .WORD   ERR9

                                ;EXIT TEST IF ERROR
                                ;
045220                          ESCAPE  TST
045220  104410                                                          TRAP    C$ESCAPE
045222  000712                                                          .WORD   L10020-.

045224  032777  000001  145136  OUTH1:  BIT     #BACC,@SEL2     ;IS THE OUTPUT BACC
045232  001002                          BNE     1$              ; BR IF NO
045234  000137  045660                  JMP     OUTH2           ;IF SO GO TO 2
```

```
                                                        ;ELSE LOG ERROR AND PRINT IT
045240  017737  145134  007436  1$:     MOV     @SEL6,TEMP4
                                        ; IF NO BUFFER OUTPUT JUST COUNT THEM

045246  032737  000004  007436          BIT     #BIT2,TEMP4
045254  001404                          BEQ     OUTH6           ;IF NO BUFF INC COUNT AND EXIT
                                                                ;ELSE GO TO 6
045256  005237  007374                  INC     NOBUF
045262  000137  045752                  JMP     OUTHEX

045266  023727  012412  000006  OUTH6:  CMP     OPTYP,#6        ;DMR ?
045274  002426                          BLT     51$             ;IF NOT DMR MODE SKIP TO 51
045276  032737  002000  007502          BIT     #BTUP,FLAG      ;HERE BECAUSE OF BASE TABLE UPDATE REQ ? REV B BY EC
045304  001402                          BEQ     50$             ;NO,BRANCH  REV B BY EC
045306  000137  045752                  JMP     OUTHEX          ;EXIT
045312  032737  000040  007436  50$:    BIT     #BIT5,TEMP4     ;IS IT RUN STATE
045320  001414                          BEQ     51$             ;IF NOT BRANCH
045322  032737  001000  007502          BIT     #DMRRUN,FLAG    ;IS RUN EXPECTED
045330  001405                          BEQ     52$             ;IF NOT BRANCH
045332  042737  001000  007502          BIC     #DMRRUN,FLAG    ;IF SO THEN CLEAR EXPECTED
045340  000137  045752                  JMP     OUTHEX          ;AND EXIT
045344  012737  025437  007442  52$:    MOV     #RUNSBM,CONOTM
045352  012737  024751  007432  51$:    MOV     #DVEM7,TEMP2
045360  017737  145004  007434          MOV     @SEL2,TEMP3

045366  004737  026434                  JSR     PC,LGDVE
045372  012737  014641  007442          MOV     #LP0,CONOTM     ;LOAD "NULL STRING" TO INIT CONOTM
045400  032737  000001  007436          BIT     #BIT0,TEMP4     ;IS THIS DATA CHECK
045406  001403                          BEQ     1$
045410  012737  025424  007442          MOV     #DATCKM,CONOTM
045416  032737  000002  007436  1$:     BIT     #BIT1,TEMP4     ;IS THIS TIMEOUT
045424  001403                          BEQ     2$
045426  012737  025413  007442          MOV     #TIMOM,CONOTM
045434  032737  000010  007436  2$:     BIT     #BIT3,TEMP4     ;IS THIS DDCMP MAINT RECVD
045442  001403                          BEQ     4$
045444  012737  025373  007442          MOV     #DDCMRM,CONOTM
045452  032737  000020  007436  4$:     BIT     #BIT4,TEMP4     ;IS THIS LOST DATA
045460  001403                          BEQ     5$
045462  012737  025361  007442          MOV     #LOSDAM,CONOTM
045470  032737  000100  007436  5$:     BIT     #BIT6,TEMP4     ;IS THIS DISCONNECT
045476  001403                          BEQ     6$
045500  012737  025346  007442          MOV     #DISCOM,CONOTM
045506  032737  000200  007436  6$:     BIT     #BIT7,TEMP4     ;IS THIS DDCMP START RECVD
045514  001403                          BEQ     7$
045516  012737  025326  007442          MOV     #DDCSRM,CONOTM
045524  032737  000400  007436  7$:     BIT     #BIT8,TEMP4     ;IS THIS NON-EXSISTENT MEMORY
045532  001403                          BEQ     8$
045534  012737  025310  007442          MOV     #NXMM,CONOTM
045542  032737  001000  007436  8$:     BIT     #BIT9,TEMP4     ;IS THIS PROCEDURE ERROR
045550  001403                          BEQ     9$
045552  012737  025270  007442          MOV     #PROEM,CONOTM
045560  023727  012412  000006  9$:     CMP     OPTYP,#6        ;IS THIS DMR MODE
045566  002416                          BLT     11$             ;IF NOT BRANCH
045570  032737  002000  007436          BIT     #BIT10,TEMP4    ;IS THIS A RX IDLE
045576  001403                          BEQ     10$             ;IF NOT BRANCH
045600  012737  025461  007442          MOV     #RXIDM,CONOTM   ;IF SO SET UP MESSAGE
```

```
6744  045606  032737  004000  007436  10$:    BIT     #BIT11,TEMP4    ;IS THIS CTS FAILED
6745  045614  001403                          BEQ     11$             ;IF NOT BRANCH
6746  045616  012737  025505  007442          MOV     #CTSFM,CONOTM   ;IF SO SET UP MESSAGE
6747  045624  032737  010000  007436  11$:    BIT     #BIT12,TEMP4    ;IS THIS CD GLITCHED
6748  045632  001403                          BEQ     12$             ;BR IF NO
6749  045634  012737  025471  007442          MOV     #CDGLM,CONOTM   ;IF SO SET UP MESSAGE
6750
6751  045642  005237  007400          12$:    INC     ERRCNT
6752  045646                                  ERRSOFT 19,DVEM7,ERR8
6753  045646  104457                                                          TRAP    C$ERSOFT
6754  045650  000023                                                          .WORD   19
6755  045652  024751                                                          .WORD   DVEM7
6756  045654  025750                                                          .WORD   ERR8
6757  045656  000435                          BR      OUTHEX          ;CLEAR RDO AND RETURN TO CALLER
6758
6759  045660                          OUTH2:
6760  045660  032777  000004  144502          BIT     #RXBIT,@SEL2    ;IS THIS RX BACC OUT
6761  045666  001012                          BNE     OUTH3           ;IF NOT THEN IT MUST BE TX.
6762  045670  052737  000020  007502          BIS     #CTX,FLAG
6763  045676  017737  144472  045762          MOV     @SEL4,TSEL4
6764  045704  017737  144470  045764          MOV     @SEL6,TSEL6
6765  045712  000417                          BR      OUTHEX
6766
6767  045714  052737  000040  007502  OUTH3:  BIS     #CRX,FLAG       ;SET RX COMPL
6768  045722  017737  144446  045766  OUTH4:  MOV     @SEL4,RSEL4     ;THEN MOVE TO TEMP
6769  045730  017737  144444  045770          MOV     @SEL6,RSEL6     ;AND SEL6 TO TEMP
6770  045736  005737  012412                  TST     OPTYP           ;DMC ?
6771  045742  001003                          BNE     OUTHEX          ;NO,BRANCH
6772  045744  042737  140000  045770          BIC     #BIT15!BIT14,RSEL6 ;CLEAR Q SYNC & SELECT BITS
6773  045752  042777  000200  144410  OUTHEX: BIC     #RDO,@SEL2      ;CLEAR RDO
6774  045760  000207                          RTS     PC              ;RETURN TO CALLER
6775  045762  000000                  TSEL4:  .WORD   0
6776  045764  000000                  TSEL6:  .WORD   0
6777  045766  000000                  RSEL4:  .WORD   0
6778  045770  000000                  RSEL6:  .WORD   0
6779
6780  045772  000207                          RTS     PC
6781
6782
```

```
6783
6784                                    .EVEN
6785
6786
6787
6788                                .SBTTL  REQUEST BASE TABLE UPDATE
6789                                ;       THIS ROUTINE ADDED FOR REV B BY EC
6790                                ;++
6791                                ;FUNCTIONAL DESCRIPTION:
6792                                ;       DVBTUP - THIS ROUTINE IS CALLED AT END OF PASS TO UPDATE THE DMR
6793                                ;                BASE TABLE IN LOCAL MEMORY BY ISSUING AN UPDATE BASE TABLE
6794                                ;                REQUEST TO THE DMR.
6795                                ;
6796                                ;USE OF FLAGS:
6797                                ;       FLAG - BITS IN THIS WORD ARE SET BY THE DEVICE INTERRUPT ROUTINE.
6798                                ;              WHEN SET TO A 1, THE FOLLOWING BITS MEAN
6799                                ;                ININT = BIT 1 = DEVICE INPUT READY
6800                                ;                OTINT = BIT 2 = DEVICE OUT READY
6801                                ;
6802                                ;INPUTS: NONE REQUIRED.
6803                                ;
6804                                ;SUBORDINATE ROUTINES USED:
6805                                ;
6806                                ;       CLRAW - CLEAR RQI AND WAIT FOR DEVICE TO CLEAR RDI.
6807                                ;       OUTHDL - THE DEVICE OUTPUT SERVICE ROUTINE.
6808                                ;       LGDVE - LOG A DEVICE ERROR TO EVENT LOG.
6809                                ;
6810                                ;CALLING SEQUENCE:
6811                                ;       JSR     PC,DVBTUP
6812                                ;--
6813                                ;DEVICE BASE TABLE UPDATE ROUTINE
6814  045774                        DVBTUP:
6815  045774  023727  012412  000006        CMP     OPTYP,#6        ;DMR IN DMR MODE ?
6816  046002  002453                       BLT     40$             ;NO,BRANCH
6817  046004  112777  000151  144352        MOVB    #151,@BSEL0     ;REQUEST BASE TABLE UPDATE
6818  046012  052777  000100  144350        BIS     #IEO,@SEL2      ;ENABLE INTERRUPT ON OUTPUT READY
6819  046020  004737  045012                JSR     PC,TOORIO       ;WAIT FOR RDI
6820  046024  004737  044704                JSR     PC,CLRAW        ;CLEAR RQI AND WAIT FOR RDI TO CLEAR
6821  046030  012737  000200  007542        MOV     #200,TIMER1     ;WAIT FOR INTERRUPT
6822  046036                        10$:    BREAK                                                   TRAP    C$BRK
6823  046036  104422
6824  046040  032737  000002  007502        BIT     #OTINT,FLAG     ;OUTPUT READY ?
6825  046046  001021                       BNE     30$             ;YES,BRANCH
6826  046050  005737  007542                TST     TIMER1          ;TIME OUT ?
6827  046054  001370                       BNE     10$             ;NO,BRANCH
6828  046056  012737  024424  007432        MOV     #DVEM3,TEMP2    ;DEVICE ERROR
6829  046064  017737  144274  007434        MOV     @SEL0,TEMP3     ;SAVE REGISTER
6830  046072  017737  144272  007436        MOV     @SEL2,TEMP4     ;SAVE REGISTER
6831  046100  004737  026434                JSR     PC,LGDVE        ;GO LOG DEVICE ERROR
6832  046104  005237  007400                INC     ERRCNT          ;BUMP ERROR COUNT
6833  046110  000410                       BR      40$             ;RETURN
6834  046112  052737  002000  007502  30$:   BIS     #BTUP,FLAG      ;SET BASE TABLE UPDATE BIT
6835  046120  004737  045134                JSR     PC,OUTHDL       ;OUTPUT SERVICE ROUTINE
6836  046124  042737  002000  007502        BIC     #BTUP,FLAG      ;CLEAR BASE TABLE UPDATE BIT
6837  046132  000207                40$:    RTS     PC              ;RETURN
6838
```

```
                                ENDTST
                                                                L10020:
                                                                        TRAP    C$ETST
```

```
                                .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.
                                ;--

046136                                  BGNHRD
046136  000025                                                    .WORD L10023-L$HARD/2
046140                                                          L$HARD::

                                .SBTTL          DEVICE INDEPENDENT SECTION

046140                                  GPRML   DPLX,0,1,YES
046140  000130                                                    .WORD   T$CODE
046142  046212                                                    .WORD   DPLX
046144  000001                                                    .WORD   1


                                .SBTTL          DEVICE DEPENDENT SECTION

046146                                  GPRMA   CSRADR,2,0,160000,177776,YES
046146  001031                                                    .WORD   T$CODE
046150  046243                                                    .WORD   CSRADR
046152  160000                                                    .WORD   T$LOLIM
046154  177776                                                    .WORD   T$HILIM
046156                                  GPRMA   VECTOR,4,0,300,776,YES
046156  002031                                                    .WORD   T$CODE
046160  046271                                                    .WORD   VECTOR
046162  000300                                                    .WORD   T$LOLIM
046164  000776                                                    .WORD   T$HILIM
046166                                  GPRMD   PRIOR,6,0,340,4,7,YES
046166  003032                                                    .WORD   T$CODE
046170  046324                                                    .WORD   PRIOR
046172  000340                                                    .WORD   340
046174  000004                                                    .WORD   T$LOLIM
046176  000007                                                    .WORD   T$HILIM
                                ;       GPRMD   DEVPRM,10,D,17,0,15.,YES
046200                                  GPRMD   OPTN,12,0,7,0,7,YES
046200  005032                                                    .WORD   T$CODE
046202  046352                                                    .WORD   OPTN
046204  000007                                                    .WORD   7
046206  000000                                                    .WORD   T$LOLIM
046210  000007                                                    .WORD   T$HILIM
                                ;       GPRMD   BAUD,14,0,7,0,7,YES
                                ;       GPRMD   LININ,16,0,7,0,7,YES
```

```
      046212                                   ENDHRD

      046212                                                                     L10023:  .EVEN

                                       .NLIST  BEX

                                       ;DEVICE INDEPENDENT QUESTIONS

      046212  052506  046114  042040  DPLX:   .ASCIZ  /FULL DUPLEX OPERATION : /

                                       ;DEVICE DEPENDENT QUESTION

      046243     104  053105  041511  CSRADR: .ASCIZ  /DEVICE CSR ADDRESS : /
      046271     111  052116  051105  VECTOR: .ASCIZ  /INTERRUPT VECTOR ADDRESS: /
      046324  047111  042524  051122  PRIOR:  .ASCIZ  /INTERRUPT PRIORITY : /
      046352  042504  044526  042503  OPTN:   .ASCIZ  /DEVICE OPTION TYPE : (0=DMC,5=DMR-DMC MODE ,7=DMR)/

                                       .LIST   BEX
              046436                           .EVEN
```

```
                                ;.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
                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

046436                          $PATCH:
046436  000030                          .BLKW   30


046516                                  LASTAD
                                                                        .EVEN
046516  000000                                                          .WORD   0
046520  000000                                                          .WORD   0
046522                          L$LAST::
046522                                  ENDMOD

                        000001  .END
```

```
ABO    = 000026            2301#  4818
ACT    = 000003            2253#  4944    5399    5479    5723
ACTATV  037754             5165   5399#
ACTBCR  037560             5183   5360#
ACTCHK  040170             5145   5445#
ACTCLB  037102             5249   5263#
ACTCLP  040302             5179   5473#
ACTCLR  036536             5143   5196#
ACTCOP  037400             5153   5323#
ACTCRC  040204             5174   5451#
ACTCSE  036672             5148   5225#
ACTCST  037020             5149   5251#
ACTDLL  040022             5169   5413#
ACTDME  037326             5185   5304    5307#
ACTDMQ  037320             5186   5306#
ACTDMS  037276             5184   5301#
ACTDMX  037334             5308#
ACTECH  040100             5173   5429#
ACTEQO  037522             5157   5349#
ACTEXT  036622             5189   5214#
ACTHLP  036556             5147   5202#
ACTLIS  040012             5168   5410#
ACTLLP  040312             5180   5475#
ACTLPX  040330             5470   5472    5474    5476    5479#
ACTLXX  040372             5443   5464    5467    5480    5489#
ACTMEX  037746             5342   5358    5380    5385    5391    5394    5396#
ACTME1  037702             5369   5371    5373    5375    5377    5384#
ACTMOP  040262             5177   5469#
ACTMOS  040212             5188   5454#
ACTMS0  037602             5158   5368#
ACTMS1  037610             5159   5370#
ACTMS2  037620             5160   5372#
ACTMS3  037630             5161   5374#
ACTMS4  037640             5162   5376#
ACTMS5  037650             5163   5378#
ACTMS6  037666             5164   5381#
ACTM2X  040050             5400   5408    5411    5414    5417    5421#
ACTNO   040070             5172   5426#
ACTNUF  036526             5182   5193#
ACTNUL  036534             5142   5194#
ACTNUM  037410             5154   5326#
ACTOPM  037502             5155   5344#
ACTPAS  037764             5166   5402#
ACTPRO  040220             5175   5457#
ACTPRT  036632             5187   5216#
ACTQFG  040224             5446   5449    5452    5455    5459#
ACTREC  040004             5167   5407#
ACTREX  027576             3634   3660#
ACTRHL  027532             3633   3646#
ACTRLG  027606             3635   3664#
ACTRLP  040322             5181   5477#
ACTRNF  027522             3639   3642#
ACTRNL  027530             3632   3643#
ACTRPS  040252             5176   5466#
ACTRUN  036646             5146   5220#
ACTSEX  037712             5190   5388#
```

```
ACTSHO  036546          5144    5199#
ACTSHW  037142          5236    5255    5274#   5293
ACTSTE  037342          5150    5313#
ACTSTS  040176          5156    5448#
ACTSTT  037352          5151    5316#
ACTSTX  037360          5314    5317#
ACTSWE  027622          3636    3669#
ACTSWF  027764          3637    3690#
ACTSWO  030076          3638    3707#
ACTSZE  037370          5152    5320#
ACTTAL  040042          5171    5419#
ACTTLP  040272          5178    5471#
ACTTRA  040032          5170    5416#
ADDCC   031712          4067#   5070    5116
ADDC1   032006          4069    4084#
ADR   = 000020 G        2226#
ALCK    040764          2797    5633#
ALCK1   041076          5688    5697#   5741
ALCK2   041142          5698    5707#   5773
ALCK2A  041362          5738    5743#
ALCK3   041406          5710    5745    5748#
ALCK3A  041500          5755    5761#
ALCK3B  041512          5762    5764#
ALCK3C  041506          5759    5763#   5770
ALCK4   041534          5749    5768#
ALCK4A  041542          5767    5771#
ALCK4B  041554          5772    5774#
ALCK5   041026          5687#   5763
ALLTR   041026          5565    5588    5612    5638    5686#
ASSEMB= 000010          1996
ATVMOD= 000027          2355#   2961
BACC  = 000001          2409#   6685
BAD     007447          2762#   3169    5849*
BADCHR= 000051          2373#   3058
BASE    017370          3163#   3674    3676    3682    3684    3694    3696    3700    3702    3731    3733    3739    3741
                        6245
BASM1   016257          3163#   4011
BASM1A  016224          3163#
BASM2   016250          3163#   4030
BASM3   016241          3163#   4022
BDATA   007456          2770#   3676*   3677*   3684*   3685*   3696*   3702*   3733*   3734*   3741*   3742*   3773
BEND    007454          2765#   3674*   3675*   3682*   3683*   3694*   3695*   3700*   3701*   3710*   3712*   3713    3731*
                        3732*   3739*   3740*   3790
BIT0  = 000001 G        2199#   2273    2397    2409    6715
BIT00 = 000001 G        2188#   2199
BIT01 = 000002 G        2187#   2198
BIT02 = 000004 G        2186#   2197
BIT03 = 000010 G        2185#   2196
BIT04 = 000020 G        2184#   2195
BIT05 = 000040 G        2183#   2194
BIT06 = 000100 G        2182#   2193
BIT07 = 000200 G        2181#   2192
BIT08 = 000400 G        2180#   2191
BIT09 = 001000 G        2179#   2190
BIT1  = 000002 G        2198#   2274    6718
BIT10 = 002000 G        2178#   2413    6741
```

```
BIT11 = 004000 G        2177#   2411    6744
BIT12 = 01C000 G        2176#   6747
BIT13 = 020000 G        2175#
BIT14 = 040000 G        2174#   2400    2410    6772
BIT15 = 100000 G        2173#   6772
BIT2  = 000004 G        2197#   2275    2395    2414    6264    6267    6692
BIT3  = 000010 G        2196#   2276    2396    6264    6270    6721
BIT4  = 000020 G        2195#   2277    6724
BIT5  = 000040 G        2194#   2278    2398    2406    6703
BIT6  = 000100 G        2193#   2415    6727
BIT7  = 000200 G        2192#   2399    2407    2408    6730
BIT8  = 000400 G        2191#   2412    6733
BIT9  = 001000 G        2190#   2401    6736
BLDBEX  032120          4128    4132#
BLDBUF  032010          4110#   4935    4942    5071    5117    5270    5539
BLDB1   032020          4115#
BLDB2   032062          4123#   4130
BLDB3   032100          4126#   4131
BOE   = 000400 G        2230#
BSEL0   012364          3101#   6243*   6262*   6276*   6282*   6331*   6370*   6425*   6817*
BSEL1   012366          3102#   4724*   4725*   6213*   6216*
BSEL2   012370          3104#
BSEL3   012372          3105#   4728*   4729*
BSEL4   012374          3107#
BSEL5   012376          3108#   4732*   4733*
BSEL6   012400          3110#   6267*   6270*
BSEL7   012402          3111#   4736*   4737*
BTHEAD  030476          3767    3796#
BTUP  = 002000          2315#   6700    6834    6836
BUFEX   015227          3163#   5043    5084
BUFLIM= 001000          2241#   2711    2712    2713    4068    4080    4199    5040    5081    5263    5535
BYTBIT  007406          2742#   4016    5302*   5306*
CABLE = 000002          2260#
CBLLOP= 000045          2369#   3079
CCURAD  007344          2722#   4211*   4219    4223*   4931*   4939    5115    5120*   5248*
CDGLM   025471          3163#   6749
CHECK = 000003          2335#   3009
CLEAR = 000001          2333#   2947    5033    5196
CLIACT  036346          4989    5134#
CLIALN= 000007          2326#   3056
CLIALP= 000006          2325#   2995    3005    3061
CLIBCR  012740          3163#   5361
CLIBDL  012633          3163#   5484
CLIBIF= 000003          2322#
CLIBR = 000002          2321#   2944    2946    2948    2951    2953    2962    2964    2966    2968    2970    2973    2975
                        2990    2992    3002    3010    3017    3019    3023    3025    3028    3035    3037    3039    3041
                        3043    3045    3047    3049    3054    3057    3064    3066    3071    3078    3080    3082    3084
                        3089    3615
CLIBRX  012613          3163#   4449
CLIDEC= 000011          2328#   3070    3088
CLIERM  012516          3163#   3579    4995
CLIERR= 000000          2319#   2954    3058    3625
CLIEXI= 000001          2320#   2936    2938    2940    2942    2981    2983    3094    3607    3609    3611    3613    3618
                        3620    3624    3626
CLINBG  012571          3163#   4492
CLINPS  012675          3163#   5435
```

```
CLINUF  012546          3163#   3589    5004
CLINUM= 000005          2324#
CLIOCT= 000010          2327#   2997    2999    3623
CLIPW   013036          3163#   5501
CLIRAC  027464          3573    3627#
CLIRT   027266          3572    3604#
CLISEO  013005          3163#   5331
CLISPA= 000004          2323#   2934    2958    2979    2988    2996    3055    3605
CLISTR= 000012          2329#   2937    2939    2941    2943    2945    2947    2950    2952    2959    2961    2963    2965
                        2967    2969    2972    2974    2980    2982    2989    2991    3007    3009    3016    3018    3022
                        3024    3027    3034    3036    3038    3040    3042    3044    3046    3048    3063    3065    3077
                        3079    3081    3083    3608    3610    3612    3614    3617    3619    3621
CLITRE  010664          2930#   4988
CLI$PM  012502          3163#   4982
CLI$RP  012510          3163#   3566
CLKBR   007524          2805#   4675*
CLKCSR  007522          2804#   3368*   3389*   4646    4664*   4665*   4666*   4676*   4811*   4895*
CLKEN   007532          2808#   3389    4654*   4667*   4676    4895
CLKHZ   007530          2807#   3371    3386    3516    4680    4698
CLKINT  026230 G        3366#   4748
CLKSET  026204          3321#   4653    4663
CLKVEC  007526          2806#   4644*   4674*   4749
CLRAW   044704          6256    6271    6278    6295    6335    6375    6430    6555#   6820
CLRA1   044724          6558#   6563
CLRA2   044746          6559    6566#
CLRA3   044716          6557#   6576
CMDBUF  003060          2574#   3564    3571    4980    4987
CMPBUF  005562          2713#   4197*   4211    4931    5247
CMPPTR  007336          2719#   4205*   4206*   4219*   4220*   4221*   4225*   4925*   4926*   4938    5098*   5099*   5113*
                        5114    5118*   5229*   5230*   5231    5244*   5245*   5246    5514*   5515*   5516    5813
CMPSEX  042030          5811    5891#
CMPSR   041554          5760    5810#
CMPS1   041714          5842#   5860
CMPS2   041720          5845#
CMPS3   041606          5816#   5872
CMPS5   042016          5869#
CMPS5A  042022          5862    5870#
CMPS6   041764          5843    5847    5858#
CMPS7   041702          5831    5840#
CMPTOT  007340          2720#   4214*   4217    4226*   4943*   5093*   5100    5119*   5233    5237*   5498    5523
CMSG0 = 000020          2348#   3036
CMSG1 = 000021          2349#   3034
CMSG2 = 000022          2350#   3038
CMSG3 = 000023          2351#   3040
CMSG4 = 000024          2352#   3044
CMSG5 = 000025          2353#   3042
CMSG6 = 000026          2354#   3046
CONOTM  007442          2759#   3212    3269    6001*   6018*   6026*   6031*   6709*   6714*   6717*   6720*   6723*   6726*
                        6729*   6732*   6735*   6738*   6743*   6746*   6749*
CONTIN  007444          2760#   6294*   6334
CPTR    007416          2749#   4113    4117*   4934*   4938*   5068*   5072    5114*   5118    5246*   5259*   5537*   5564*
                        5584*   5609*   5634*   5699    5704*   5716    5764*   5771    5812*   5817    5821*   6149*
CPTRR   007414          2748#   5561*   5587*   5610*   5636*   5689    5694*   5722*   5727    5735*   5747*   5768    5813*
                        5823    5826*   6103*
CR      020000          3163#   3500
CRC   = 000040          2364#
```

```
CRCB   = 000020   2277#  5451
CRX    = 000040   2310#  6433  6473  6494  6512  6767
CSHEXP= 000006   2338#  2980
CSHTRN= 000007   2339#  2982
CSRADR  046243   6876   6905#
CTQTCC  007342   2721#  4213*  4221  4223  4224*  4229*  4918*  5080  5091  5121*  5238*
CTS    = 000004   2395#  2829
CTSFM   025505   3163#  6746
CTX    = 000020   2309#  6433  6473  6475  6493  6762
CURADD  007420   2750#  4115  4120  4122  4132*  4933*  4939*  5069*  5074  5115*  5120  5264*  5536*
                 5948*  5954*  5989  5990
CURCC   007412   2747#  4067  4079  4082*  4083  4116  4121  4132  4929*  4941*  5266*  5338*  5351*
                 5379*  5382*  5384*  5535*  5949*  5955*  5991  5992
C$AU   = 000052   1996#  4875
C$AUTO= 000061   1996#  4799
C$BRK  = 000022   1996#  6221  6452  6561  6621  6823
C$BSEG= 000004   1996#
C$BSUB= 000002   1996#
C$CEFG= 000045   1996#
C$CLCK= 000062   1996#  4649  4659
C$CLEA= 000012   1996#  4832
C$CLOS= 000035   1996#
C$CLP1= 000006   1996#
C$CVEC= 000036   1996#
C$DCLN= 000044   1996#  4617
C$DODU= 000051   1996#
C$DRPT= 000024   1996#
C$DU   = 000053   1996#  4853
C$EDIT= 000003   1996#  2058
C$ERDF= 000055   1996#
C$ERHR= 000056   1996#  5971
C$ERRO= 000060   1996#
C$ERSF= 000054   1996#
C$ERSO= 000057   1996#  5834  5852  5865  6231  6444  6463  6484  6503  6572  6614  6674  6753
C$ESCA= 000010   1996#  6682
C$ESEG= 000005   1996#
C$ESUB= 000003   1996#
C$ETST= 000001   1996#  6841
C$EXIT= 000032   1996#  4775  4824  4968  5031
C$GETB= 000026   1996#
C$GETW= 000027   1996#
C$GMAN= 000043   1996#  3562  4678  4978  5933  6081
C$GPHR= 000042   1996#  4713
C$GPLO= 000030   1996#
C$GPRI= 000040   1996#
C$INIT= 000011   1996#  4783
C$INLP= 000020   1996#
C$MANI= 000050   1996#  4962
C$MEM  = 000031   1996#
C$MSG  = 000023   1996#  3180  3193  3207  3229  3250  3264  3279
C$OPEN= 000034   1996#
C$PNTB= 000014   1996#  3176  3189  3203  3218  3225  3239  3246  3260  3275
C$PNTF= 000017   1996#  3503  3511  3554  3582  3592  3652  3720  3770  3788  4014  4025  4033
                 4077  4452  4495  4693  4908  4953  4998  5007  5047  5062  5088  5108  5208
                 5289  5334  5364  5438  5487  5504  5965  6049  6142  6165
C$PNTS= 000016   1996#  3809  3832  3846  3859  3872  3880  3906  3922  3937  3955  3975  4276
```

CZCLKCO DMR,DMC-11 DATA COMM. LINK TEST MACY11 30A(1052) 23-MAR-82 16:45 PAGE 176
CZCLKC.P11 19-MAR-82 18:32 CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    4306
C$PNTX= 000015      1996#
C$QIO = 000377      1996#
C$RDBU= 000007      1996#  4671
C$REFG= 000047      1996#  4622   4627   4632   4638
C$RESE= 000033      1996#
C$REVI= 000003      1996#  2057
C$RFLA= 000021      1996#
C$RPT = 000025      1996#  4585
C$SEFG= 000046      1996#
C$SPRI= 000041      1996#  4773   4814
C$SVEC= 000037      1996#  4751   4761   4768
C$TPRI= 000013      1996#
DAM     020233      2870   3163#
DATCKB= 000002      2274#  4285   5445   5496   5583   5719   5766   5810   5947   6075   6137
DATCKM  025424      3163#  6717
DCD   = 000001      2397#  2831
DCK   = 000014      2296#  3464
DCLFLG  007464      2773#  4613   4615*  5029*
DDCMRM  025373      3163#  6723
DDCSRM  025326      3163#  6732
DDE   = 000022      2299#  3472
DER   = 000010      2294#  3450
DEV1    010654      2910#  3865*  3876   3884*  3886*  4257   4955*  5294*
DEV2    010656      2911#  3866*  3875   3885*  3887*  4260   4956*  5295*
DEV3    010660      2912#  3888*  4267   4957*  5296*
DEV4    010662      2913#  3889*  4281   4285   4289   4293   4958*  5297*
DFPTBL  002130 G    2124#
DIAGMC= 000000      1996
DISCOM  025346      3163#  6729
DLE   = 000020      2298#  3468
DLL     042114      2798   5932#
DLLAB   025214      3163#  5973   6004
DLLCM   015034      3163#  5962
DLLEA   042272      5969#  6006
DLLE1   042550      5997   6016#
DLLE2   042416      5995#  6015
DLLE3   042630      6025   6029#
DLLE4   042620      6026#
DLLE5   042732      6014   6051#
DLLE5A  042650      6030   6035#
DLLE5B  042704      6038   6040   6043#
DLLE6   042502      5999   6008#
DLLE7   042450      6002#  6019   6027   6032
DLLE8   042570      6017   6020#
DLLGA = 000400      2313#  5956   5996   6012
DLLIND  010544      2866#  6041
DLLMOD= 000033      2359#  2969
DLLM1   002647      2437   2521#  5948
DLLM1C  002172      2437#  5949
DLLM1E  002654      2437   2526#
DLLM2   002654      2438   2528#  5954
DLLM2C  002174      2438#  5955   6013
DLLM2E  003060      2438   2568#
DLLPRI  042246      5960#
DLLQ1   013126      3163#  5937
```

```
DLM     020225            2868    3163#
DLTXRX  042306            5950    5957    5980#
DLVM    020251            2874    3163#
DMC   = 000000            2282#
DMCEND  003462            2669#   3673    3693    3730
DMCIND  003432            2657#   3671    3692    3728
DMCM    020242            2872    3163#
DMC002  023604            2659    3163#
DMC003  023625            2660    3163#
DMC004  023662            2661    3163#
DMC005  023723            2662    3163#
DMC006  023756            2663    3163#
DMC007  024013            2664    3163#
DMC010  024050            2665    3163#
DMC011  024103            2666    3163#
DMC012  024125            2667    3163#
DMC013  024147            2668    3163#
DMC377  024206            2669    3163#
DMFMT   030547            3785    3796#
DMPE  = 000053            2375#   2999
DMPM    020255            2875    3163#
DMPQ  = 000054            2376#   3001
DMPS  = 000052            2374#   2997    5022    5303
DMRC6 = 000004            2283#   6211
DMRC7 = 000005            2284#
DMREND  003430            2654#   3681    3699    3738
DMRIND  003250            2598#   3679    3698    3736
DMRRUN= 001000            2314#   6299    6705    6707
DMR000  020342            2598    3163#
DMR001  020402            2599    3163#
DMR002  020432            2600    3163#
DMR003  020467            2601    3163#
DMR004  020532            2602    3163#
DMR005  020566            2603    3163#
DMR006  020620            2604    3163#
DMR007  020663            2605    3163#
DMR010  020717            2606    3163#
DMR011  020751            2607    3163#
DMR012  021003            2608    3163#
DMR013  021035            2609    3163#
DMR014  021065            2610    3163#
DMR015  021114            2611    3163#
DMR016  021146            2612    3163#
DMR017  021176            2613    3163#
DMR020  021226            2614    3163#
DMR021  021255            2615    3163#
DMR022  021307            2616    3163#
DMR023  021333            2617    3163#
DMR024  021365            2618    3163#
DMR025  021410            2619    3163#
DMR026  021463            2620    3163#
DMR027  021513            2621    3163#
DMR030  021544            2622    3163#
DMR031  021627            2623    3163#
DMR032  021664            2624    3163#
DMR033  021721            2625    3163#
```

```
DMR034  021756          2626    3163#
DMR035  022042          2627    3163#
DMR036  022107          2628    3163#
DMR037  022154          2629    3163#
DMR040  022212          2630    3163#
DMR041  022245          2631    3163#
DMR042  022303          2632    3163#
DMR043  022351          2633    3163#
DMR044  022405          2634    3163#
DMR045  022441          2635    3163#
DMR046  022472          2636    3163#
DMR047  022540          2637    3163#
DMR050  022602          2638    3163#
DMR051  022630          2639    3163#
DMR052  022664          2640    3163#
DMR053  022711          2641    3163#
DMR054  022744          2642    3163#
DMR055  022766          2643    3163#
DMR056  023041          2644    3163#
DMR057  023114          2645    3163#
DMR060  023136          2646    3163#
DMR061  023170          2647    3163#
DMR062  023222          2648    3163#
DMR063  023272          2649    3163#
DMR064  023342          2650    3163#
DMR065  023376          2651    3163#
DMR066  023432          2652    3163#
DMR067  023475          2653    3163#
DMR177  023540          2654    3163#
DMR6   = 000006         2285#   6214
DMR7   = 000007         2286#
DMSGAD  002176          2442#   4124
DMSGCT  002150          2427#   4123
DMUNKN  020322          2657    2658    3163#
DMVM    020316          2883    3163#
DNM     020246          2873    3163#
DOW    = 000004         2254#   5413    6285
DPLX    046212          6866    6905#
DPM     020217          2866    3163#
DQM     020230          2869    3163#
DSR    = 000010         2396#   2830
DTEM    020261          2876    3163#
DUM     020222          2867    3163#
DUMEX   031710          4036    4042#
DUMPSR  031554          4007#   5024
DUM1    031646          4017    4028#
DUM2    031670          4027    4035#
DUM3    031604          4016#   4040
DUM4    031560          4008#   4039
DUPM    020236          2871    3163#
DVBTUP  045774          4810    5902    6814#
DVEM0   024252          3163#   6566    6574
DVEM1   024340          3163#   6608    6616
DVEM3   024424          3163#   6225    6233    6828
DVEM4   024510          3163#   6438    6446
DVEM5   024602          3163#   6456    6465
```

```
DVEM6   024674            3163#   6668    6676
DVEM7   024751            3163#   6710    6755
DVEM8   025036            3163#   6498    6505
DVEM9   025125            3163#   6479    6486
DVI   = 000012            2295#   3455
DVINEX  044046            6298    6300#
DVINIT  043364            5533    6203#   6236
DVINS   044664 G          4758    6522#
DVIN1   043526            6219    6238#
DVIN10  043722            6269    6271#
DVIN2   043444            6215    6218#   6223
DVIN3   043632            6254    6256#
DVIN4   044016            6288    6291    6294#
DVIN5   044002            6286    6290#
DVIN6   043426            6212    6214#
DVIN7   043600            6249    6251#
DVIN8   043744            6261    6282#
DVIN9   043704            6266    6268#
DVM     020265            2877    3163#
DVMODS  044050            3480    6331#
DVOUTS  044674 G          4765    6529#
DVRCC   007364            2732#   5693*   5712    5984*   6021    6146*   6157    6159*   6160*   6374    6510*
DVRCT   007366            2733#   5562*   5635*   5721*   5725*   5743*   5744    5814*   5871*
DVREX   044162            6367    6376#
DVRXA   007362            2731#   5691*   5711    5982*   6020    6144*   6156    6159    6373    6509*
DVRXQ   044104            5695    5986    6151    6365#
DVTCC   007350            2725#   5703*   5751    5991*   6009    6013    6100*   6108    6429    6491*
DVTCT   007352            2726#   5585*   5608*   5633*   5753*   5758    5761
DVTREX  044662            6495    6513#
DVTRX1  044654            6508    6512#
DVTR3   044242            6423    6431#   6448
DVTR4   044442            6434    6475#
DVTR4A  044522            6478    6490#
DVTR4B  044544            6489    6493#
DVTR5   044552            6476    6494#
DVTR5A  044632            6497    6509#
DVTXA   007346            2724#   5701*   5750    5989*   6008    6097*   6107    6428    6490*
DVTXRX  044164            5707    5995    6105    6154    6422#
DZM     020270            2878    3163#
ECHO  = 000037            2363#   3018
ECHOB = 000004            2275#   4289    5421    5431    5441    5714
EDABO   016213            2862    3163#
EDDCK   015774            2857    3163#
EDDDE   016107            2860    3163#   5854    5867
EDDER   015757            2855    3163#
EDDLE   016052            2859    3163#   5836
EDDVI   016024            2856    3163#
EDEOP   016142            2861    3163#
EDMOS   016156            3163#
EDRXC   015731            2854    3163#
EDRXQ   015704            2853    3163#
EDTXC   015655            2852    3163#
EDTXQ   015631            2851    3163#
EFM11   015554            3163#   3200
EFM2    015457            3163#   3186
EF.CON= 000036 G          2206#   4631
```

```
EF.NEW= 000035 G        2207#   4637
EF.PWR= 000034 G        2208#
EF.RES= 000037 G        2205#   4626
EF.STA= 000040 G        2204#   4621
EMSG0   002221          2428    2454#
EMSG1   002222          2429    2456#
EMSG2   002223          2430    2458#
EMSG3   002224          2431    2460#
EMSG4   002324          2432    2474#
EMSG5   002416          2433    2488#
EMSG6   002517          2434    2502#
EMSG8   002647          2436    2516#
ENADD   007404          2741#   4035    5307*
ENDEVT  031432          3811    3821    3825    3942#
ENDIT   034530          4635    4745#
ENDQO = 000017          2347#   3053
EOP   = 000024          2300#   3476
ERRCNT  007400          2739#   3479    3482    4821    5529*   5832*   5850*   5863*   5895    6229*   6442*   6461*   6570*
                        6612*   6672*   6751*   6832*
ERR1    025630 G        3166#   5855
ERR10   025720 G        3196#   5837
ERR13   026110 G        3253#   6234    6447    6466    6487    6506
ERR14   026142 G        3267#   5974
ERR2    025672 G        3183#   5868
ERR8    025750 G        3210#   6756
ERR9    026032 G        3232#   6575    6617    6677
ERX   = 000100          2311#   5563    5611    5637    5739    5757    5946    6148    6496
ETX   = 000200          2312#   5586    5637    5740    5756    5994    6011    6102    6114    6477
EVL   = 000004 G        2224#
EVMCTS  017255          2840    3163#
EVMDCD  017265          2842    3163#
EVMDSR  017261          2841    3163#
EVMOCG  017135          3163#
EVMOHD  017160          3163#   3952
EVMOST  017240          3163#   3972
EVMRI   017275          2844    3163#
EVMRTS  017271          2843    3163#
EVMSQD  017301          2845    3163#
EVMTM   017305          2846    3163#
EVTADD  010616          2890#   3850*   3855    3896*   3902    3912*   3918    3927*   3933
EVTBCT  010620          2891#   3851*   3854    3897*   3901    3913*   3917    3928*   3932
EVTEND  010454          2821#   3524    3818    4702
EVTF0   016325          3163#   3829
EVTF1   016420          3163#   3843
EVTF2   016447          3163#   3856
EVTF3   016521          3163#   3869
EVTF3C  016533          3163#   3236    3257    3877
EVTF3D  016550          3163#   3215    3272
EVTF4   016572          3163#   3919
EVTF4A  016674          3163#   3934
EVTF4B  016772          3163#   3903
EVTF5A  017051          3163#   3173
EVTLOG  007552          2819    2820#   3528    3803    3816    4699
EVTLST  010514          2851#   3839
EVTMIN  010612          2888#   3837*   3842
EVTPTR  007550          2819#   3514    3529*   3802    3823    4700*
```

```
EVTSEC  010610          2887#  3836*  3841
EVTTCK  010614          2889#  3835*  3840
EVTTMP  010622          2892#  3864*  3868   3898*  3900   3914*  3916   3929*  3931
EXIT  = 000057          2379#  2941   4815   5027   5214
EXITCL  034732          4816   4823#
E$END = 002100          1996#
E$LOAD= 000035          1996#  2081
FACSIM  032134          4194#  5392
FHDPLX  007474          2781#  4718*  6290
FLAG    007502          2791#  5526*  5563*  5586*  5611*  5637*  5687   5697   5709   5739*  5740*  5746*  5748
                        5756*  5757*  5765*  5946*  5956*  5981*  5994*  5996   5998   6011*  6012*  6016   6102*
                        6114*  6148*  6242*  6299*  6366   6369*  6422   6424*  6433   6453   6460*  6469   6473
                        6475   6477   6492*  6493*  6494   6496   6511*  6512*  6523*  6530*  6622   6627   6629*
                        6665*  6700   6705   6707*  6762*  6767*  6824   6834*  6836*
F$AU  = 000015          1996#  4863   4874
F$AUTO= 000020          1996#  4794   4798
F$BGN = 000040          1996#  1998   3166   3183   3196   3210   3232   3253   3267   3284   3366   4574   4594
                        4610   4775   4794   4808   4824   4841   4847   4863   4869   4890   4968   5031   6522
                        6529   6682   6840   6858   6942
F$CLEA= 000007          1996#  4808   4831
F$DU  = 000016          1996#  4841   4852
F$END = 000041          1996#  1998   3181   3194   3208   3230   3251   3265   3280   3282   3393   4586   4775
                        4784   4800   4824   4833   4845   4854   4867   4876   4890   4968   5031   6527   6534
                        6682   6840   6842   6904   6942
F$HARD= 000004          1996#  6858   6902
F$HW  = 000013          1996#  2122   2153
F$INIT= 000006          1996#  4610   4782
F$JMP = 000050          1996#  3282   4775   4824   4845   4867   4968   5031
F$MOD = 000000          1996#  1998   6942
F$MSG = 000011          1996#  3166   3179   3183   3192   3196   3206   3210   3228   3232   3249   3253   3263
                        3267   3278
F$PROT= 000021          1996#  4594   4601
F$PWR = 000017          1996#
F$RPT = 000012          1996#  4574   4584
F$SEG = 000003          1996#
F$SOFT= 000005          1996#
F$SRV = 000010          1996#  3366   3391   6522   6525   6529   6532
F$SUB = 000002          1996#
F$SW  = 000014          1996#
F$TEST= 000001          1996#  4891   6840
GETCL   035354          4964   4975#  5000   5009   5013   5016   5018   5025   5034   5036   5049   5064   5078
                        5090   5110   5124   5506
GETPRM  034310          4641   4707#  4716
GETRCL  027074          3557#  3584   3594   3598
GOOD    007446          2761#  3171   5848*
GTRA2   034762          4898#
GTRA3   035024          4901   4911#
GTRA4   035032          4916#
GTRA5   035270          4912   4955#  5903   5967   5976
GTREX   040534          5526#
GTRX2   040570          5535#  5901   6116   6172
GTR9    040400          4971   5021   5494#
GTXRXB  034762          4897#
G$CNTO= 000200          1996#
G$DELM= 000372          1996#
G$DISP= 000003          1996#
```

```
G$EXCP= 000400          1996#
G$HILI= 000002          1996#
G$LOLI= 000001          1996#
G$NO  = 000000          1996#   3565    4981    5936    6084
G$OFFS= 000400          1996#   3565    4681    4981    5936    6084    6865    6875    6880    6885    6893
G$OFSI= 000376          1996#   3565    4681    4981    5936    6084    6865    6875    6880    6885    6893
G$PRMA= 000001          1996#   6875    6880
G$PRMD= 000002          1996#   3565    4681    4981    5936    6084    6885    6893
G$PRML= 000000          1996#   6865
G$RADA= 000140          1996#   3565    4981    6084
G$RADB= 000000          1996#
G$RADD= 000040          1996#   4681
G$RADL= 000120          1996#   6865
G$RADO= 000020          1996#   5936    6875    6880    6885    6893
G$XFER= 000004          1996#
G$YES = 000010          1996#   4681    6865    6875    6880    6885    6893
HALFDB= 002000          2413#   6287    6292
HELP  = 000000             1#   1996    2010    2100    2112    2126    2157    2161    2168    3121    3142    3159    3163
                        3293    4576    4580    4582    4602    4612    4778    4796    4810    4827    4843    4848    4865
                        4870    4887    4888    4892    6783    6788    6861    6908    6922    6927    6935
HELPDC= 000000             1#       3    2099    2135    2156    2389    2403    3095    3122    3143    3163    3209    4720
                        4756    6177    6182    6308    6340    6380    6517    6520    6528    6870    6905
HLP   = 000005          2337#   2935    2937    5015    5212
HLPEND  003230          2587#   5210
HLPF    013231          3163#   3649    5205
HLPTAB  003210          2579#   5202
HLP0    013153          3163#   4950
HLP1    013236          2579    3163#
HLP2    013251          2580    3163#
HLP3    013366          2581    3163#
HLP3A   013453          2582    3163#
HLP4    013500          2583    3163#
HLP4A   013557          2584    3163#
HLP5    013635          2585    3163#
HLP6    013725          2586    3163#
HOE   = 100000 G        2237#
IBE   = 010000 G        2234#
IDU   = 000040 G        2227#
IEO   = 000100          2415#   6251    6818
IER   = 020000 G        2235#
INDEX   007450          2763#   3671*   3672*   3679*   3680*   3692*   3698*   3728*   3729*   3736*   3737*   3772
INDEXE  007452          2764#   3673*   3681*   3693*   3699*   3730*   3738*   3776    3778
ININT = 000001          2305#   6453    6460    6523    6627    6629
INIT1   033766          4614    4619#
INTPRI  012410          3116#   4742*   4757    4764
INVEC   012404          3114#   4739*   4759
ISR   = 000100 G        2228#
IXE   = 004000 G        2233#
I$AU  = 000041          1996#   4863#   4876#
I$AUTO= 000041          1996#   4794#   4800#
I$CLN = 000041          1996#   4808#   4824    4833#
I$DU  = 000041          1996#   4841#   4854#
I$HRD = 000041          6858#   6904#
I$INIT= 000041          1996#   4610#   4775    4784#
I$MOD = 000041          1996#   1998#   6942#
I$MSG = 000041          1996#   3166#   3181#   3183#   3194#   3196#   3208#   3210#   3230#   3232#   3251#   3253#   3265#
```

```
                    3267#   3280#
I$PROT= 000040      1996#   4594#
I$PTAB= 000041      1996#
I$PWR = 000041      1996#
I$RPT = 000041      1996#   4574#   4586#
I$SEG = 000041      1996#   4890
I$SETU= 000041      1996#
I$SRV = 000041      1996#   3366#   3393#   6522#   6527#   6529#   6534#
I$SUB = 000041      1996#   4890
I$TST = 000041      1996#   4890#   4968    5031    6682    6840#   6842#
J$JMP = 000167      1996#   3282    4845    4867
KDPM    020303      2880    3163#
KDZM    020307      2881    3163#
KEYWD1  003202      2575#   3596    3656*   3660*   3665*   4612*   4815    5012    5015    5017    5019    5022    5027
                    5033    5035    5037    5196*   5199*   5212*   5214*   5216*   5220*   5235    5254    5303*   5313*
                    5316*   5388    5393*
KLM     020313      2882    3163#
LCLKEN= 000100      2267#   4654
LGDVE   026434      3448#   6005    6228    6441    6459    6482    6501    6569    6611    6671    6713    6831
LIS   = 000006      2256#   5410    6115
LISCK   043154      2800    6137#
LISCKA  043202      6144#   6168    6170
LISMOD= 000032      2358#   2967
LISP    014757      3163#   6139
LMDLOP= 000046      2370#   3081
LNCNT   007372      2736#   3495    3497*   3506*   5530*
LOE   = 040000 G    2236#
LOGCMD  026554      3470#   5869
LOGCML  026536      3466#   5838
LOGCMP  026520      3462#   5828
LOGDVI  026452      3453#   5532
LOGEOP  026572      3474#   5896
LOGEX   027044      3484    3531#
LOGRXC  026424      3445#   5713    6022    6158
LOGRXQ  026406      3440#   5696    5987    6152
LOGS1   026610      3433    3438    3443    3447    3479#
LOGS2   027036      3525    3529#
LOGS3   026642      3451    3460    3465    3469    3473    3477    3488#
LOGS4   026716      3496    3505#
LOGS5   026742      3490    3492    3513#   4822*
LOGTXC  026370      3435#   5752    6010    6109
LOGTXQ  026352      3430#   5705    5993    6101
LOGUNT  007460      2771#   4705*   4707*   4708    4712
LOOPS   003532      2692#   4266
LOSDAM  025361      3163#   6726
LOT   = 000010 G    2225#
LP0     014641      2692    2858    3163#   4265    6714
LP00    014642      3163#   4262
LP1     014651      2693    3163#
LP2     014662      2694    3163#
LP3     014670      2695    3163#
LP4     014703      2696    3163#
LULOOP= 004000      2411#   6252    6255
L$ACP   002110 G    2088#
L$APT   002036 G    2046#
L$AU    034746 G    2073    4863#
```

```
L$AUT   002070 G        2072#
L$AUTO  034646 G        2089    4794#
L$CCP   002106 G        2086#
L$CLEA  034650 G        2087    4808#
L$CO    002032 G        2042#
L$DEPO  002011 G        2024#
L$DESC  012430 G        2079    3148#
L$DESP  002076 G        2078#
L$DEVP  002060 G        2064#
L$DISP  002124 G        2049    2109#
L$DLY   002116 G        2094#
L$DTP   002040 G        2048#
L$DTYP  002034 G        2044#
L$DU    034740 G        2075    4841#
L$DUT   002072 G        2074#
L$DVTY  012414 G        2065    3137#
L$EF    002052 G        2059#
L$ENVI  002044 G        2052#
L$ETP   002102 G        2082#
L$EXP1  002046 G        2054#
L$EXP4  002064 G        2068#
L$EXP5  002066 G        2070#
L$HARD  046140 G        2031    6858    6859#
L$HIME  002120 G        2096#
L$HPCP  002016 G        2030#
L$HPTP  002022 G        2034#
L$HW    002130 G        2035    2122    2123#
L$ICP   002104 G        2084#
L$INIT  033746 G        2085    4610#
L$LADP  002026 G        2038#
L$LAST  046522 G        2039    6940#
L$LOAD  002100 G        2080#
L$LUN   002074 G        2076#
L$MREV  002050 G        2056#
L$NAME  002000 G        2013#
L$PRIO  002042 G        2050#
L$PROT  033740 G        2091    4594#
L$PRT   002112 G        2090#
L$REPP  002062 G        2066#
L$REV   002010 G        2022#
L$RPT   033732 G        2067    4574#
L$SPC   002056 G        2062#
L$SPCP  002020 G        2032#
L$SPTP  002024 G        2036#
L$STA   002030 G        2040#
L$TEST  002114 G        2092#
L$TIML  002014 G        2028#
L$UNIT  002012 G        2026#   4708
L10000  002150          2122    2153#
L10001  025670          3179#
L10002  025716          3192#
L10003  025746          3206#
L10004  026030          3228#
L10005  026106          3249#
L10006  026140          3263#
L10007  026176          3278#   3283
```

```
L10010  026350          3391#
L10011  033736          4584#
L10013  034644          4776    4782#
L10014  034646          4798#
L10015  034736          4825    4831#
L10016  034744          4846    4852#
L10017  034752          4868    4874#
L10020  046134          4969    5032    6683    6840#
L10021  044672          6525#
L10022  044702          6532#
L10023  046212          6858    6903#
L5060   014775          3163#   4682
MAINTB= 000400          2412#   6287
MCLR  = 040000          2410#   6209
MLTYP   007472          2778#   3457    4945*   4956    4970*   5295    5404*   5423*   5469*   5471*   5473*   5475*   5477*
                        5482*   5494    6253    6265    6268
MOBITE  010476          2836#   3969
MOBITS  010460          2829#   3957
MOCHK = 000010          2276#   4293    5454
MODE    007504          2794#   5542
MODES   003514          2684#   4259
MODLOC= 000003          2261#   6268
MODREM= 000004          2262#   6265
MODS    010456          2825#   3486    6333*   6372*   6427*
MODTYP  007470          2775#   3456    4944*   4955    5294    5399*   5402*   5407*   5410*   5413*   5416*   5419*   5432
                        5479    5540    5723    5737    5754    6115*   6171*   6285
MOMSGS  010476          2840#   3958
MOP   = 000043          2263#   2367#
MOSC  = 000056          2378#   3027
MO0     014550          2684    3163#
MO1     014560          2685    3163#
MO2     014571          2686    3163#
MO3     014601          2687    3163#
MO4     014610          2688    3163#
MO5     014625          2689    3163#
MO6     014632          2690    3163#
MSG     002734          2541    2549#
MSGLIM= 000017          2243#   4202    4922    5055    5095    5101    5226    5241    5511    5517
MSGTRN  015245          3163#   5044    5059    5085    5105
MSGTRU  015276          3163#   4074
MSGTYP  007410          2745#   4118    4928*   4940*   5265*   5344*   5368*   5370*   5372*   5374*   5376*   5378*   5381*
                        5538*
MSG0    002220          2428    2443    2453#
MSG0C   002150          2428#
MSG1    002221          2429    2444    2455#
MSG1C   002152          2429#
MSG2    002222          2430    2445    2457#
MSG2C   002154          2430#
MSG3    002223          2431    2446    2459#
MSG3C   002156          2431#
MSG4    002224          2432    2447    2461#
MSG4C   002160          2432#
MSG5    002324          2433    2448    2475#
MSG5C   002162          2433#   4929    4941    5266    5379
MSG6    002416          2434    2449    2489#
MSG6C   002164          2434#   5382
```

```
 CZCLKCO DMR,DMC-11 DATA COMM. LINK TEST        MACY11 30A(1052)  23-MAR-82  16:45  PAGE 186
CZCLKC.P11    19-MAR-82 18:32            CROSS REFERENCE TABLE -- USER SYMBOLS

MSG8     002646            2436    2451    2515#
MSG8C    002170            2436#
NEW      034302            4640    4705#   4709
NO     = 000036            2362#   3007    5426    5429    5460
NOBUF    007374            2737#   4819    5527*   5893    6695*
NOCLK    015106            3163#   4690    4905
NOD0     010664            2934#
NOD1     010670            2935#
NOD10    010746            2942#
NOD100   011564            3017#
NOD101   011570            3018#
NOD102   011604            3019#
NOD103   011610            3022#
NOD104   011624            3023#
NOD105   011630            3024#
NOD106   011644            3025#
NOD107   011650            3027#
NOD11    010750            2943#
NOD110   011664            3028#
NOD111   011670            3031#
NOD112   011674            3034#
NOD113   011710            3035#
NOD114   011714            3036#
NOD115   011732            3037#
NOD116   011736            3038#
NOD117   011752            3039#
NOD12    010762            2944#
NOD120   011756            3040#
NOD121   011772            3041#
NOD122   011776            3042#
NOD123   012012            3043#
NOD124   012016            3044#
NOD125   012032            3045#
NOD126   012036            3046#
NOD127   012052            3047#
NOD13    010766            2945#
NOD130   012056            3048#
NOD131   012076            3049#
NOD132   012102            3052#
NOD133   012106            3053#
NOD134   012112            3054#
NOD135   012116            3055#
NOD136   012122            3056#
NOD137   012126            3057#
NOD14    011002            2946#
NOD140   012132            3058#
NOD141   012134            3061#
NOD142   012140            3062#
NOD143   012144            3063#
NOD144   012160            3064#
NOD145   012164            3065#
NOD146   012200            3066#
NOD147   012204            3069#
NOD15    011006            2947#
NOD150   012210            3070#
NOD151   012214            3071#
```

```
NOD152  012220          3074#
NOD153  012224          3077#
NOD154  012246          3078#
NOD155  012252          3079#
NOD156  012266          3080#
NOD157  012272          3081#
NOD16   011022          2948#
NOD160  012314          3082#
NOD161  012320          3083#
NOD162  012342          3084#
NOD163  012346          3087#
NOD164  012352          3088#
NOD165  012356          3089#
NOD166  012362          3094#
NOD167  027266          3605#
NOD17   011026          2949#
NOD170  027272          3606#
NOD171  027276          3607#
NOD172  027300          3608#
NOD173  027314          3609#
NOD174  027316          3610#
NOD175  027332          3611#
NOD176  027334          3612#
NOD177  027346          3613#
NOD2    010674          2936#
NOD20   011032          2950#
NOD200  027350          3614#
NOD201  027364          3615#
NOD202  027370          3616#
NOD203  027374          3617#
NOD204  027410          3618#
NOD205  027412          3619#
NOD206  027426          3620#
NOD207  027430          3621#
NOD21   011044          2951#
NOD210  027446          3622#
NOD211  027452          3623#
NOD212  027456          3624#
NOD213  027460          3625#
NOD214  027462          3626#
NOD22   011050          2952#
NOD23   011062          2953#
NOD24   011066          2954#
NOD25   011070          2958#
NOD26   011074          2959#
NOD27   011110          2960#
NOD3    010676          2937#
NOD30   011114          2961#
NOD31   011132          2962#
NOD32   011136          2963#
NOD33   011154          2964#
NOD34   011160          2965#
NOD35   011176          2966#
NOD36   011202          2967#
NOD37   011220          2968#
NOD4    010712          2938#
```

```
NOD40   011224            2969#
NOD41   011250            2970#
NOD42   011254            2971#
NOD43   011260            2972#
NOD44   011276            2973#
NOD45   011302            2974#
NOD46   011314            2975#
NOD47   011320            2979#
NOD5    010714            2939#
NOD50   011324            2980#
NOD51   011342            2981#
NOD52   011344            2982#
NOD53   011364            2983#
NOD54   011366            2988#
NOD55   011372            2989#
NOD56   011410            2990#
NOD57   011414            2991#
NOD6    010730            2940#
NOD60   011434            2992#
NOD61   011440            2995#
NOD62   011444            2996#
NOD63   011450            2997#
NOD64   011454            2998#
NOD65   011460            2999#
NOD66   011464            3000#
NOD67   011470            3001#
NOD7    010732            2941#
NOD70   011474            3002#
NOD71   011500            3005#
NOD72   011504            3006#
NOD73   011510            3007#
NOD74   011522            3008#
NOD75   011526            3009#
NOD76   011542            3010#
NOD77   011546            3016#
NONE  = 000000            2258#
NOTNUF= 000050            2372#  2945  2948  2949  2959  2998  3000  3006  3062
NULEVT  016265            3163#  3806
NULL  = 000000            2332#
NUM   = 000014            2344#  3070
NXMM    025310            3163#  6735
N10$    010670            2934#
N100$   011320            2948   2951  2978#
N102$   011324            2979#
N104$   011344            2980   2981#
N110$   011366            2953   2987#
N111$   011372            2988#
N112$   011414            2989   2990#
N114$   011504            3005#
N115$   011500            2962   2964  2966  2968  2970  2973  2975  3004#  3010  3017  3019  3028  3078
                          3080   3082  3084  3089
N116$   011522            3007#
N117$   011546            3008   3009  3015#
N118$   011570            3016   3017#
N120$   011670            2990   2992  3030#
N121$   012134            3035   3037  3039  3041  3043  3045  3047  3054  3060#  3071
```

CZCLKCO DMR,DMC-11 DATA COMM. LINK TEST MACY11 30A(1052) 23-MAR-82 16:45 PAGE 189
CZCLKC.P11 19-MAR-82 18:32 CROSS REFERENCE TABLE -- USER SYMBOLS

```
N122$   012164          3063    3064#
N123$   012140          3061#
N125$   012362          2998    3000    3002    3006    3049    3062    3093#
N126$   012204          3064    3066    3068#
N130$   011610          3018    3021#
N131$   011630          3022    3023#
N132$   011650          3024    3026#
N140$   012220          3025    3073#
N141$   012224          3076#
N142$   012252          3077    3078#
N143$   012272          3079    3080#
N144$   012320          3081    3082#
N150$   012346          3023    3086#
N20$    011026          2947    2948#
N25$    011050          2950    2951#
N30$    011066          2949    2952    2953#   2958    2959    2960    2971    2974    2979    2982    2988    2991    2997
                        2999    3001    3027    3031    3052    3065    3069    3070    3074    3083    3087    3088
N40$    011006          2945    2946#
N42$    010676          2935    2936#
N43$    010714          2937    2938#
N44$    010732          2939    2940#
N45$    010750          2941    2942#
N46$    010766          2943    2944#
N50$    011440          2946    2994#
N51$    011444          2995#
N52$    011450          2996#
N60$    011674          3033#
N61$    011714          3034    3035#
N62$    011736          3036    3037#
N63$    011756          3038    3039#
N64$    011776          3040    3041#
N65$    012016          3042    3043#
N66$    012036          3044    3045#
N67$    012056          3046    3047#
N68$    012102          3048    3051#
N70$    012106          3052#   3057
N71$    012116          3053    3054#
N72$    012122          3055#
N73$    012132          3056    3057#
N80$    011070          2944    2957#
N81$    011074          2958#
N82$    011136          2961    2962#
N83$    011160          2963    2964#
N84$    011202          2965    2966#
N85$    011224          2967    2968#
N86$    011254          2969    2970#
N87$    011302          2972    2973#
OFSET   007424          2752#   3172    5841*   5858*
OPBFPT  002520          2508#   6162
OPBUF   002524          2450    2509#   5354    6076    6083    6091    6097    6098    6110    6112    6144    6145    6167
                        6169
OPCNT   002166          2435#   5352*   6095*   6099    6100
OPEND   002646          2510#   6078
OPRMM   014770          3163#   6085
OPRMSG= 000015          2345#   3052
OPTN    046352          6894    6905#
```

```
OPTYP   012412          3117#  3669   3690   3708   3726   4743*  6211   6214   6248   6260   6297   6698   6739
                        6770   6815
OTINT = 000002          2306#  6469   6530   6622   6665   6824
OUTHDL  045134          6472   6626   6664#  6835
OUTHEX  045752          6696   6702   6708   6757   6765   6771   6773#
OUTH1   045224          6667   6685#
OUTH2   045660          6687   6759#
OUTH3   045714          6761   6767#
OUTH4   045722          6768#
OUTH6   045266          6693   6698#
OUTVEC  012406          3115#  4740*  4741*  4766
O$APTS= 000000          1996#  2040
O$AU  = 000001          1996#  2009#  2072
O$BGNR= 000001          1996#  2009#  2066
O$BGNS= 000000          1996#  2032
O$DU  = 000001          1996#  2009#  2074
O$ERRT= 000000          1996#  2082
O$GNSW= 000000          1996#  2036
O$POIN= 000001          1996#  2009#  2098
O$SETU= 000000          1996#  2026   6938
PARAM   007476          2782#  3459   3491   4947*  4958   5297   5421*  5431*  5441*  5459*  5462*  5496   5583*
                        5714   5719   5766   5810   5947*  6075*  6137*
PAS   = 000002          2252#  5402   5432   5737   5754
PASC  = 000042          2366#  3088
PASMOD= 000030          2356#  2963
PASS1   002650          2522#  5942*
PASS2   002651          2523#  5943*
PASS3   002652          2524#  5944*
PASS4   002653          2525#  5945*
PCADD   007462          2772#  3221   3242   6555*  6603*  6664*
PCK     014732          3163#  4284
PCLKCT= 001600          2269#  4665
PCLKEN= 000111          2268#  4667
PCPM    015522          3163#  3222   3243
PEC     014742          3163#  4288
PLCK    040730          2796   5607#
PLCK2   040730          5608#
PLCK3   040744          5610#
PMS     014751          3163#  4292
PNCK    014730          3163#  4287
PNEC    014740          3163#  4291
PNMS    014747          3163#  4295
PNST    014717          3163#  4283
PNT   = 001000 G        2231#
PRI   = 002000 G        2232#
PRIOR   046324          6886   6905#
PRI00 = 000000 G        2220#  4772
PRI01 = 000040 G        2219#
PRI02 = 000100 G        2218#
PRI03 = 000140 G        2217#
PRI04 = 000200 G        2216#
PRI05 = 000240 G        2215#
PRI06 = 000300 G        2214#
PRI07 = 000340 G        2051   2213#  4813
PRNT  = 000055          2377#  2939   5017   5216
PROEM   025270          3163#  6738
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
PROTO = 000041          2365#
PROTOB= 000040          2278#   5457
PSCNT   007376          2738#   4820    5528*   5891*   5894
PST     014721          3163#   4280
PTREND  007332          2715#
PTRTAB  006562          2714#   4205    4210    4919    5053    5094    5225    5240    5251    5258    5509
P$ACT   003550          2702#   3573*   4375    4989*
P$BUFA  003544          2700#   3571*   4330    4987*
P$CNT   003552          2703#   4472*   4479    4541*   4548*   4553
P$EXIT  032760          4333    4368#   4392    4397
P$GDBD  003561          2707#   3557*   3576    3722*   4390*   4395*   4403    4454*   4497*   4975*   4992    5336*   5390*
                        5440*   5481*
P$NNUF  003560          2706#   3558*   3586    3642*   3707*   3792*   4976*   5001    5193*   5221*   5234*   5253*   5308*
                        5396*   5403*   5422*   5489*
P$NUM   003554          2704#   3713    3716    3724    3732    3734    3740    3742    4467*   4470*   4472    4473*   4475*
                        4479*   4481*   4487*   5301    5307    5328    5338    5340    5466
P$RADX  003556          2705#   4427*   4430*   4437*   4444    4477    4485
P$TREE  003546          2701#   3572*   4331    4988*
P$TRV   032642          3575    4329#   4991
P$TR5   032652          4332#   4341    4361    4366
QCOPY = 000013          2343#   3065    5323
QRX   = 000004          2307#   5563    5611    5637    5687    5709    5739    5746    5757    5981    6016    6148    6366
                        6369    6511
QTX   = 000010          2308#   5586    5637    5697    5740    5748    5756    5765    5994    5998    6011    6102    6114
                        6422    6424    6492
QUALFG  003204          2577#   3574*   4990*   5320*   5323*   5326    5426*   5429    5442*   5460    5463*
QUALVL  003206          2578#   5076*   5122*   5317*   5340*   5341*
RDI   = 000200          2407#   6562
RDO   = 000200          2408#   6666    6773
REC   = 000000          2250#
RECMOD= 000031          2357#   2965
REPLOG  030574          3664    3799#
REPORT  027046          3545#   4577    5217
RESFLG  007466          2774#   4619*   4643*   4911
RESTRT  034242          4629    4655    4668    4687    4696#
RHLPEN  003246          2596#   3654
RHLPTB  003230          2589#   3646
RHLP0   014024          3163#   3551
RHLP1   014063          2589    3163#
RHLP2   014106          2590    3163#
RHLP3   014141          2591    3163#
RHLP4   014172          2592    3163#
RHLP5   014224          2593    3163#
RHLP6   014263          2594    3163#
RHLP7   014322          2595    3163#
RI    = 000200          2399#   2833
RMDLOP= 000047          2371#   3083
RNOTNF= 000007          2388#   3614    3616    3621
RPASS   007500          2790#   3458    4946*   4957    4965    5222*   5296    5466*   5897    5899*
RPBASE  030336          3686    3703    3743    3764#
RPEXT = 000002          2383#   3596    3610    3660
RPHLP = 000001          2382#   3606    3608    3656
RPLOG = 000003          2384#   3612    3665
RPSWE = 000004          2385#   3617
RPSWF = 000005          2386#   3619
RPSWO = 000006          2387#   3623
```

```
RPT     030642            2903    3813#   3862    3882    3893    3909    3924    3940
RPTAA   030712            3628#
RPTABO  031226            2907    3895#
RPTDCK  031356            2902    3927#
RPTDDE  031302            2905    3912#
RPTDER  031066            2900    3864#
RPTDLE  031356            2904    3926#
RPTDSP  010624            2896#   3848
RPTDVI  031162            2901    3884#
RPTEOP  031226            2906    3896#
RPTIV   014374            3163#   3717
RPTMSB  031442            3861    3951#
RPTTXQ  031014            2896    2897    2898    2899    3850#
RPTO    030706            3804    3820    3824    3827#
RPT1    030674            3817    3823#
RQI   = 000040            2406#   6556
RSEL4   045766            6499    6509    6768*   6777#
RSEL6   045770            6500    6510    6769*   6772*   6778#
RTS   = 000040            2398#   2832
RUN   = 000004            2336#   2943    5019    5220
RUNSBM  025437            3163#   6709
RXBIT = 000004            2414#   6760
RXBUF   004562            2712#   5536    5982    5983    6002    6023    6029    6036
RXC   = 000006            2293#   3446    3489
RXIDM   025461            3163#   6743
RXMTOT  007370            2734#   5523*   5562    5635    5814
RXM1    025561            3163#   6026
RXM2    025604            3163#   6031
RXNC    025541            3163#   6018
RXONLY  040636            2794    5560#
RXON2   040636            5561#
RXPTR   007332            2717#   4921*   5520*   5521*   5537    5561    5610    5636    5722    5812
RXQ   = 000004            2292#   3442
R10$    027272            3605#
R11$    027300            3606    3607#
R12$    027316            3608    3609#
R125$   027462            3616    3625#
R13$    027334            3610    3611#
R14$    027350            3612    3613#
R15$    027370            3615#
R16$    027412            3617    3618#
R17$    027430            3619    3620#
R30$    027460            3614    3621    3622    3623    3624#
SCM     020047            3163#   3463
SCMD    020102            3163#   3471
SCML    020071            3163#   3467
SDVE    020036            3163#   3449
SDVI    020060            3163#   3454
SECRM   020135            3163#   6046
SEL0    012364            3100#   4723*   6209*   6218    6226    6252*   6255*   6439    6457    6556*   6562    6567    6610
                          6829
SEL2    012370            3103#   4726*   4727*   6227    6251*   6440    6458    6568    6609    6666    6669    6685    6711
                          6760    6773*   6818*   6830
SEL4    012374            3106#   4730*   4731*   6208*   6245*   6333    6372    6373*   6427    6428*   6763    6768
SEL6    012400            3109#   4734*   4735*   6207*   6247*   6250*   6264*   6284*   6287*   6292*   6294    6334*   6374*
                          6429*   6670    6689    6764    6769
```

```
SEOP    020113          3163#   3475
SETET = 000060          2380#   3048    5012    5393
SETEXP= 000010          2340#   2989    5037    5313    5388
SETTRN= 000011          2341#   2991    5316
SHF0    015361          3163#   4273
SHF1    015417          3163#   4303
SHMSG   014434          3163#   5286
SHOW  = 000002          2334#   2950    5035    5199    5235    5254
SHTAB   003504          2678#   5274    5280
SHTEND  003513          2681#   5277
SHTYP0  014470          2671    3163#
SHTYP1  014477          2671    3163#
SHTYP2  014504          2671    3163#
SHTYP3  014511          2671    3163#
SHTYP4  014516          2671    3163#
SHTYP5  014524          2671    3163#
SHTYP6  014531          2671    3163#
SHTYP7  014537          2671    3163#
SHTYTB  003464          2671#   5285
SHWOP   032340          3891    4257#   4959    5298
SIZE  = 000012          2342#   3063    5320    5326
SMSC    020124          3163#
SQD   = 040000          2400#   2834
SRXQ    020025          3163#   3441
STADD   007402          2740#   4007    5301*
START   034042          4624    4643#
STATB = 000001          2273#   3491    4281    5448
STATUS= 000016          2346#   3016
STXC    020014          3163#   3436
STXQ    020003          3163#   3431
SVCGBL= 000000          1996#   2013    2022    2024    2026    2028    2030    2032    2034    2036    2038    2040    2042
                        2044    2046    2048    2050    2052    2054    2056    2059    2062    2064    2066    2068    2070
                        2072    2074    2076    2078    2080    2082    2084    2086    2088    2090    2092    2094    2096
                        2109    2123    2124    3137    3148    3166    3183    3196    3210    3232    3253    3267    3366
                        4574    4594    4610    4794    4808    4841    4863    6522    6529    6859    6940#   6941
SVCINS= 000001          1996#   2014    2015    2016    2017    2018    2019    2020    2021    2023    2025    2027    2029
                        2031    2033    2035    2037    2039    2041    2043    2045    2047    2049    2051    2053    2055
                        2057    2058    2060    2061    2063    2065    2067    2069    2071    2073    2075    2077    2079
                        2081    2083    2085    2087    2089    2091    2093    2095    2097    2108    2110    2122    3138
                        3140    3149    3156    3168    3169    3170    3171    3172    3173    3174    3175    3176    3177
                        3180    3185    3186    3187    3188    3189    3190    3193    3198    3199    3200    3201    3202
                        3203    3204    3207    3212    3213    3214    3215    3216    3217    3218    3219    3221    3222
                        3223    3224    3225    3226    3229    3234    3235    3236    3237    3238    3239    3240    3242
                        3243    3244    3245    3246    3247    3250    3255    3256    3257    3258    3259    3260    3261
                        3264    3269    3270    3271    3272    3273    3274    3275    3276    3279    3282    3283    3392
                        3500    3501    3502    3503    3504    3508    3509    3510    3511    3512    3551    3552    3553
                        3554    3555    3562    3563    3564    3565    3566    3567    3568    3569    3579    3580    3581
                        3582    3583    3589    3590    3591    3592    3593    3648    3649    3650    3651    3652    3653
                        3716    3717    3718    3719    3720    3721    3767    3768    3769    3770    3771    3781    3782
                        3783    3784    3785    3786    3787    3788    3789    3806    3807    3808    3809    3810    3829
                        3830    3831    3832    3833    3839    3840    3841    3842    3843    3844    3845    3846    3847
                        3854    3855    3856    3857    3858    3859    3860    3868    3869    3870    3871    3872    3873
                        3875    3876    3877    3878    3879    3880    3881    3900    3901    3902    3903    3904    3905
                        3906    3907    3916    3917    3918    3919    3920    3921    3922    3923    3931    3932    3933
                        3934    3935    3936    3937    3938    3952    3953    3954    3955    3956    3972    3973    3974
                        3975    3976    4010    4011    4012    4013    4014    4015    4020    4021    4022    4023    4024
```

```
                    4025    4026    4029    4030    4031    4032    4033    4034    4074    4075    4076    4077    4078
                    4269    4270    4271    4272    4273    4274    4275    4276    4277    4299    4300    4301    4302
                    4303    4304    4305    4306    4307    4449    4450    4451    4452    4453    4492    4493    4494
                    4495    4496    4585    4617    4621    4622    4624    4626    4627    4629    4631    4632    4634
                    4637    4638    4640    4648    4649    4650    4652    4658    4659    4660    4662    4671    4673
                    4678    4679    4680    4681    4682    4683    4684    4685    4690    4691    4692    4693    4694
                    4712    4713    4714    4716    4747    4748    4749    4750    4751    4752    4757    4758    4759
                    4760    4761    4762    4764    4765    4766    4767    4768    4769    4772    4773    4775    4776
                    4783    4799    4813    4814    4824    4825    4832    4845    4846    4853    4867    4868    4875
                    4905    4906    4907    4908    4909    4950    4951    4952    4953    4954    4962    4964    4968
                    4969    4978    4979    4980    4981    4982    4983    4984    4985    4995    4996    4997    4998
                    4999    5004    5005    5006    5007    5008    5031    5032    5043    5044    5045    5046    5047
                    5048    5058    5059    5060    5061    5062    5063    5084    5085    5086    5087    5088    5089
                    5104    5105    5106    5107    5108    5109    5204    5205    5206    5207    5208    5209    5284
                    5285    5286    5287    5288    5289    5290    5331    5332    5333    5334    5335    5361    5362
                    5363    5364    5365    5435    5436    5437    5438    5439    5484    5485    5486    5487    5488
                    5501    5502    5503    5504    5505    5834    5835    5836    5837    5852    5853    5854    5855
                    5865    5866    5867    5868    5933    5934    5935    5936    5937    5938    5939    5940    5962
                    5963    5964    5965    5966    5971    5972    5973    5974    6044    6045    6046    6047    6048
                    6049    6050    6081    6082    6083    6084    6085    6086    6087    6088    6139    6140    6141
                    6142    6143    6162    6163    6164    6165    6166    6221    6231    6232    6233    6234    6444
                    6445    6446    6447    6452    6463    6464    6465    6466    6484    6485    6486    6487    6503
                    6504    6505    6506    6526    6533    6561    6572    6573    6574    6575    6614    6615    6616
                    6617    6621    6674    6675    6676    6677    6682    6683    6753    6754    6755    6756    6823
                    6841    6858    6865    6866    6867    6875    6876    6877    6878    6880    6881    6882    6883
                    6885    6886    6887    6888    6889    6893    6894    6895    6896    6897    6902    6937    6938
                    6939
SVCSUB= 000001       1996#
SVCTAG= 000001       1996#   2153    3179    3192    3206    3228    3249    3263    3278    3391    3570    4584    4686
                     4782    4798    4831    4852    4874    4986    5941    6089    6525    6532    6840    6903
SVCTST= 000001       1996#   4890
S$LSYM= 010000       1996#   2154#   3180#   3193#   3207#   3229#   3250#   3264#   3279#   3392#   3563    3570    3571#
                     4585#   4679    4686    4687#   4783#   4799#   4832#   4853#   4875#   4979    4986    4987#   5934
                     5941    5942#   6082    6089    6090#   6526#   6533#   6841#   6904#
S1          034030   4634    4636#
S2          034104   4652    4657#
S3          034154   4662    4670#
S4          034222   4673    4689#
TABEX       015167   3163#   5058    5104
TAL     =   000005   2255#   5419    6171
TALCK       042734   2799    6074#   6111    6113
TALMOD=     000035   2361#   2974
TCURAD      007360   2729#   4930*   4933    5069    5074*   5261*
TEMP        007426   2753#   3432*   3437*   3442*   3446*   3450*   3455*   3464*   3468*   3472*   3476*   3489    3515
                     3516*   3517*   3518    4018*   4021    4080*   4081*   4082    4120*   4121*   4127    4259*   4272
                     4280*   4283*   4302    4818*   5282*   5284    5345*   5346*   5350    5353    5728*   5729*   5733
                     6035*   6041*   6045
TEMP1       007430   2754#   3431*   3436*   3441*   3449*   3454*   3463*   3467*   3471*   3475*   3508    3775*   3783
                     4266*   4270    4284*   4287*   4301
TEMP2       007432   2755#   3456*   3457*   3521    3779*   3781    4123*   4125*   4129    4267*   4269    4288*   4291*
                     4300    4820*   5690*   5700*   5711*   5717    5731*   5732    5750*   5818*   5894*   5983*   5990*
                     6004*   6008*   6020*   6098*   6107*   6145*   6156*   6225*   6438*   6456*   6479*   6498*   6566*
                     6608*   6668*   6710*   6828*
TEMP3       007434   2756#   3198    3214    3235    3256    3271    3458*   3522    3774*   3784    4262*   4265*   4271
                     4821*   5692*   5702*   5712*   5718    5729    5730    5731    5751*   5820*   5830    5895*   5935
                     5942    5943    5944    5945    5985*   5992*   6002*   6009*   6021*   6099*   6108*   6147*   6157*
```

```
                    6226* 6439* 6457* 6480* 6499* 6567* 6610* 6669* 6711* 6829*
TEMP4   007436      2757# 3185  3213  3234  3255  3270  3459* 3486* 3523  4819*  5827* 5840* 5845*
                    5846  5861  5893* 6003* 6227* 6440* 6458* 6481* 6500* 6568*  6609* 6670* 6689*
                    6692  6703  6715  6718  6721  6724  6727  6730  6733  6736   6741  6744  6747
                    6830*
TEMP5   007440      2758# 4292* 4295* 4299
TIMERS  007546      2816# 3384  3388* 6432* 6436
TIMER1  007542      2814# 3378  3380* 4899* 4900  6206* 6222  6557* 6558  6604*  6605  6821* 6826
TIMER2  007544      2815# 3381  3383*
TIMMIN  007534      2810# 3375* 3520  4696*
TIMOM   025413      3163# 6720
TIMSEC  007536      2811# 3372* 3373  3376* 3519  4697*
TIMTCK  007540      2812# 3369* 3371* 3386  3517  4698*
TM    = 001000      2401# 2835
TOINOT  044250      6433# 6467  6470  6474
TOIN1   044332      6437  6451#
TOIN2   044416      6454  6469#
TOORIO  045012      6244  6263  6277  6283  6332  6371  6426  6603# 6618  6819
TOOR1   045076      6606  6620#
TOOR2   045114      6624  6627#
TOOR3   045024      6605# 6628
TOTCC   007422      2751# 4067* 4068  4079* 4081  4083* 4916* 5039* 5040  5073  5080* 5081 5121
TRA   = 000001      2251# 5416
TRAMOD= 000034      2360# 2972
TRVACT  032762      4362  4373# 4389  4394  4399  4402  4422  4488  4511  4532  4556
TRVALN  033554      4351  4515#
TRVALP  033510      4350  4501#
TRVBIF  033066      4347  4402#
TRVBR   033056      4346  4399#
TRVBRC  033002      4360  4380# 4400  4405  4424  4498  4513  4534  4560
TRVDEC  033162      4353  4427#
TRVERR  033020      4344  4389#
TRVEXI  033040      4345  4394#
TRVNMA  033202      4428  4431#
TRVNOB  033012      4385# 4406  4423  4489  4512  4533
TRVNUM  033174      4349  4430#
TRVOCT  033174      4352  4429#
TRVSPA  033110      4348  4408#
TRVSTR  033642      4354  4538#
TSEL4   045762      6480  6490  6763* 6775#
TSEL6   045764      6481  6491  6764* 6776#
TTL   = 000001      2259# 4970  6253
TTLLOP= 000044      2368# 3077
TTOTCC  007356      2728# 4229  4917* 5039  5050  5073* 5257*
TXBUF   003562      2711# 4197  4930  5260  6003
TXC   = 000002      2291# 3437
TXMTOT  007354      2727# 4217  4936* 5052* 5054  5075* 5252  5256* 5498  5585  5608  5633
TXNC    025521      3163# 6001
TXONLY  040670      2795  5583#
TXON2   040676      5584#
TXPTR   007334      2718# 4210* 4212* 4213  4222* 4224  4920* 4934  5053* 5067* 5068  5072* 5258*
                    5259  5510* 5584  5609  5634
TXQ   = 000000      2290# 3432
T$ARGC= 000001      2014# 2015# 2016# 2017# 2018# 2019# 3168# 3177  3185# 3190  3198# 3204 3212#
                    3219  3221# 3226  3234# 3240  3242# 3247  3255# 3261  3269# 3276  3500# 3504
                    3508# 3512  3551# 3555  3579# 3583  3589# 3593  3648# 3653  3716# 3721 3767#
```

```
                3771    3781#   3789    3806#   3810    3829#   3833    3839#   3847    3854#   3860    3868#   3873
                3875#   3881    3900#   3907    3916#   3923    3931#   3938    3952#   3956    3972#   3976    4010#
                4015    4020#   4026    4029#   4034    4074#   4078    4269#   4277    4299#   4307    4449#   4453
                4492#   4496    4690#   4694    4905#   4909    4950#   4954    4995#   4999    5004#   5008    5043#
                5048    5058#   5063    5084#   5089    5104#   5109    5204#   5209    5284#   5290    5331#   5335
                5361#   5365    5435#   5439    5484#   5488    5501#   5505    5962#   5966    6044#   6050    6139#
                6143    6162#   6166
T$CODE= 005032   3565#   4681#   4981#   5936#   6084#   6865#   6875#   6880#   6885#   6893#
T$ERRN= 000023   1996#   5835#   5853#   5866#   5972#   6232#   6445#   6464#   6485#   6504#   6573#   6615#   6675#
                6754#
T$EXCP= 000000   3565#   3570    4681#   4686    4981#   4986    5936#   5941    6084#   6089    6875#   6879    6880#
                6884    6885#   6890    6893#   6898
T$FLAG= 000040   3282#   3284    4775#   4824#   4845#   4847    4867#   4867    4968#   5031#   6682#
T$GMAN= 000000   1996#   3562#   3565    3571#   4678#   4687#   4978#   4981    4987#   5933#   5942#   6081#   6084
                6090#
T$HILI= 000007   3565#   3569    4681#   4685    4981#   4985    5936#   5940    6084#   6088    6875#   6878    6880#
                6883    6885#   6889    6893#   6897
T$LAST= 000001   1996#   6938#
T$LOLI= 000000   3565#   3568    4681#   4684    4981#   4984    5936#   5939    6084#   6087    6875#   6877    6880#
                6882    6885#   6888    6893#   6896
T$LSYM= 010000   1996#   2154    3180    3193    3207    3229    3250    3264    3279    3392    4585    4783    4799
                4832    4853    4875    6526    6533    6841    6904
T$LTNO= 000001   6941#
T$NEST= 177777   1996#   1998#   2122#   2153#   3166#   3179#   3183#   3192#   3196#   3206#   3210#   3228#   3232#
                3249#   3253#   3263#   3267#   3278#   3366#   3391#   4574#   4584#   4594#   4601#   4610#   4782#
                4794#   4798#   4808#   4831#   4841#   4852#   4863#   4874#   4891#   6522#   6525#   6529#   6532#
                6840#   6858#   6902#   6942#
T$NS0 = 000000   1998#   6942
T$NS1 = 000004   2122#   2153    3166#   3179    3183#   3192    3196#   3206    3210#   3228    3232#   3249    3253#
                3263    3267#   3278    3366#   3391    4574#   4584    4594#   4601    4610#   4782    4794#   4798
                4808#   4831    4841#   4852    4863#   4874    4891#   6840    6858#   6902
T$NS2 = 000010   6522#   6525    6529#   6532
T$PTNU= 000000   1996#
T$SAVL= 177777   1996#
T$SEGL= 177777   1996#
T$SUBN= 000000   1996#   4890#
T$TAGL= 177777   1996#
T$TAGN= 010024   1996#   2122#   3166#   3183#   3196#   3210#   3232#   3253#   3267#   3366#   4574#   4594#   4610#
                4794#   4808#   4841#   4863#   4891#   6522#   6529#   6858#
T$TEMP= 000000   2110#   2111#   2153#   3179#   3192#   3206#   3228#   3249#   3263#   3278#   3282#   3283    3391#
                3565#   4584#   4601#   4681#   4775#   4776    4782#   4798#   4824#   4825    4831#   4845#   4846
                4852#   4867#   4868    4874#   4968#   4969    4981#   5031#   5032    5936#   6084#   6525#   6532#
                6682#   6683    6840#   6865#   6875#   6880#   6885#   6893#   6902#   6942#
T$TEST= 000001   1996#   4890#   6941
T$TSTM= 177777   1996#   3176    3180    3189    3193    3203    3207    3218    3225    3229    3239    3246    3250
                3260    3264    3275    3279    3503    3511    3554    3562    3582    3592    3652    3720    3770
                3788    3809    3832    3846    3859    3872    3880    3906    3922    3937    3955    3975    4014
                4025    4033    4077    4276    4306    4452    4495    4585    4617    4622    4627    4632    4638
                4649    4659    4671    4678    4693    4713    4751    4761    4768    4773    4775    4783    4799
                4814    4824    4832    4853    4875    4908    4953    4962    4968    4978    4998    5007    5031
                5047    5062    5088    5108    5208    5289    5334    5364    5438    5487    5504    5834    5852
                5865    5933    5965    5971    6049    6081    6142    6165    6221    6231    6444    6452    6463
                6484    6503    6561    6572    6614    6621    6674    6682    6753    6823    6841
T$TSTS= 000001   1996#   4891#
T$$AU = 010017   4863#   4867    4874
T$$AUT= 010014   4794#   4798
```

```
T$$CLE= 010015          4808#   4824    4831
T$$DU = 010016          4841#   4845    4852
T$$HAR= 010023          6858#   6903
T$$HW = 010000          2122#   2153
T$$INI= 010013          4610#   4775    4782
T$$MSG= 010007          3166#   3179    3183#   3192    3196#   3206    3210#   3228    3232#   3249    3253#   3263    3267#
                        3278    3282
T$$PRO= 010012          4594#
T$$RPT= 010011          4574#   4584
T$$SRV= 010022          3366#   3391    6522#   6525    6529#   6532
T$$TES= 010020          4891#   4968    5031    6682    6840
T1      034754 G        2110    4890#
UAM   = 000200 G        2229#
UNKM    020273          2879    3163#   6035
UPTABL  041176          5714#
UPTA1   041264          5720    5727#
UPTA3   041262          5724    5726#
UPTA4   041222          5715    5719#
UPTEX   041334          5726    5736#
VECTOR  046271          6881    6905#
X$    = 000215          1999#   2934#   2935#   2936#   2937#   2938#   2939#   2940#   2941#   2942#   2943#   2944#   2945#
                        2946#   2947#   2948#   2949#   2950#   2951#   2952#   2953#   2954#   2958#   2959#   2960#   2961#
                        2962#   2963#   2964#   2965#   2966#   2967#   2968#   2969#   2970#   2971#   2972#   2973#   2974#
                        2975#   2979#   2980#   2981#   2982#   2983#   2988#   2989#   2990#   2991#   2992#   2995#   2996#
                        2997#   2998#   2999#   3000#   3001#   3002#   3005#   3006#   3007#   3008#   3009#   3010#   3016#
                        3017#   3018#   3019#   3022#   3023#   3024#   3025#   3027#   3028#   3031#   3034#   3035#   3036#
                        3037#   3038#   3039#   3040#   3041#   3042#   3043#   3044#   3045#   3046#   3047#   3048#   3049#
                        3052#   3053#   3054#   3055#   3056#   3057#   3058#   3061#   3062#   3063#   3064#   3065#   3066#
                        3069#   3070#   3071#   3074#   3077#   3078#   3079#   3080#   3081#   3082#   3083#   3084#   3087#
                        3088#   3089#   3094#   3605#   3606#   3607#   3608#   3609#   3610#   3611#   3612#   3613#   3614#
                        3615#   3616#   3617#   3618#   3619#   3620#   3621#   3622#   3623#   3624#   3625#   3626#
X$ALWA= 000000          1996#
X$FALS= 000040          1996#
X$OFFS= 000400          1996#
X$TRUE= 000020          1996#
$PATCH  046436          6932#
.     = 046522          1996#   2503#   2509#   2541    2548    2574#   2682#   2711#   2712#   2713#   2714#   2820#   2821#
                        2937#   2941#   2945#   2952#   2959#   2961#   2967#   2969#   2980#   2982#   2989#   2991#   3007#
                        3009#   3016#   3018#   3022#   3024#   3034#   3036#   3038#   3040#   3042#   3048#   3063#   3065#
                        3081#   3140#   3156#   3163#   3283    3608#   3610#   3614#   3619#   3621#   3796#   4776    4825
                        4846    4868    4969    5032    6683    6906#   6933#
```

CROSS REFERENCE TABLE -- MACRO NAMES

```
BCOMPL      1#  1996#  4623  4628  4639  4963
BERROR      1#  1996#
BGNAU       1#  1996#  4862
BGNAUT      1#  1996#  4793
BGNCLN      1#  1996#  4807
BGNDU       1#  1996#  4840
BGNHRD      1#  1996#  6857
BGNHW       1#  1996#  2121
BGNINI      1#  1996#  4609
BGNMOD      1#  1996#  1997
BGNMSG      1#  1996#  3165  3182  3195  3209  3231  3252  3266
BGNPRO      1#  1996#  4593
BGNPTA      1#  1996#
BGNRPT      1#  1996#  4573
BGNSEG      1#  1996#
BGNSET      1#  1996#
BGNSFT      1#  1996#
BGNSRV      1#  1996#  3365  6521  6528
BGNSUB      1#  1996#
BGNSW       1#  1996#
BGNTST      1#  1996#  4889
BNCOMP      1#  1996#  4633  4651  4661  4672  4715
BNERRO      1#  1996#
BREAK       1#  1996#  6220  6451  6560  6620  6822
BRESET      1#  1996#
CKLOOP      1#  1996#
CLI      2000#  2933   2934  2935  2936  2937  2938  2939  2940  2941  2942  2943  2944  2945  2946
         2947   2948   2949  2950  2951  2952  2953  2957  2958  2959  2960  2961  2962  2963  2964
         2965   2966   2967  2968  2969  2970  2971  2972  2973  2974  2978  2979  2980  2981  2982
         2987   2988   2989  2990  2991  2994  2995  2996  2997  2998  2999  3000  3001  3004  3005
         3006   3007   3008  3009  3015  3016  3017  3018  3021  3022  3023  3024  3026  3027  3030
         3033   3034   3035  3036  3037  3038  3039  3040  3041  3042  3043  3044  3045  3046  3047
         3048   3051   3052  3053  3054  3055  3056  3057  3060  3061  3062  3063  3064  3065  3068
         3069   3070   3073  3076  3077  3078  3079  3080  3081  3082  3083  3086  3087  3088  3093
         3604   3605   3606  3607  3608  3609  3610  3611  3612  3613  3614  3615  3616  3617  3618
         3619   3620   3621  3622  3623  3624  3625
CLOCK       1#  1996#  4647  4657
CLOSE       1#  1996#
CLRVEC      1#  1996#
COMMEN      1#  1996#
DELAY       1#  1996#
DESCRI      1#  1996#  3147
DEVTYP      1#  1996#  3136
DISPAT      1#  1996#  2107
DISPLA      1#  1996#
DOCLN       1#  1996#  4616
DODU        1#  1996#
DORPT       1#  1996#
ENDAU       1#  1996#  4873
ENDAUT      1#  1996#  4797
ENDCLN      1#  1996#  4830
ENDCOM      1#  1996#
ENDDU       1#  1996#  4851
ENDHRD      1#  1996#  6901
ENDHW       1#  1996#  2152
ENDINI      1#  1996#  4781
```

CROSS REFERENCE TABLE -- MACRO NAMES

ENDMOD	1#	1996#	6941										
ENDMSG	1#	1996#	3178	3191	3205	3227	3248	3262	3277				
ENDPRO	1#	1996#	4600										
ENDPTA	1#	1996#											
ENDRPT	1#	1996#	4583										
ENDSEG	1#	1996#											
ENDSET	1#	1996#											
ENDSFT	1#	1996#											
ENDSRV	1#	1996#	3390	6524	6531								
ENDSUB	1#	1996#											
ENDSW	1#	1996#											
ENDTST	1#	1996#	6839										
EQUALS	1#	1996#	2169										
ERRDF	1#	1996#											
ERRHRD	1#	1996#	5970										
ERROR	1#	1996#											
ERRSF	1#	1996#											
ERRSOF	1#	1996#	5833	5851	5864	6230	6443	6462	6483	6502	6571	6613	6673	6752
ERRTBL	1#	1996#											
ESCAPE	1#	1996#	6681										
EXIT	1#	1996#	3281	4774	4823	4844	4866	4967	5030				
FEQUAL	1#	1996#											
GETBYT	1#	1996#											
GETPRI	1#	1996#											
GETWOR	1#	1996#											
GMANIA	1#	1996#											
GMANID	1#	1996#	3561	4677	4977	5932	6080						
GMANIL	1#	1996#											
GPHARD	1#	1996#	4711										
GPRMA	1#	1996#	6874	6879									
GPRMD	1#	1996#	3562#	3565	4678#	4681	4978#	4981	5933#	5936	6081#	6084	6884	6892
GPRML	1#	1996#	6864										
HEADER	1#	1996#	2012										
INLOOP	1#	1996#											
IOSETU	1#	1996#											
IOSTAR	1#	1996#											
KT11	1#	1996#											
LASTAD	1#	1996#	6936										
MANUAL	1#	1996#	4961										
MEMORY	1#	1996#											
MSBYTE	1#	1996#	2013#	2019	2020	2021							
MSCHEC	1#	1996#	3282#	4775#	4824#	4845#	4867#	4968#	5031#				
MSCNTO	1#	1996#	3565#	4681#	4981#	5936#	6084#	6865#	6875#	6880#	6885#	6893#		
MSCOUN	1#	1996#	3168#	3185#	3198#	3212#	3221#	3234#	3242#	3255#	3269#	3500#	3508#	3551#	3579#
	3589#	3648#	3716#	3767#	3781#	3806#	3829#	3839#	3854#	3868#	3875#	3900#	3916#	3931#	3952#
	3972#	4010#	4020#	4029#	4074#	4269#	4299#	4449#	4492#	4690#	4905#	4950#	4995#	5004#	5043#
	5058#	5084#	5104#	5204#	5284#	5331#	5361#	5435#	5484#	5501#	5962#	6044#	6139#	6162#	
MSDATA	1#	1996#	2013#	2022	2024	2026	2028	2030	2032	2034	2036	2038	2040	2042	2044
	2046	2048	2050	2052#	2054	2056	2059	2062	2064	2066	2068	2070	2072	2074	2076
	2078	2080	2082	2084	2086	2088	2090	2092	2094	2096	3137#	3148#			
MSDECR	1#	1996#	2153#	3179#	3192#	3206#	3228#	3249#	3263#	3278#	3391#	4584#	4601#	4782#	4798#
	4831#	4852#	4874#	6525#	6532#	6840#	6902#	6942#							
MSDEFA	1#	1996#	3565#	4681#	4981#	5936#	6084#	6865#	6875#	6880#	6885#	6893#			
MSENDE	1#	1996#	2153#	3179#	3192#	3206#	3228#	3249#	3263#	3278#	3391#	4584#	4782#	4798#	4831#
	4852#	4874#	6525#	6532#	6840#	6902#	6942#								
MSERRI	1#	1996#	5834#	5852#	5865#	5971#	6231#	6444#	6463#	6484#	6503#	6572#	6614#	6674#	6753#

M$ESCA 1# 1996# 6682# 6683
M$ESCS 1# 1996# 6682#
M$EXCP 1# 1996# 3565# 4681# 4981# 5936# 6084# 6875# 6880# 6885# 6893#
M$EXIT 1# 1996# 3282# 4775# 4776 4824# 4825 4845# 4867# 4968# 4969 5031# 5032
M$EXSE 1# 1996# 3282# 4775# 4824# 4845# 4867# 4968# 5031#
M$EXTJ 1# 1996# 3282# 3283 4775# 4824# 4845# 4846 4867# 4868 4968# 5031#
M$GEN 1# 1996# 2013# 2022# 2024# 2026# 2028# 2030# 2032# 2034# 2036# 2038# 2040# 2042# 2044#
2046# 2048# 2050# 2052# 2054# 2056# 2059# 2062# 2064# 2066# 2068# 2070# 2072# 2074# 2076#
2078# 2080# 2082# 2084# 2086# 2088# 2090# 2092# 2094# 2096# 2109# 2123# 2124# 2153# 3137#
3148# 3166# 3179# 3183# 3192# 3196# 3206# 3210# 3228# 3232# 3249# 3253# 3263# 3267# 3278#
3366# 3391# 3570# 4574# 4584# 4594# 4610# 4686# 4782# 4794# 4798# 4808# 4831# 4841# 4852#
4863# 4874# 4890# 4986# 5941# 6089# 6522# 6525# 6529# 6532# 6840# 6859# 6903# 6940#
M$GENB 1# 1996# 3562# 3563 4678# 4679 4978# 4979 5933# 5934 6081# 6082
M$GETS 1# 1996# 2153# 3179# 3192# 3206# 3228# 3249# 3263# 3278# 3391# 4584# 4601# 4782# 4798#
4831# 4852# 4874# 6525# 6532# 6840# 6902# 6942#
M$GETT 1# 1996# 3282# 4775# 4824# 4845# 4867# 4968# 5031# 6682#
M$GNGB 1# 1996# 1998# 2013# 2022# 2024# 2026# 2028# 2030# 2032# 2034# 2036# 2038# 2040# 2042#
2044# 2046# 2048# 2050# 2052# 2054# 2056# 2059# 2062# 2064# 2066# 2068# 2070# 2072# 2074#
2076# 2078# 2080# 2082# 2084# 2086# 2088# 2090# 2092# 2094# 2096# 2108# 2109 2122# 2123
2124 3137# 3148# 3166# 3183# 3196# 3210# 3232# 3253# 3267# 3366# 4574# 4594# 4610# 4794#
4808# 4841# 4863# 6522# 6529# 6858# 6859 6937# 6940
M$GNIN 1# 1996# 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# 2049 2050# 2051 2052# 2053 2054# 2055
2056# 2057 2058 2059# 2060 2061# 2062# 2063 2064# 2065 2066# 2067 2068# 2069 2070#
2071 2072# 2073 2074# 2075 2076# 2077 2078# 2079 2080# 2081 2082# 2083 2084# 2085
2086# 2087 2088# 2089 2090# 2091 2092# 2093 2094# 2095 2096# 2097 2108# 2110# 2122#
3137# 3138 3140 3148# 3149 3156 3168# 3169 3170# 3171 3172# 3173# 3174# 3175 3176#
3177 3180# 3185# 3186# 3187# 3188 3189# 3190 3193# 3198# 3199# 3200# 3201# 3202 3203#
3204 3207# 3212# 3213# 3214# 3215# 3216# 3217 3218# 3219 3221# 3222# 3223# 3224 3225#
3226 3229# 3234# 3235# 3236# 3237# 3238 3239# 3240 3242# 3243# 3244# 3245 3246# 3247
3250# 3255# 3256# 3257# 3258# 3259 3260# 3261 3264# 3269# 3270# 3271# 3272# 3273# 3274
3275# 3276 3279# 3282# 3283# 3391# 3392 3500# 3501# 3502 3503# 3504 3508# 3509# 3510
3511# 3512 3551# 3552# 3553 3554# 3555 3562# 3563# 3564# 3565# 3566 3567 3568 3569
3579# 3580# 3581 3582# 3583 3589# 3590# 3591 3592# 3593 3648# 3649# 3650# 3651 3652#
3653 3716# 3717# 3718# 3719 3720# 3721 3767# 3768# 3769 3770# 3771 3781# 3782# 3783
3784# 3785# 3786# 3787 3788# 3789 3806# 3807# 3808 3809# 3810 3829# 3830# 3831 3832#
3833 3839# 3840# 3841# 3842# 3843# 3844# 3845 3846# 3847 3854# 3855# 3856# 3857# 3858
3859# 3860 3868# 3869# 3870# 3871 3872# 3873 3875# 3876# 3877# 3878# 3879 3880# 3881
3900# 3901# 3902# 3903# 3904# 3905 3906# 3907 3916# 3917# 3918# 3919# 3920# 3921 3922#
3923 3931# 3932# 3933# 3934# 3935# 3936 3937# 3938 3952# 3953# 3954 3955# 3956 3972#
3973# 3974 3975# 3976 4010# 4011# 4012# 4013 4014# 4015 4020# 4021 4022# 4023# 4024
4025# 4026 4029# 4030# 4031# 4032 4033# 4034 4074# 4075# 4076 4077# 4078 4269# 4270#
4271# 4272# 4273# 4274# 4275 4276# 4277 4299# 4300# 4301# 4302# 4303# 4304# 4305 4306#
4307 4449# 4450# 4451 4452# 4453 4492# 4493# 4494 4495# 4496 4585# 4617# 4621# 4622#
4624# 4626# 4627# 4629# 4631# 4632# 4634# 4637# 4638# 4640# 4648# 4649# 4650# 4652# 4658#
4659# 4660# 4662# 4671# 4673# 4678# 4679# 4680# 4681# 4682 4683 4684 4685 4690# 4691#
4692 4693# 4694 4712# 4713# 4714# 4716# 4747# 4748# 4749# 4750# 4751# 4752 4757# 4758#
4759# 4760# 4761# 4762 4764# 4765# 4766# 4767# 4768# 4769 4772# 4773# 4775# 4776# 4783#
4799# 4813# 4814# 4824# 4825# 4832# 4845# 4846# 4853# 4867# 4868# 4875# 4905# 4906# 4907
4908# 4909 4950# 4951# 4952 4953# 4954 4962# 4964# 4968# 4969# 4978# 4979# 4980# 4981#
4982 4983 4984 4985 4995# 4996# 4997 4998# 4999 5004# 5005# 5006 5007# 5008 5031#
5032# 5043# 5044# 5045# 5046 5047# 5048 5058# 5059# 5060# 5061 5062# 5063 5084# 5085#
5086# 5087 5088# 5089 5104# 5105# 5106# 5107 5108# 5109 5204# 5205# 5206# 5207 5208#
5209 5284# 5285# 5286# 5287# 5288 5289# 5290 5331# 5332# 5333 5334# 5335 5361# 5362#
5363 5364# 5365 5435# 5436# 5437 5438# 5439 5484# 5485# 5486 5487# 5488 5501# 5502#

```
            5503    5504#   5505    5834#   5835#   5836#   5837#   5852#   5853#   5854#   5855#   5865#   5866#   5867#   5868#
            5933#   5934#   5935#   5936#   5937    5938    5939    5940    5962#   5963#   5964    5965#   5966    5971#   5972#
            5973#   5974#   6044#   6045#   6046#   6047#   6048    6049#   6050    6081#   6082#   6083#   6084#   6085    6086
            6087    6088    6139#   6140#   6141    6142#   6143    6162#   6163#   6164    6165#   6166    6221#   6231#   6232#
            6233#   6234#   6444#   6445#   6446#   6447#   6452#   6463#   6464#   6465#   6466#   6484#   6485#   6486#   6487#
            6503#   6504#   6505#   6506#   6525#   6526    6532#   6533    6561#   6572#   6573#   6574#   6575#   6614#   6615#
            6616#   6617#   6621#   6674#   6675#   6676#   6677#   6682#   6683#   6753#   6754#   6755#   6756#   6823#   6841#
            6858#   6865#   6866    6867    6875#   6876    6877    6878    6880#   6881    6882    6883    6885#   6886    6887
            6888    6889    6893#   6894    6895    6896    6897    6902#   6937#   6938#   6939#
M$GNLS        1#    1996#   3562#   3570    4678#   4686    4978#   4986    5933#   5941    6081#   6089
M$GNSU        1#    1996#
M$GNTA        1#    1996#   2153#   3179#   3192#   3206#   3228#   3249#   3263#   3278#   3391#   4584#   4782#   4798#   4831#
            4852#   4874#   6525#   6532#   6840#   6902#   6903
M$GNTE        1#    1996#   4890#
M$HAPT        1#    1996#   2013#
M$HNAP        1#    1996#   2013#   2052
M$INCR        1#    1996#   1998#   2122#   3166#   3176#   3180#   3183#   3189#   3193#   3196#   3203#   3207#   3210#   3218#
            3225#   3229#   3232#   3239#   3246#   3250#   3253#   3260#   3264#   3267#   3275#   3279#   3366#   3503#   3511#
            3554#   3562#   3571    3582#   3592#   3652#   3720#   3770#   3788#   3809#   3832#   3846#   3859#   3872#   3880#
            3906#   3922#   3937#   3955#   3975#   4014#   4025#   4033#   4077#   4276#   4306#   4452#   4495#   4574#   4585#
            4594#   4610#   4617#   4622#   4627#   4632#   4638#   4649#   4659#   4671#   4678#   4687    4693#   4713#   4751#
            4761#   4768#   4773#   4775#   4783#   4794#   4799#   4808#   4814#   4824#   4832#   4841#   4853#   4863#   4875#
            4890#   4891#   4908#   4953#   4962#   4968#   4978#   4987    4998#   5007#   5031#   5047#   5062#   5088#   5108#
            5208#   5289#   5334#   5364#   5438#   5487#   5504#   5834#   5852#   5865#   5933#   5942    5965#   5971#   6049#
            6081#   6090    6142#   6165#   6221#   6231#   6444#   6452#   6463#   6484#   6503#   6522#   6529#   6561#   6572#
            6614#   6621#   6674#   6682#   6753#   6823#   6841#   6858#
M$IOSE        1#    1996#
M$LDRO        1#    1996#   4621#   4626#   4631#   4637#   4648#   4658#   4712#   4772#   4813#
M$MASK        1#    1996#
M$MCHI        1#    1996#
M$MCLO        1#    1996#
M$MSK1        1#    1996#
M$POP         1#    1996#   2153#   3179#   3192#   3206#   3228#   3249#   3263#   3278#   3391#   4584#   4601#   4782#   4798#
            4831#   4852#   4874#   6525#   6532#   6840#   6902#   6942#
M$PRIN        1#    1996#   3168#   3185#   3198#   3212#   3221#   3234#   3242#   3255#   3269#   3500#   3508#   3551#   3579#
            3589#   3648#   3716#   3767#   3781#   3806#   3829#   3839#   3854#   3868#   3875#   3900#   3916#   3931#   3952#
            3972#   4010#   4020#   4029#   4074#   4269#   4299#   4449#   4492#   4690#   4905#   4950#   4995#   5004#   5043#
            5058#   5084#   5104#   5204#   5284#   5331#   5361#   5435#   5484#   5501#   5962#   6044#   6139#   6162#
M$PUSH        1#    1996#   1998#   2122#   3166#   3183#   3196#   3210#   3232#   3253#   3267#   3366#   4574#   4594#   4610#
            4794#   4808#   4841#   4863#   4890#   4891    6522#   6529#   6858#
M$PUT         1#    1996#   3168#   3185#   3198#   3212#   3221#   3234#   3242#   3255#   3269#   3500#   3508#   3551#   3579#
            3589#   3648#   3716#   3767#   3781#   3806#   3829#   3839#   3854#   3868#   3875#   3900#   3916#   3931#   3952#
            3972#   4010#   4020#   4029#   4074#   4269#   4299#   4449#   4492#   4690#   4747#   4757#   4764#   4905#   4950#
            4995#   5004#   5043#   5058#   5084#   5104#   5204#   5284#   5331#   5361#   5435#   5484#   5501#   5962#   6044#
            6139#   6162#
M$PUT1        1#    1996#   3168#   3170    3172    3173    3174    3185#   3186    3187    3198#   3199    3200    3201    3212#
            3213    3214    3215    3216    3221#   3222    3223    3234#   3235    3236    3237    3242#   3243    3244    3255#
            3256    3257    3258    3269#   3270    3271    3272    3273    3500#   3501    3508#   3509    3551#   3552    3579#
            3580    3589#   3590    3648#   3649    3650    3716#   3717    3718    3767#   3768    3781#   3782    3784    3785
            3786    3806#   3807    3829#   3830    3839#   3840    3841    3842    3843    3844    3854#   3855    3856    3857
            3868#   3869    3870    3875#   3876    3877    3878    3900#   3901    3902    3903    3904    3916#   3917    3918
            3919    3920    3931#   3932    3933    3934    3935    3952#   3953    3972#   3973    4010#   4011    4012    4020#
            4022    4023    4029#   4030    4031    4074#   4075    4269#   4270    4271    4272    4273    4274    4299#   4300
            4301    4302    4303    4304    4449#   4450    4492#   4493    4690#   4691    4747#   4748    4749    4750    4757#
            4758    4759    4760    4764#   4765    4766    4767    4905#   4906    4950#   4951    4995#   4996    5004#   5005
            5043#   5044    5045    5058#   5059    5060    5084#   5085    5086    5104#   5105    5106    5204#   5205    5206
```

```
          5284#  5285   5286   5287   5331#  5332   5361#  5362   5435#  5436   5484#  5485   5501#  5502   5962#
          5963   6044#  6045   6046   6047   6139#  6140   6162#  6163
M$RADI       1#  1996#  3565#  4681#  4981#  5936#  6084#  6865#  6875#  6880#  6885#  6893#
M$RBRO       1#  1996#
M$RNRO       1#  1996#  4648#  4650   4658#  4660   4712#  4714
M$SETS       1#  1996#  1998#  2122#  3166#  3183#  3196#  3210#  3232#  3253#  3267#  3366#  4574#  4594#  4610#
          4794#  4808#  4841#  4863#  4891#  6522#  6529#  6858#
M$STAR       1#  1996#
M$SVC        1#  1996#  3168#  3176   3179#  3180   3185#  3189   3192#  3193   3198#  3203   3206#  3207   3212#
          3218   3221#  3225   3228#  3229   3234#  3239   3242#  3246   3249#  3250   3255#  3260   3263#  3264
          3269#  3275   3278#  3279   3282#  3500#  3503   3508#  3511   3551#  3554   3562#  3579#  3582   3589#
          3592   3648#  3652   3716#  3720   3767#  3770   3781#  3788   3806#  3809   3829#  3832   3839#  3846
          3854#  3859   3868#  3872   3875#  3880   3900#  3906   3916#  3922   3931#  3937   3952#  3955   3972#
          3975   4010#  4014   4020#  4025   4029#  4033   4074#  4077   4269#  4276   4299#  4306   4449#  4452
          4492#  4495   4584#  4585   4617#  4621#  4622   4626#  4627   4631#  4632   4637#  4638   4648#  4649
          4658#  4659   4671#  4678#  4690#  4693   4712#  4713   4747#  4751   4757#  4761   4764#  4768   4772#
          4773   4775#  4782#  4783   4798#  4799   4813#  4814   4824#  4831#  4832   4845#  4852#  4853   4867#
          4874#  4875   4905#  4908   4950#  4953   4962#  4968#  4978#  4995#  4998   5004#  5007   5031#  5043#
          5047   5058#  5062   5084#  5088   5104#  5108   5204#  5208   5284#  5289   5331#  5334   5361#  5364
          5435#  5438   5484#  5487   5501#  5504   5834   5852   5865   5933#  5962#  5965   5971   6044#  6049
          6081#  6139#  6142   6162#  6165   6221#  6231   6444   6452#  6463   6484   6503   6561#  6572   6614
          6621#  6674   6682#  6753   6823#  6840#  6841
M$TLAB       1#  1996#  3176#  3180#  3189#  3193#  3203#  3207#  3218#  3225#  3229#  3239#  3246#  3250#  3260#
          3264#  3275#  3279#  3503#  3511#  3554#  3562#  3582#  3592#  3652#  3720#  3770#  3788#  3809#  3832#
          3846#  3859#  3872#  3880#  3906#  3922#  3937#  3955#  3975#  4014#  4025#  4033#  4077#  4276#  4306#
          4452#  4495#  4585#  4617#  4622#  4627#  4632#  4638#  4649#  4659#  4671#  4678#  4693#  4713#  4751#
          4761#  4768#  4773#  4775#  4783#  4799#  4814#  4824#  4832#  4853#  4875#  4908#  4953#  4962#  4968#
          4978#  4998#  5007#  5031#  5047#  5062#  5088#  5108#  5208#  5289#  5334#  5364#  5438#  5487#  5504#
          5834#  5852#  5865#  5933#  5965#  5971#  6049#  6081#  6142#  6165#  6221#  6231#  6444#  6452#  6463#
          6484#  6503#  6561#  6572#  6614#  6621#  6674#  6682#  6753#  6823#  6841#
M$TSTL       1#  1996#  3176#  3180#  3189#  3193#  3203#  3207#  3218#  3225#  3229#  3239#  3246#  3250#  3260#
          3264#  3275#  3279#  3503#  3511#  3554#  3562#  3582#  3592#  3652#  3720#  3770#  3788#  3809#  3832#
          3846#  3859#  3872#  3880#  3906#  3922#  3937#  3955#  3975#  4014#  4025#  4033#  4077#  4276#  4306#
          4452#  4495#  4585#  4617#  4622#  4627#  4632#  4638#  4649#  4659#  4671#  4678#  4693#  4713#  4751#
          4761#  4768#  4773#  4775#  4783#  4799#  4814#  4824#  4832#  4853#  4875#  4908#  4953#  4962#  4968#
          4978#  4998#  5007#  5031#  5047#  5062#  5088#  5108#  5208#  5289#  5334#  5364#  5438#  5487#  5504#
          5834#  5852#  5865#  5933#  5965#  5971#  6049#  6081#  6142#  6165#  6221#  6231#  6444#  6452#  6463#
          6484#  6503#  6561#  6572#  6614#  6621#  6674#  6682#  6753#  6823#  6841#
M$WORD       1#  1996#  2052#  2061   2108#  2110   3282#  3562#  3564   3565#  4678#  4680   4681#  4775#  4824#
          4845#  4867#  4968#  4978#  4980   4981#  5031#  5834#  5835   5836   5837   5852#  5853   5854   5855
          5865#  5866   5867   5868   5933#  5935   5936#  5971#  5972   5973   5974   6081#  6083   6084#  6231#
          6232   6233   6234   6444#  6445   6446   6447   6463#  6464   6465   6466   6484#  6485   6486   6487
          6503#  6504   6505   6506   6572#  6573   6574   6575   6614#  6615   6616   6617   6674#  6675   6676
          6677   6753#  6754   6755   6756   6865#  6875#  6880#  6885#  6893#  6938   6939
M$XFER       1#  1996#
NODCL     2002#  2934   2935   2936   2937   2938   2939   2940   2941   2942   2943   2944   2945   2946   2947
          2948   2949   2950   2951   2952   2953   2954   2958   2959   2960   2961   2962   2963   2964   2965
          2966   2967   2968   2969   2970   2971   2972   2973   2974   2975   2979   2980   2981   2982   2983
          2988   2989   2990   2991   2992   2995   2996   2997   2998   2999   3000   3001   3002   3005   3006
          3007   3008   3009   3010   3016   3017   3018   3019   3022   3023   3024   3025   3027   3028   3031
          3034   3035   3036   3037   3038   3039   3040   3041   3042   3043   3044   3045   3046   3047   3048
          3049   3052   3053   3054   3055   3056   3057   3058   3061   3062   3063   3064   3065   3066   3069
          3070   3071   3074   3077   3078   3079   3080   3081   3082   3083   3084   3087   3088   3089   3094
          3605   3606   3607   3608   3609   3610   3611   3612   3613   3614   3615   3616   3617   3618   3619
          3620   3621   3622   3623   3624   3625   3626
OPEN         1#  1996#
```

```
 CZCLKCO DMR,DMC-11 DATA COMM. LINK TEST       MACY11 30A(1052)  23-MAR-82  16:45  PAGE 204
CZCLKC.P11    19-MAR-82 18:32              CROSS REFERENCE TABLE -- MACRO NAMES

POINTE      1#    1996#   2008
PRINTB      1#    1996#   3167    3184    3197    3211    3220    3233    3241    3254    3268
PRINTF      1#    1996#   3499    3507    3550    3578    3588    3647    3715    3766    3780    4009    4019    4028    4073
         4448     4491    4689    4904    4949    4994    5003    5042    5057    5083    5103    5203    5283    5330    5360
         5434     5483    5500    5961    6043    6138    6161
PRINTS      1#    1996#   3805    3828    3838    3853    3867    3874    3899    3915    3930    3951    3971    4268    4298
PRINTX      1#    1996#
READBU      1#    1996#   4670
READEF      1#    1996#   4620    4625    4630    4636
RFLAGS      1#    1996#
SETPRI      1#    1996#   4771    4812
SETVEC      1#    1996#   4746    4756    4763
SLASH       1#    1996#
STARS       1#    1996#
SVC         1#    1996#
XFER        1#    1996#   3282#   4775#   4824#   4845#   4867#   4968#   5031#
XFERF       1#    1996#
XFERT       1#    1996#


. ABS.  046522     000


 ERRORS DETECTED:  0

 CZCLKC,CZCLKC.LST/CRF/SOL=SVC34R.MLB,CZCLKC.P11
 RUN-TIME: 26 32 4 SECONDS
 RUN-TIME RATIO: 93/62=1.4
 CORE USED:  22K  (43 PAGES)
```