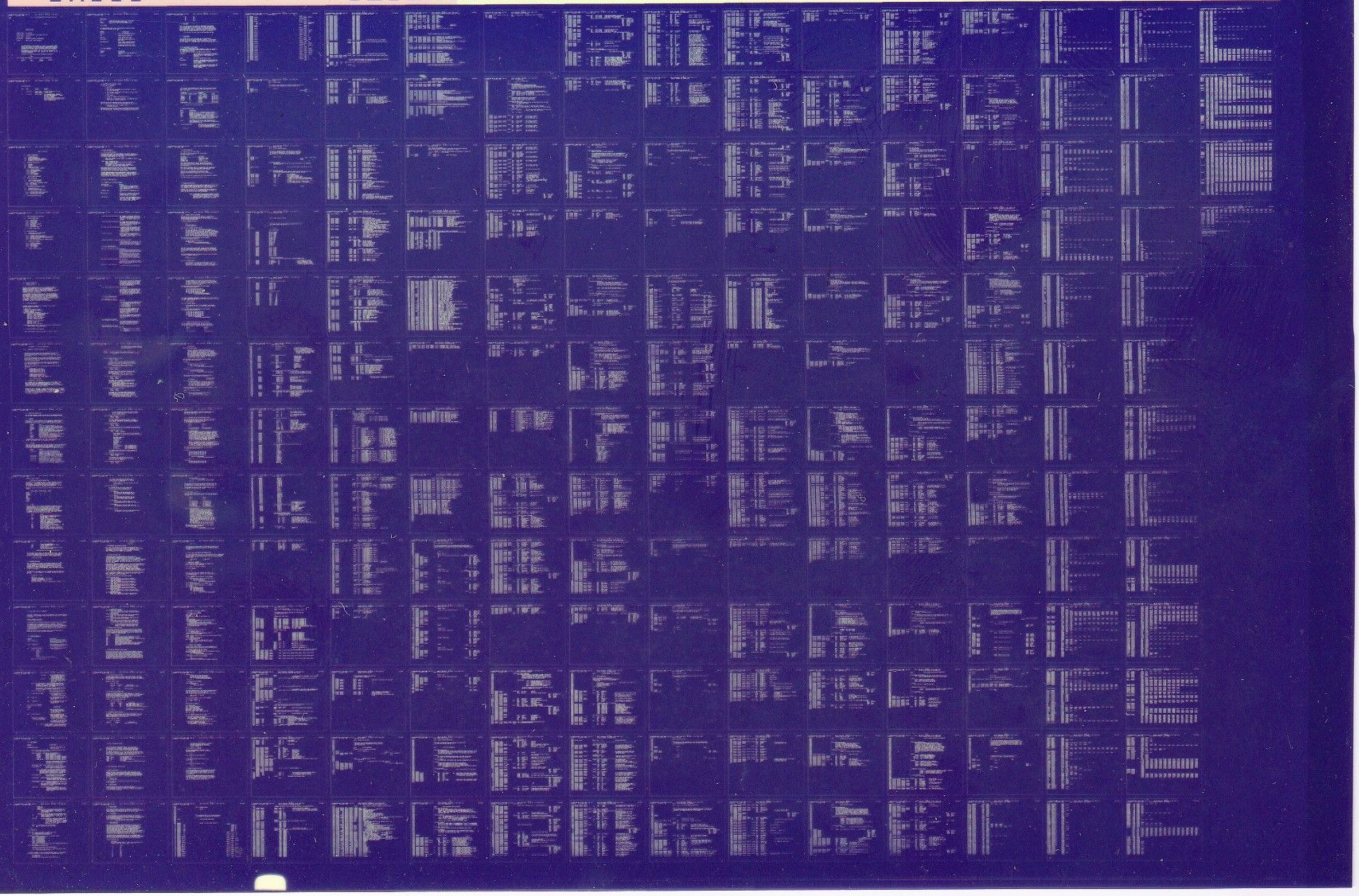
DMR11
DMC11

DMR/C11 DCLT
CZCLKB0

AH-F593B-MC
FICHE 1 OF 1



```
2207                                .TITLE  CZCLKB0 DMR,DMC-11 DATA COMM. LINK TEST
2208
```

.REM &

IDENTIFICATION

PRODUCT CODE: AC-F591B-MC

PRODUCT NAME: CZCLKB0 DMR,DMC-11 DATA COMM. LINK TEST

PRODUCT DATE: 26-OCT-81

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

D 1

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS DCLT (DATA COMMUNICATION LINK TEST) PROGRAM IS MEANT TO PROVIDE FIELD SERVICE WITH A TOOL TO MAINTAIN 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 ^IAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE ^IAGNOSTIC 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.
	WHERE "/B" IS OPTIONAL: DEFAULT IS PRINT WORDS "/B" CAUSES PRINT BYTES	IF "-EEEEEE" IS NOT SPECIFIED THEN THE CONTENTS OF 'SSSSSS" IS PRINTED IN WORD FORMAT.

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 DFFAULTS 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.

D 2

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	MMMMMMM
XXXXXX	XXXXXX	

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

.....
.....

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

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.

```
 4217
 4218                                   .SBTTL  PROGRAM HEADER
 4248
 4249  002000                                   BGNMOD
 4250
 4255
 4259
 4260
 4280
 4281                                   ;++
 4282                                   ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
 4283                                   ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
 4284                                   ;--
 4285
 4286  002000                                   POINTER BGNRPT,BGNAU,BGNDU
 4287
 4304
 4305
 4306  002000                                   HEADER  CZCLK,B,0,1800.,0,#PRI07
   (4) 002000                                                                    L$NAME::
   (4) 002000      103                                                                   .ASCII /C/
   (4) 002001      132                                                                   .ASCII /Z/
   (4) 002002      103                                                                   .ASCII /C/
   (4) 002003      114                                                                   .ASCII /L/
   (4) 002004      113                                                                   .ASCII /K/
   (6) 002005      000                                                                   .BYTE   0
   (6) 002006      000                                                                   .BYTE   0
   (5) 002007      000                                                                   .BYTE   0
   (5) 002010                                                                    L$REV::
   (4) 002010      102                                                                   .ASCII  /B/
   (5) 002011                                                                    L$DEPO::
   (4) 002011      060                                                                   .ASCII  /0/
   (5) 002012                                                                    L$UNIT::
   (4) 002012  000000                                                                    .WORD   0
   (5) 002014                                                                    L$TIML::
   (4) 002014  003410                                                                    .WORD   1800.
   (5) 002016                                                                    L$HPCP::
   (4) 002016  046060                                                                    .WORD   L$HARD
   (5) 002020                                                                    L$SPCP::
   (4) 002020  000000                                                                    .WORD   0
   (5) 002022                                                                    L$HPTP::
   (4) 002022  002130                                                                    .WORD   L$HW
   (5) 002024                                                                    L$SPTP::
   (4) 002024  000000                                                                    .WORD   0
   (5) 002026                                                                    L$LADP::
   (4) 002026  046442                                                                    .WORD   L$LAST
   (5) 002030                                                                    L$STA::
   (4) 002030  000000                                                                    .WORD   0
   (5) 002032                                                                    L$CO::
   (4) 002032  000000                                                                    .WORD   0
   (5) 002034                                                                    L$DTYP::
   (4) 002034  000000                                                                    .WORD   0
   (5) 002036                                                                    L$APT::
   (4) 002036  000000                                                                    .WORD   0
   (5) 002040                                                                    L$DTP::
   (4) 002040  002124                                                                    .WORD   L$DISPATCH
```

```
   (5)  002042
   (4)  002042  000340
   (5)  002044
   (4)  002044  000000
   (5)  002046
   (4)  002046  000000
   (5)  002050
   (4)  002050     003
   (3)  002051     003
   (5)  002052
   (4)  002052  000000
   (5)  002054  000000
   (5)  002056
   (4)  002056  000000
   (5)  002060
   (4)  002060  012410
   (5)  002062
   (4)  002062  033732
   (5)  002064
   (4)  002064  000000
   (5)  002066
   (4)  002066  000000
   (5)  002070
   (4)  002070  034702
   (5)  002072
   (4)  002072  034674
   (5)  002074
   (4)  002074  000000
   (5)  002076
   (4)  002076  012424
   (5)  002100
   (4)  002100  104035
   (5)  002102
   (4)  002102  000000
   (5)  002104
   (4)  002104  033746
   (5)  002106
   (4)  002106  034650
   (5)  002110
   (4)  002110  034646
   (5)  002112
   (4)  002112  033740
   (5)  002114
   (4)  002114  000000
   (5)  002116
   (4)  002116  000000
   (5)  002120
   (4)  002120  000000
4307
4314
```

```
L$PRIO::
        .WORD   #PRIO7
L$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
  (4)  002122  000001                                                   .WORD   1
  (3)  002124                                                   L$DISPATCH::
  (6)  002124  034710                                                   .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
(3) 002126  000010                                                      .WORD   L10000-L$HW/2
(3) 002130                                                          L$HW::
(3) 002130                                                          DFPTBL::

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

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


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

002132  160170                         .WORD   160170          ;[2] CSR ADDRESS
002134  000300                         .WORD   300             ;[4] INTERRUPT VECTOR
002136  000240                         .WORD   240             ;[6] INTERRUPT PRIORITY (5)
002140  000000                         .WORD   0               ;[10] 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
(3) 002150                                                          L10000:
```

```
                                .SBTTL  GLOBAL EQUATES SECTION



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


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

```
   (1)                          ;
   (1)                          ; PRIORITY LEVEL DEFINITIONS
   (1)                          ;
   (1)          000340          PRI07== 340
   (1)          000300          PRI06== 300
   (1)          000240          PRI05== 240
   (1)          000200          PRI04== 200
   (1)          000140          PRI03== 140
   (1)          000100          PRI02== 100
   (1)          000040          PRI01== 40
   (1)          000000          PRI00== 0
   (1)                          ;
   (1)                          ;OPERATOR FLAG BITS
   (1)                          ;
   (1)          000004          EVL==        4
   (1)          000010          LOT==       10
   (1)          000020          ADR==       20
   (1)          000040          IDU==       40
   (1)          000100          ISR==      100
   (1)          000200          UAM==      200
   (1)          000400          BOE==      400
   (1)          001000          PNT==     1000
   (1)          002000          PRI==     2000
   (1)          004000          IXE==     4000
   (1)          010000          IBE==    10000
   (1)          020000          IER==    20000
   (1)          040000          LOE==    40000
   (1)          100000          HOE==   100000
 4479
```

```
                        ;***** 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
```

```
4537   000012                    DVI=    12              ;DEVICE ABOUT TO INIT
4538   000014                    DCK=    14              ;DATA COMPARISON RESULTS
4539
4540   000020                    DLE=    20              ;DATA COMPARISON LENGH ERROR
4541   000022                    DDE=    22              ;DATA COMPARISON DATA ERROR
4542   000024                    EOP=    24              ;END OF PASS
4543
4544                        ;;;;EQUATES FOR FLAG WORD;;;;;;
4545
4546   000001                    ININT= 1                ;INPUT INT. REC.
4547   000002                    OTINT= 2                ;OUTPUT INT REC
4548   000004                    QRX=   4                ;RX QUED /COMPL
4549   000010                    QTX=    10              ;TX QUED/COMPL
4550   000020                    CTX=    20              ;TX COMPL AND IN TXSEL4 AND TSEL6
4551   000040                    CRX=    40              ;RX COMPL AND IN TSEL4 AND TSEL6
4552   000100                    ERX=    100             ;EXPECT TO GET A RX COMPLED
4553   000200                    ETX=    200             ;EXPECT TO GET A TX COMPLETED
4554   000400                    DLLGA-  400             ;DOWN LINE LOAD GO AHEAD BIT
4555   001000                    DMRRUN= 1000            ;DMR RUN MODE EXPECTED
4556   002000                    BTUP=   2000            ;BASE TABLE UPDATE REQUESTED
4557
4558                        ; SPECIAL CLI CODES FOR ''CHAR'' ARGUMENT IN CLI CALLS
4559                        ;  (COMMAND LINE INTERPRETER DEFINITIONS)
4560   000000                    CLIERR= 0
4561   000001                    CLIEXI= 1
4562   000002                    CLIBR=  2
4563   000003                    CLIBIF= 3
4564   000004                    CLISPA= 4
4565   000005                    CLINUM= 5
4566   000006                    CLIALP= 6
4567   000007                    CLIALN= 7
4568   000010                    CLIOCT= 8.
4569   000011                    CLIDEC= 9.
4570   000012                    CLISTR= 10.
4571
4572                        ; DEFS FOR COMMAND LINE INTERPRETATION ACTION VALUES
4573   000000                    NULL=0
4574   000001                    CLEAR=1
4575   000002                    SHOW=2
4576   000003                    CHECK=3
4577   000004                    RUN=4
4578   000005                    HLP=5
4579   000006                    CSHEXP=6
4580   000007                    CSHTRN=7
4581   000010                    SETEXP=10
4582   000011                    SETTRN=11
4583   000012                    SIZE=12
4584   000013                    QCOPY=13
4585   000014                    NUM=14
4586   000015                    OPRMSG=15
4587   000016                    STATUS=16
4588   000017                    ENDQO=17
4589   000020                    CMSG0=20
4590   000021                    CMSG1=21
4591   000022                    CMSG2=22
4592   000023                    CMSG3=23
```

```
4593    000024                  CMSG4=24
4594    000025                  CMSG5=25
4595    000026                  CMSG6=26
4596    000027                  ATVMOD=27
4597    000030                  PASMOD=30
4598    000031                  RECMOD=31
4599    000032                  LISMOD=32
4600    000033                  DLLMOD=33
4601    000034                  TRAMOD=34
4602    000035                  TALMOD=35
4603    000036                  NO=36
4604    000037                  ECHO=37
4605    000040                  CRC=40
4606    000041                  PROTO=41
4607    000042                  PASC=42
4608    000043                  MOP=43
4609    000044                  TTLLOP=44
4610    000045                  CBLLOP=45
4611    000046                  LMDLOP=46
4612    000047                  RMDLOP=47
4613    000050                  NOTNUF=50
4614    000051                  BADCHR=51
4615    000052                  DMPS=52
4616    000053                  DMPE=53
4617    000054                  DMPQ=54
4618    000055                  PRNT=55
4619    000056                  MOSC=56                 ;MODEM/NOMODEM REV B BY EC
4620    000057                  EXIT=57                 ;EXIT COMMAND REV B BY EC
4621    000060                  SETET=60                ;S E=T COMMAND REV B BY EC
4622                    ;FOLLOWING EQUATES USED IN REPORT CLI    REV B BY EC
4623    000001                  RPHLP=1                 ;HELP COMMAND
4624    000002                  RPEXT=2                 ;EXIT COMMAND
4625    000003                  RPLOG=3                 ;PRINT EVENT LOG COMMAND
4626    000004                  RPSWE=4                 ;BASE/ERROR COMMAND
4627    000005                  RPSWF=5                 ;BASE/FULL COMMAND
4628    000006                  RPSWO=6                 ;BASE/OFFSET
4629    000007                  RNOTNF=7                ;MORE COMMAND NEEDED
4641
4642                    ;***** DEVICE DEPENDENT EQUATES
4643                    ; MODEM SIGNAL BIT DEFINITONS
4644                    ;       IF SIGNAL AVAILABLE IN DEVICE, EQUATE NAME TO BIT POSITION,
4645                    ;       ELSE EQUATE IT TO = 0
4646
4647    000004                  CTS=    BIT2            ;CLEAR TO SEND (CIRCUIT CB)
4648    000010                  DSR=    BIT3            ;DATA SET READY (CIRCUIT CC)
4649    000001                  DCD=    BIT0            ;DATA CARRIER DETECT (CIRCUIT CF)
4650    000040                  RTS=    BIT5            ;REQUEST TO SEND (CIRCUIT CA)
4651    000200                  RI=     BIT7            ;RING INDICATOR (CIRCUIT CE)
4652    040000                  SQD=    BIT14           ;SIGNAL QUALITY DETECT (CIRCUIT CG)
4653    001000                  TM=     BIT9            ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)
4654
 662
 663
4664                    ; DEVICE SIGNALS
4665    000040                  RQI=    BIT5            ;REQUEST IN
4666    000200                  RDI=    BIT7            ;READY IN
```

```
4667    000200          RDO=    BIT7
4668    000001          BACC=   BIT0          ;BUFFER ADDR. CHAR COUNT
4669    040000          MCLR=   BIT14         ;MASTER CLEAR
4670    004000          LULOOP= BIT11         ;LINE UNIT LOOP(TTL)
4671    000400          MAINTB= BIT8          ;MAINT MODE BIT
4672    002000          HALFDB= BIT10         ;HALF DUPLEX BIT
4673    000004          RXBIT=  BIT2          ;RX BIT
4674    000100          IEO=    BIT6          ;ENABLE OUTPUT INTERRUPT BIT
4675
```

```
                                .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
```

```
                002320  166632  016741
4726            002324                          EMSG4:
4727            002324                          MSG5:                           ;''INTERPROCESSOR TEST PROGRAM'S (ITEP)'' MESSAGE
4728                                                                            ; #1, (DP1:)
4729            002324  077577  040444  052040          .ASCII  <177><177>/$A THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG./
                002332  042510  050440  044525
                002340  045503  041040  047522
                002346  047127  043040  054117
                002354  045040  046525  042520
                002362  020104  053117  051105
                002370  052040  042510  046040
                002376  055101  020131  047504
                002404  027107
4730            002406  005015  077401  077577          .ASCIZ  <15><12><001><177><177><177><177>
                002414  000177
4731            002416                          EMSG5:
4732            002416                          MSG6:                           ;ALPHA-NUMERICS (OR FUTURE COMM TURNAROUND MSG)
4733            002416  022043  021041  023040          .ASCII  /#$!'' &'()*+,-.0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ/
                002424  024047  025051  026053
                002432  027055  030460  031462
                002440  032464  033466  034470
                002446  035472  036474  037476
                002454  040500  041502  042504
                002462  043506  044510  045512
                002470  046514  047516  050520
                002476  051522  052524  053526
                002504  054530     132
4734            002507     057  056133  057135          .ASCIZ  ?/[\]^_%?
                002514  022537     000
4735            002517                          EMSG6:
4736                    002520                          .EVEN
4737
4738                                            ; ***************************************************************************
4739                                            ;THESE THREE STORAGE AREAS MUST NOT BE SEPERATED !!!!
4740
4741            002520  047045  040445          OPBFPT: .ASCII  /%N%A/
4742            002524  000122                  OPBUF:  .BLKB   82.             ;BUFFER FOR OPERATOR SPEC'D MESSAGES
4743            002646                          OPEND:
4744
4745                                            ; THE ABOVE THREE LINES MUST BE KEPT TOGETHER
4746                                            ; ***************************************************************************
4747
4748            002646      033                 MSG8:   .BYTE   33              ;RECEIVE BUFFER FILL PATTERN
4749            002647                          EMSG8:
4750
4751                                            ; DOWN-LINE-LOAD MESSAGE DEFINITIONS
4752                                                    ;;;;ENTER MOP MODE MESSAGE FORMAT
4753                                                    ;;;;THE NODE WILL ENTER MAINTENANCE MODE ONLY IF THE PASSWORD MATCHES.
4754            002647      006                 DLLM1:  .BYTE   6       ;BINARY CODE FOR MAINTENANCE MODE
4755            002650      000                 PASS1:  .BYTE   0       ;PASSWORD BYTE #1   LEGAL VALUE 0 - 255
4756            002651      000                 PASS2:  .BYTE   0       ;;VALUE IN BYTE 1 IS DUPLICATED HERE
4757            002652      000                 PASS3:  .BYTE   0       ;;;AND HERE
4758            002653      000                 PASS4:  .BYTE   0       ;;;;AND HERE.
4759            002654                          DLLM1E:                 ;END ENTER MOP MODE MESSAGE FORMAT
4760                                                    ;;;;MEMORY LOAD WITH TRANSFER ADDRESS MESSAGE FORMAT
4761            002654      000                 DLLM2:  .BYTE   0       ;CODE
```

```
 4762  002655  000                        .BYTE   0       ;LOAD NUMBER
 4763  002656  006                        .BYTE   6       ;LOAD ADDRESS LSB
 4764  002657  000                        .BYTE   0
 4765  002660  000                        .BYTE   0
 4766  002661  000                        .BYTE   0       ;LOAD ADDRESS
 4767                                ;
 4768                                ;IMAGE DATA
 4769                                ;
 4770  002662  005037  000006           CLR     @#6
 4771  002666  012706  001000           MOV     #1000,SP
 4772  002672  012701  177560           MOV     #177560,R1      ;SET UP TTY
 4773  002676  010700                    MOV     PC,R0           ;MAKE ADDR.PIC
 4774  002700  062700  000034           ADD     #<MSG-.>,R0     ;ADDRESS MSG.
 4775  002704  105761  000004     1$:   TSTB    4(R1)           ;TTY READY?
 4776  002710  100375                    BPL     1$              ;WAIT TIL YES
 4777  002712  112061  000006           MOVB    (R0)+,6(R1)     ;TYPE A CHAR
 4778  002716  001372                    BNE     1$              ;KEEP GOING
 4779  002720  012737  000026  000024   MOV     #26,@#24        ;SET UP POWER FAIL
 4780  002726  005037  000026           CLR     @#26            ;MAKE SURE T BIT CLAER
 4781  002732  000777                    BR      .               ;JUMP ON YOURSELF
 4782  002734  006412  047502  052117 MSG:    .ASCII  <12><15>/BOOT MESSAGE WAS RECEIVED SUCCESSFULLY -END OF TEST!!/
       002742  046440  051505  040523
       002750  042507  053440  051501
       002756  051040  041505  044505
       002764  042526  020104  052523
       002772  041503  051505  043123
       003000  046125  054514  026440
       003006  047105  020104  043117
       003014  052040  051505  020524
       003022     041
 4783  003023     012  027015  027056        .ASCIZ  <12><15>/....RELOAD PROGRAM..../
       003030  051056  046105  040517
       003036  020104  051120  043517
       003044  040522  027115  027056
       003052  000056
 4784  003054     006                        .BYTE   6               ;NEXT FOUR BYTES CONTAINS TRANSFER ADDRESS
 4785  003055     000                        .BYTE   0               ;;OF PROGRAM JUST DOWNLINE LOADED.
 4786  003056     000                        .BYTE   0               ;;;THIS PROGRAM STARTS AT ADDRESS 6.
 4787  003057     000                        .BYTE   0
 4788  003060                        DLLM2E:                 ;END MEMORY LOAD MESSAGE FORMAT
 4789
 4790                                .EVEN
 4791
```

```
 4793                                   ;COMMAND LINE BUFFER, DATA LOCATIONS AND MESSAGES FOR ACTION ROUTINES
 4794
 4795  003060  000122                   CMDBUF: .BLKB   82.              ;BUFFER FOR OPERATOR COMMANDS
 4796  003202  000000                   KEYWD1: .WORD   0                ;THIS LOC WILL =1 IF CLEAR TYPED, 2 FOR SHOW,
 4797                                                                    ; A 4 IF RUN WAS TYPED, 5 IF HELP WAS TYPED
 4798  003204  000000                   QUALFG: .WORD   0                ;THIS LOC HOLDS QUALIFIER VALUE (SIZE OR COPY)
 4799  003206  000000                   QUALVL: .WORD   0
 4800  003210  013232                   HLPTAB: .WORD   HLP1
 4801  003212  013245                           .WORD   HLP2
 4802  003214  013362                           .WORD   HLP3
 4803  003216  013447                           .WORD   HLP3A
 4804  003220  013474                           .WORD   HLP4
 4805  003222  013553                           .WORD   HLP4A
 4806  003224  013631                           .WORD   HLP5
 4807  003226  013721                           .WORD   HLP6
 4808  003230                         HLPEND:
 4809                                   ;INDEX TABLE FOR REPORT 'RPT>' HELP MESSAGES  REV B BY EC
 4810  003230  014057                   RHLPTB: .WORD   RHLP1
 4811  003232  014102                           .WORD   RHLP2
 4812  003234  014135                           .WORD   RHLP3
 4813  003236  014166                           .WORD   RHLP4
 4814  003240  014220                           .WORD   RHLP5
 4815  003242  014257                           .WORD   RHLP6
 4816  003244  014316                           .WORD   RHLP7
 4817  003246  000000                   RHLPEN: .WORD   0                ;END OF REPORT HELP TABLE
 4818                                   ;INDEX TABLE FOR DMR BASE TABLE DATA DESCRIPTION MESSAGES REV B BY EC
 4819  003250  020342                   DMRIND: .WORD   DMR000
 4820  003252  020402                           .WORD   DMR001
 4821  003254  020432                           .WORD   DMR002
 4822  003256  020467                           .WORD   DMR003
 4823  003260  020532                           .WORD   DMR004
 4824  003262  020566                           .WORD   DMR005
 4825  003264  020620                           .WORD   DMR006
 4826  003266  020663                           .WORD   DMR007
 4827  003270  020717                           .WORD   DMR010
 4828  003272  020751                           .WORD   DMR011
 4829  003274  021003                           .WORD   DMR012
 4830  003276  021035                           .WORD   DMR013
 4831  003300  021065                           .WORD   DMR014
 4832  003302  021114                           .WORD   DMR015
 4833  003304  021146                           .WORD   DMR016
 4834  003306  021176                           .WORD   DMR017
 4835  003310  021226                           .WORD   DMR020
 4836  003312  021255                           .WORD   DMR021
 4837  003314  021307                           .WORD   DMR022
 4838  003316  021333                           .WORD   DMR023
 4839  003320  021365                           .WORD   DMR024
 4840  003322  021410                           .WORD   DMR025
 4841  003324  021463                           .WORD   DMR026
 4842  003326  021513                           .WORD   DMR027
 4843  003330  021544                           .WORD   DMR030
 4844  003332  021627                           .WORD   DMR031
 4845  003334  021664                           .WORD   DMR032
 4846  003336  021721                           .WORD   DMR033
 4847  003340  021756                           .WORD   DMR034
 4848  003342  022042                           .WORD   DMR035
```

```
 4849  003344  022107                          .WORD   DMR036
 4850  003346  022154                          .WORD   DMR037
 4851  003350  022212                          .WORD   DMR040
 4852  003352  022245                          .WORD   DMR041
 4853  003354  022303                          .WORD   DMR042
 4854  003356  022351                          .WORD   DMR043
 4855  003360  022405                          .WORD   DMR044
 4856  003362  022441                          .WORD   DMR045
 4857  003364  022472                          .WORD   DMR046
 4858  003366  022540                          .WORD   DMR047
 4859  003370  022602                          .WORD   DMR050
 4860  003372  022630                          .WORD   DMR051
 4861  003374  022664                          .WORD   DMR052
 4862  003376  022711                          .WORD   DMR053
 4863  003400  022744                          .WORD   DMR054
 4864  003402  022766                          .WORD   DMR055
 4865  003404  023041                          .WORD   DMR056
 4866  003406  023114                          .WORD   DMR057
 4867  003410  023136                          .WORD   DMR060
 4868  003412  023170                          .WORD   DMR061
 4869  003414  023222                          .WORD   DMR062
 4870  003416  023272                          .WORD   DMR063
 4871  003420  023342                          .WORD   DMR064
 4872  003422  023376                          .WORD   DMR065
 4873  003424  023432                          .WORD   DMR066
 4874  003426  023475                          .WORD   DMR067
 4875  003430  023540              DMREND:     .WORD   DMR177          ;NO DMR MESSAGES MUST FOLLOW DMREND
 4876
 4877                              ;INDEX  TABLE FOR DMC BASE TABLE DATA DESCRIPTION MESSAGES REV B BY EC
 4878  003432  020322              DMCIND:     .WORD   DMUNKN
 4879  003434  020322                          .WORD   DMUNKN
 4880  003436  023604                          .WORD   DMC002
 4881  003440  023625                          .WORD   DMC003
 4882  003442  023662                          .WORD   DMC004
 4883  003444  023723                          .WORD   DMC005
 4884  003446  023756                          .WORD   DMC006
 4885  003450  024013                          .WORD   DMC007
 4886  003452  024050                          .WORD   DMC010
 4887  003454  024103                          .WORD   DMC011
 4888  003456  024125                          .WORD   DMC012
 4889  003460  024147                          .WORD   DMC013
 4890  003462  024206              DMCEND:     .WORD   DMC377          ;NO DMC MESSAGES MUST FOLLOW DMCEND
 4891
 4892  003464  014464  014473  014500  SHTYTB: .WORD   SHTYP0,SHTYP1,SHTYP2,SHTYP3,SHTYP4,SHTYP5,SHTYP6,SHTYP7
       003472  014505  014512  014520
       003500  014525  014533
 4893
 4894                              ; THE LIST OF BYTES BELOW ARE THE FIRST BYTES OF THE PREDEFINED MESSAGES
 4895                              ;  USED TO "SHOW" THE TRANSMIT AND COMPARE BUFFER CONTE.
 4896
 4897  003504    000     377     252  SHTAB:  .BYTE   0,377,252,125,203,177,043
       003507    125     203     177
       003512    043
 4898  003513                         SHTEND:
 4899          003514                         .EVEN
 4900
```

```
 4901  003514  014544                MODES:  .WORD   M00     ;ADDRESSES OF MODE TYPES IN ASCII
 4902  003516  014554                        .WORD   M01
 4903  003520  014565                        .WORD   M02
 4904  003522  014575                        .WORD   M03
 4905  003524  014604                        .WORD   M04
 4906  003526  014621                        .WORD   M05
 4907  003530  014626                        .WORD   M06
 4908
 4909  003532  014635                LOOPS:  .WORD   LP0     ;ADDRESSES OF LOOP TYPES IN ASCII
 4910  003534  014645                        .WORD   LP1
 4911  003536  014656                        .WORD   LP2
 4912  003540  014664                        .WORD   LP3
 4913  003542  014677                        .WORD   LP4
 4914
 4915                          ;COMMAND LINE TRAVERSE LOCATIONS (USED BY 'P$TRV')
 4916
 4917  003544  000000          P$BUFA: .WORD   0               ;LOC. TO HOLD ADDR. OF CMD LINE BUFFER
 4918  003546  000000          P$TREE: .WORD   0               ;LOC. TO HOLD ADDR. OF PARSING TREE
 4919  003550  000000          P$ACT:  .WORD   0               ;LOC. TO HOLD ADDR. OF ACTION ROUTINE
 4920  003552  000000          P$CNT:  .WORD   0               ;LOC. TO BE A COUNTER LOCATION
 4921  003554  000000          P$NUM:  .WORD   0               ;LOC. TO HOLD NUMERIC VALUE FROM PARSE
 4922  003556  000000          P$RADX: .WORD   0               ;LOC. TO HOLD RADIX USED(LO) AND +/-(HI BYTE)
 4923  003560     000          P$NNUF: .BYTE   0               ;RETURN =0 IF ENOUGH OF COMMAND FOUND
 4924  003561     000          P$GDBD: .BYTE   0               ;RETURN CODE 0 IF NO ERROR FOUND
 4925
```

```
                                .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 L/ T OF DMC OR DMR MESSAGES
```

```
 4983  007454  000000                BEND:   .WORD   0       ;LAST LOCATION IN BASE TABLE TO BE PRINTED
 4984
 4985
 4986
 4987                                ;MORE INDEPENDENT CODE STORAGE LOCATIONS
 4988  007456  000000                BDATA:  .WORD   0       ;POINTER TO BASE TABLE
 4989  007460  000000                LOGUNT: .WORD   0       ;LOC. TO HOLD LOGICAL UNIT NUMBER
 4990  007462  000000                PCADD:  .WORD   0       ;LOC. HOLD PC OF CALLIN ROUTINE
 4991  007464  000000                DCLFLG: .WORD   0       ;CLEANUP AND EXIT FLAG. 1=DO CLEANUP ROUTINE&EXIT
 4992  007466  000000                RESFLG: .WORD   0       ;LOC TO HOLD FLAG (-1) THAT A RESTART WAS GIVEN
 4993  007470  000000                MODTYP: .WORD   0       ;DCLT MODE OF OPERATION TYPE
 4994                                                        ;  (0=REC-ONLY, 1=TX-ONLY, 2=PASSIVE-LOOPBK,
 4995                                                        ;   3=ACTIVE-LOOPBK, 4=DOWN L.L., 5=TALK, 6=LISTEN)
 4996  007472  000000                MLTYP:  .WORD   0       ;MAINTENANCE LOOP TYPE (0=NONE, 1=INTERNAL TTL,
 4997                                                        ;  2=CABLE, 3=MODEM-ANALOG LOOPBK (LOCAL),
 4998                                                        ;  4=MODEM-DIGITAL LOOPBK (REMOTE), 5=MOP)
 4999  007474  000000                FHDPLX: .WORD   0       ;FULL OR HALF DUPLEX FLAG (1=FULL FROM P-TABLE)
 5000  007476  000002                PARAM:  .WORD   2       ;PROGRAM PARAMETERS
 5001                                                        ; BIT0= STATUS MSGS TO OPR PRINTED (1=YES)
 5002                                                        ; BIT1= DATA CHECKING DONE ON RCVD MSGS (1=YES)
 5003                                                        ; BIT2= ECHO (TRANSMIT) RCV'D MSG.(PASSIVE)(1=YES)
 5004                                                        ; BIT3= SPARE
 5005                                                        ; BIT4= CRC CALC./CHECK DONE (1=YES)
 5006                                                        ; BIT5= PROTOCOL EMULATION (1=YES)
 5007                                                        ; BIT6= SPARE
 5008  007500  000000                RPASS:  .WORD   0       ;PASS NUMBER FROM RUN COMMAND
 5009  007502  000000                FLAG:   .WORD   0       ;DEVICE FLAG WORD
 5010
 5011                                ;MODE DISPATCH TABLE
 5012  007504  040572                MODE:   .WORD   RXONLY  ;RX ONLY DISPATCH
 5013  007506  040624                        .WORD   TXONLY  ;TX ONLY DISPATCH
 5014  007510  040664                        .WORD   PLCK    ;PASSIVE LOOP BACK DISP
 5015  007512  040720                        .WORD   ALCK    ;ACTIVE LOOP BACK DISP
 5016  007514  042050                        .WORD   DLL     ;DOWN LINE LOAD DISP
 5017  007516  042670                        .WORD   TALCK   ;TALK MODE DISPATCH
 5018  007520  043110                        .WORD   LISCK   ;LISTEN MODE DISPATCH
 5019
 5020
 5021                                .SBTTL          CLOCK TABLES, EVENT LOG AND POINTERS
 5022  007522  000000                CLKCSR: .WORD   0       ;CLOCK CSR ADDRESS
 5023  007524  000000                CLKBR:  .WORD   0       ;CLOCK INTERRUPT LEVEL
 5024  007526  000000                CLKVEC: .WORD   0       ;CLOCK INTERRUPT VECTOR
 5025  007530  000074                CLKHZ:  .WORD   60.     ;CLOCK'S HERTZ RATE
 5026  007532  000000                CLKEN:  .WORD   0       ;CLOCK'S CSR VALUE TO INTRPT. ENABLE IT
 5027
 5028  007534  000000                TIMMIN: .WORD   0       ;PLACE TO KEEP TIME-SINCE-START
 5029  007536  000000                TIMSEC: .WORD   0
 5030  007540  000000                TIMTCK: .WORD   0       ;PLACE TO KEEP # OF TICKS/SEC
 5031
 5032  007542  000000                TIMER1: .WORD   0       ;EVENT TIMER #1 (TICKS)
 5033  007544  000000                TIMER2: .WORD   0       ;EVENT TIMER #2 (TICKS)
 5034  007546  000000                TIMERS: .WORD   0       ;EVENT TIMER #3 (SECONDS)
 5035
```

```
                                ;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  017240                  MOMSGS: .WORD   EVMCTS          ;CLEAR TO SEND (CIRCUIT CB)
010500  017244                          .WORD   EVMDSR          ;DATA SET READY (CIRCUIT CC)
010502  017250                          .WORD   EVMDCD          ;DATA CARRIER DETECT (CIRCUIT CF)
010504  017254                          .WORD   EVMRTS          ;REQUEST TO SEND (CIRCUIT CA)
010506  017260                          .WORD   EVMRI           ;RING INDICATOR (CIRCUIT CE)
010510  017264                          .WORD   EVMSQD          ;SIGNAL QUALITY DETECT (CIRCUIT CG)
010512  017270                          .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  015625                  EVTLST: .WORD   EDTXQ    ;TRANSMIT MESSAGE QUEUED
010516  015651                          .WORD   EDTXC    ;TRANSMIT OF MESSAGE COMPLETE
010520  015700                          .WORD   EDRXQ    ;RECEIVE MESSAGE SPACE QUEUED
010522  015725                          .WORD   EDRXC    ;MESSAGE RECEIVED - RECEIVE COMPLETE
010524  015753                          .WORD   EDDER    ;DEVICE INFORMATION
010526  016020                          .WORD   EDDVI    ;DEVICE INITIALIZE STARTED
010530  015770                          .WORD   EDDCK    ;DATA COMPARISON DONE
010532  014635                          .WORD   LPO      ;NULL STRING
010534  016046                          .WORD   EDDLE    ;DATA COMPARE LENGTH ERROR
010536  016103                          .WORD   EDDDE    ;DATA COMPARE DATA ERROR
010540  016136                          .WORD   EDEOP    ;END OF PASS

                                ;;;FOLLOWING TABLE USED IN DOWNLINE LOAD ROUTINE.
                                ;;;;CONTAINS POINTERS TO ASCIZ DEVICE DESCRIPTIONS
010542  020217                  DLLIND: .WORD   DPM
010544  020222                          .WORD   DUM
010546  020225                          .WORD   DLM
010550  020230                          .WORD   DQM
010552  020233                          .WORD   DAM
010554  020236                          .WORD   DUPM
010556  020242                          .WORD   DMCM
010560  020246                          .WORD   DNM
010562  020251                          .WORD   DLVM
```

```
5093  010564  020255                    .WORD   DMPM
5094  010566  020261                    .WORD   DTEM
5095  010570  020265                    .WORD   DVM
5096  010572  020270                    .WORD   DZM
5097  010574  020273                    .WORD   UNKM
5098  010576  020303                    .WORD   KDPM
5099  010600  020307                    .WORD   KDZM
5100  010602  020313                    .WORD   KLM
5101  010604  020316                    .WORD   DMVM
5102
5103                              ;LOCATIONS USED DURING EVENT REPORTING
5104
5105  010606  000000              EVTSEC: .WORD   0       ;TEMPORARY LOCS TO KEEP EVENT TIME WHILE REPORTING
5106  010610  000000              EVTMIN: .WORD   0
5107  010612  000000              EVTTCK: .WORD   0
5108  010614  000000              EVTADD: .WORD   0       ;TEMP. LOC. TO HOLD ADDRESS DURING EVENT REPORTING
5109  010616  000000              EVTBCT: .WORD   0       ;       ''      ''  BYTE COUNT   ''      ''      ''
5110  010620  000000              EVTTMP: .WORD   0       ;       ''      ''  OTHER DATA   ''      ''      ''
5111
5112                              ;REPORT CODING DISPATCH TABLE
5113
5114  010622  031014              RPTDSP: .WORD   RPTTXQ  ;TRANSMIT QUEUED ENTRY DECODING
5115  010624  031014                      .WORD   RPTTXQ  ;TRANSMIT COMPLETE ENTRY DECODING
5116  010626  031014                      .WORD   RPTTXQ  ;RECEIVER QUEUED ENTRY DECODING
5117  010630  031014                      .WORD   RPTTXQ  ;RECEIVER COMPLETE ENTRY DECODING
5118  010632  031066                      .WORD   RPTDER  ;DEVICE ERROR ENTRY DECODING
5119  010634  031162                      .WORD   RPTDVI  ;DEVICE INIT ENTRY DECODING
5120  010636  031356                      .WORD   RPTDCK  ;DATA COMPARISON ENTRY DECODING
5121  010640  030642                      .WORD   RPT     ;PLACE HOLDER
5122  010642  031356                      .WORD   RPTDLE  ;DATA COMPARISON LENGH ERROR
5123  010644  031302                      .WORD   RPTDDE  ;DATA COMPARISON DATA ERROR
5124  010646  031226                      .WORD   RPTEOP  ;END OF PASS
5125
5126
5127  010650  000000              DEV1:   .WORD   0               ;TEMP LOCS TO HOLD DATA FOR EVENT REPORTING
5128  010652  000000              DEV2:   .WORD   0               ;  AND SHOW MODE,... SUBROUTINE
5129  010654  000000              DEV3:   .WORD   0
5130  010656  000000              DEV4:   .WORD   0
5131
```

```
                        .SBTTL          COMMAND LINE ACTION TREE

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


010660                  CLITRE:

                        ;FIRST KEYWORD
010660                          CLI     CLISPA,0,N10$                   ;SKIP ANY LEADIN SPACES
010664                  N10$:   CLI     <'?>,HLP,N42$                   ;IS THE FIRST NON-SP CHAR A '?'
010670                          CLI     CLIEXI,0                        ;  IF YES DO 'HLP' AND EXIT
010672                  N42$:   CLI     CLISTR,HLP,N43$,<'HELP'>        ;ELSE, IS FIRST WORD A 'HELP'
010706                          CLI     CLIEXI,0                        ;  IF YES DO 'HLP' AND EXIT
010710                  N43$:   CLI     CLISTR,PRNT,N44$,<'PRINT'>      ;ELSE, IS FIRST WORD A 'PRINT'
010724                          CLI     CLIEXI,0                        ;  IF YES DO 'PRINT' AND EXIT
010726                  N44$:   CLI     CLISTR,EXIT,N45$,<'EXIT'>       ;ELSE,IS FIRST WORD AN 'EXIT';REV B BY E
010742                          CLI     CLIEXI,0                        ;  IF YES DO 'EXIT' AND EXIT
010744                  N45$:   CLI     CLISTR,RUN,N46$,<'RUN'>         ;ELSE, IS FIRST WORD A 'RUN'
010756                          CLI     CLIBR,0,N80$                    ;  IF YES DO 'RUN' & GOTO N80$
010762                  N46$:   CLI     CLISTR,NOTNUF,N40$,<'DUMP'>     ;ELSE, IS FIRST WORD A 'DUMP'
010776                          CLI     CLIBR,0,N50$                    ;  IF YES GOTO N80$
011002                  N40$:   CLI     CLISTR,CLEAR,N20$,<'CLEAR'>     ;ELSE, IS FIRST WORD A 'CLEAR'
011016                          CLI     CLIBR,NOTNUF,N100$              ;  IF YES DO 'CLR' & GOTO N100$
011022                  N20$:   CLI     <'S>,NOTNUF,N30$                ;ELSE, IS FIRST CHAR. A 'S'
011026                          CLI     CLISTR,SHOW,N25$,<'HOW'>        ;  IF YES IS REST OF WORD 'HOW'
011040                          CLI     CLIBR,0,N100$                   ;    IF YES, DO 'SHOW',BR N100$
011044                  N25$:   CLI     CLISTR,0,N30$,<'ET'>            ;  ELSE, IS REST OF WORD 'ET'
011056                          CLI     CLIBR,0,N110$                   ;    IF YES, DO 'SET', BR N110$
011062                  N30$:   CLI     CLIERR,0                        ;OTHERWISE 'ILL CMD' - EXIT

                        ;SECOND KEYWORD (MODE=) FOR RUN COMMAND

011064                  N80$:   CLI     CLISPA,0,N30$                   ;SKIP LEADING SPS, IF NONE-ERR
011070                  N81$:   CLI     CLISTR,NOTNUF,N30$,<'MODE'>     ;IS NEXT WORD 'MODE='
011104                          CLI     <'=>.0,N30$                     ;  IF NO, IT'S WRONG -ERR -EXIT
011110                          CLI     CLISTR,ATVMOD,N82$,<'ACTIVE'>   ;IS NEXT WORD 'ACTIVE'
011126                          CLI     CLIBR,0,N115$                   ;  IF YES, DO 'ACTIVE',BR N115$
011132                  N82$:   CLI     CLISTR,PASMOD,N83$,<'PASSIVE'>  ;IS NEXT WORD 'PASSIVE'
011150                          CLI     CLIBR,0,N115$                   ;  IF YES, DO 'PASSVE',BR N115$
011154                  N83$:   CLI     CLISTR,RECMOD,N84$,<'RECEIVE'>  ;IS NEXT WORD 'RECEIVE'
011172                          CLI     CLIBR,0,N115$                   ;  IF YES, DO 'RECVE',BR N115$
011176                  N84$:   CLI     CLISTR,LISMOD,N85$,<'LISTEN'>   ;IS NEXT WORD 'LISTEN'
011214                          CLI     CLIBR,0,N115$                   ;  IF YES, DO 'LISTEN',BR N115$
011220                  N85$:   CLI     CLISTR,DLLMOD,N86$,<'DOWNLINELOAD'>     ;IS NEXT WORD 'DOW...'
011244                          CLI     CLIBR,0,N115$                   ;  IF YES, DO 'DWNLL',BR N115$
011250                  N86$:   CLI     <'T>,0,N30$                     ;IS NEXT CHAR A 'T'
```

```
 5189  011254                          CLI     CLISTR,TRAMOD,N87$,<'RANSMIT'>  ; IS REST OF WORD 'RANSMIT''
 5190  011272                          CLI     CLIBR,0,N115$                   ;  IF YES, DO 'TRANSM'',BR N115$
 5191  011276                N87$:     CLI     CLISTR,TALMOD,N30$,<'ALK'>      ; IS REST OF WORD ''ALK''
 5192  011310                          CLI     CLIBR,0,N115$                   ;  IF YES, DO ''TALK'',BR N115$
 5193                                                                          ; IF NO, ERROR - EXIT
 5194
 5195                                ;SECOND KEYWORD (FOR CLEAR OR SHOW)
 5196  011314                N100$:    CLI     CLISPA,0,N30$                   ;SKIP LEADING SPACES, NONE=ERR
 5197  011320                N102$:    CLI     CLISTR,CSHEXP,N104$,<'EXPECT'>  ;IS NEXT WORD 'EXPE...''
 5198  011336                          CLI     CLIEXI,0                        ; IF YES, DO CLR-EXP,EXIT
 5199  011340                N104$:    CLI     CLISTR,CSHTRN,N30$,<'TRANSMIT'> ;IS NEXT WORD ''TRANS..''
 5200  011360                          CLI     CLIEXI,0                        ; IF YES, DO CLR-TRN,EXIT
 5201                                                                          ;IF NO - ERROR - EXIT
 5202
 5203
 5204                                ;SECOND KEYWORD (FOR SET)
 5205  011362                N110$:    CLI     CLISPA,0,N30$
 5206  011366                N111$:    CLI     CLISTR,SETEXP,N112$,<'EXPECT'>
 5207  011404                          CLI     CLIBR,0,N120$
 5208  011410                N112$:    CLI     CLISTR,SETTRN,N30$,<'TRANSMIT'>
 5209  011430                          CLI     CLIBR,0,N120$
 5210
 5211                                ;GET ADDRESSES FOR DUMP COMMAND
 5212  011434                N50$:     CLI     CLIALP,0,N51$
 5213  011440                N51$:     CLI     CLISPA,0,N52$
 5214  011444                N52$:     CLI     CLIOCT,DMPS,N30$
 5215  011450                          CLI     <'->,NOTNUF,N125$
 5216  011454                          CLI     CLIOCT,DMPE,N30$
 5217  011460                          CLI     <'/>,NOTNUF,N125$
 5218  011464                          CLI     <'B>,DMPQ,N30$
 5219  011470                          CLI     CLIBR,0,N125$
 5220
 5221                                ;QUALIFIERS FOR THE RUN COMMAND
 5222  011474                N115$:    CLI     CLIALP,0,N114$
 5223  011500                N114$:    CLI     <'/>,NOTNUF,N125$
 5224  011504                          CLI     CLISTR,NO,N116$,<'NO'>
 5225  011516                N116$:    CLI     <'C>,0,N117$
 5226  011522                          CLI     CLISTR,CHECK,N117$,<'HECK'>
 5227  011536                          CLI     CLIBR,0,N115$
 5228
 5234
 5235                                ;N113$: CLI     CLISTR,CRC,N30$,<'RC16'>
 5236                                ;       CLI     CLIBR,0,N115$
 5237
 5238  011542                N117$:    CLI     CLISTR,STATUS,N118$,<'STATUS'>
 5239  011560                          CLI     CLIBR,0,N115$
 5240  011564                N118$:    CLI     CLISTR,ECHO,N130$,<'ECHO'>
 5241  011600                          CLI     CLIBR,0,N115$
 5242
 5255
 5256  011604                N130$:    CLI     CLISTR,0,N131$,<'PASS'>
 5257  011620                          CLI     CLIBR,0,N150$
 5258  011624                N131$:    CLI     CLISTR,0,N132$,<'LOOP'>
 5259  011640                          CLI     CLIBR,0,N140$
 5260
 5261  011644                N132$:    CLI     CLISTR,MOSC,N30$,<'MODEM'>      ;MODEM ACTION
```

```
 5262  011660                          CLI     CLIBR,0,N115$                   ;;ADDED BY EC
 5263
 5264                          ;GET MESSAGE TYPE FOR SET MESSAGE COMMANDS
 5265  011664                  N120$:  CLI     <'=>,0,N30$
 5266
 5267                          ;   LOOK FOR DEFAULT MESSAGE NAME
 5268  011670                  N60$:   CLI     CLISTR,CMSG1,N61$,<'ONES'>
 5269  011704                          CLI     CLIBR,0,N121$
 5270  011710                  N61$:   CLI     CLISTR,CMSG0,N62$,<'ZEROES'>
 5271  011726                          CLI     CLIBR,0,N121$
 5272  011732                  N62$:   CLI     CLISTR,CMSG2,N63$,<'1ALT'>
 5273  011746                          CLI     CLIBR,0,N121$
 5274  011752                  N63$:   CLI     CLISTR,CMSG3,N64$,<'0ALT'>
 5275  011766                          CLI     CLIBR,0,N121$
 5276  011772                  N64$:   CLI     CLISTR,CMSG5,N65$,<'ITEP'>
 5277  012006                          CLI     CLIBR,0,N121$
 5278  012012                  N65$:   CLI     CLISTR,CMSG4,N66$,<'CCITT'>
 5279  012026                          CLI     CLIBR,0,N121$
 5280  012032                  N66$:   CLI     CLISTR,CMSG6,N67$,<'ALPHA'>
 5281  012046                          CLI     CLIBR,0,N121$
 5282  012052                  N67$:   CLI     CLISTR,SETET,N68$,<'TRANSMIT'>  ;REV B BY EC
 5283  012072                          CLI     CLIBR,0,N125$                   ;REV B BY EC
 5284
 5285                          ;   LOOK FOR QUOTED MESSAGE
 5286  012076                  N68$:   CLI     <''>,OPRMSG,N30$
 5287  012102                  N70$:   CLI     <''>,ENDQO,N71$
 5288  012106                          CLI     CLIBR,0,N121$
 5289  012112                  N71$:   CLI     CLISPA,0,N72$
 5290  012116                  N72$:   CLI     CLIALN,0,N73$           ;ONLY A-Z,SP,TAB, OR 0-9 BETWEEN '"S
 5291  012122                          CLI     CLIBR,0,N70$
 5292  012126                  N73$:   CLI     CLIERR,BADCHR           ;PRINT ERROR IF NONE LEGAL CHAR FOR '"S
 5293
 5294                          ;GET QUALIFIERS (SIZE OR COPY) FOR SET MESSAGE COMMANDS
 5295  012130                  N121$:  CLI     CLIALP,0,N123$
 5296  012134                  N123$:  CLI     <'/>,NOTNUF,N125$
 5297  012140                          CLI     CLISTR,SIZE,N122$,<'SIZE'>
 5298  012154                          CLI     CLIBR,0,N126$
 5299  012160                  N122$:  CLI     CLISTR,QCOPY,N30$,<'COPY'>
 5300  012174                          CLI     CLIBR,0,N126$
 5301
 5302                          ;NUMER FOR SIZE OR COPY
 5303  012200                  N126$:  CLI     <'=>,0,N30$
 5304  012204                          CLI     CLIDEC,NUM,N30$
 5305  012210                          CLI     CLIBR,0,N121$
 5306
 5307                          ;GET MAINTENANCE LOOP TYPE FOR RUN 'LOOP'' QUIALIFIER
 5308  012214                  N140$:  CLI     <'=>,0,N30$
 5309
 5318
 5319  012220                  N141$:  CLI     CLISTR,TTLLOP,N142$,<'INTERNALTTL'>
 5320  012242                          CLI     CLIBR,0,N115$
 5321  012246                  N142$:  CLI     CLISTR,CBLLOP,N143$,<'CABLE'>
 5322  012262                          CLI     CLIBR,0,N115$
 5323  012266                  N143$:  CLI     CLISTR,LMDLOP,N144$,<'LOCALMODEM'>
 5324  012310                          CLI     CLIBR,0,N115$
 5325  012314                  N144$:  CLI     CLISTR,RMDLOP,N30$,<'REMOTEMODEM'>
```

```
 5326  012336                          CLI     CLIBR,0,N115$
 5327
 5328                          ;GET LINE NUMBER FOR 'PASS'' RUN QUALIFIER
 5329  012342                  N150$:  CLI     <'=>,0,N30$
 5330  012346                          CLI     CLIDEC,PASC,N30$
 5331  012352                          CLI     CLIBR,0,N115$
 5332
 5333
 5334
 5335                          ;END-OF-LINE
 5336  012356                  N125$:  CLI     CLIEXI,0
 5337
```

```
5349
5350
5351                                    ;DEVICE DEPENDENT STORAGE LOCATIONS FOR
5352                                    ;  CURRENT DEVICE PARAMTERS
5353
5354  012360                            SEL0:
5355  012360  000000                    BSEL0:  .WORD   0               ;ADDRESSES OF REGISTERS SEL0 THRU BSEL7
5356  012362  000000                    BSEL1:  .WORD   0
5357  012364                            SEL2:
5358  012364  000000                    BSEL2:  .WORD   0
5359  012366  000000                    BSEL3:  .WORD   0
5360  012370                            SEL4:
5361  012370  000000                    BSEL4:  .WORD   0
5362  012372  000000                    BSEL5:  .WORD   0
5363  012374                            SEL6:
5364  012374  000000                    BSEL6:  .WORD   0
5365  012376  000000                    BSEL7:  .WORD   0
5366
5367
5368  012400  000000                    INVEC:  .WORD   0               ;INPUT INTERRUPT VECTOR ADDRESS
5369  012402  000000                    OUTVEC: .WORD   0               ;OUTPUT INTERRUPT VECTOR ADDRESS
5370  012404  000000                    INTPRI: .WORD   0               ;INTERRUPT PRIORITY
5371  012406  000000                    OPTYP:  .WORD   0               ;DEVICE OPTION TYPE(0=DMC,5=DMR-DMC MODE
5372                                                                    ;7=DMR).
5373
5374
5387
5397
5398                                    ;       ERRTBL
```

```
5400                                    .SBTTL  GLOBAL TEXT SECTION
5401
5402                                    ;++
5403                                    ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
5404                                    ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
5405                                    ; MORE THAN ONE TEST.
5406                                    ;--
5407
5408                                    .SBTTL          DEVICE SUPPORTED
5409                                    ;
5410                                    ; NAMES OF DEVICES SUPPORTED BY PROGRAM
5411                                    ;
5412  012410                                    DEVTYP  <DMR,DMC-11>
 (4)  012410                                                                    L$DVTYP::
 (3)  012410  046504  026122  046504                                                .ASCIZ  /DMR,DMC-11/
 (3)  012416  026503  030461     000
 (2)          012424                                                            .EVEN
5413
5419
5428
5429                                    .SBTTL          PROGRAM IDENTIFICATION
5430                                    ; TEST DESCRIPTION
5431                                    ;
5432  012424                                    DESCRIPT        <CZCLKB0 DMR, DMC-11 DATA COMM. LINK TEST>
 (4)  012424                                                                    L$DESC::
 (3)  012424  055103  046103  041113                                                .ASCIZ  /CZCLKB0 DMR, DM
 (3)  012432  020060  046504  026122
 (3)  012440  042040  041515  030455
 (3)  012446  020061  040504  040524
 (3)  012454  041440  046517  027115
 (3)  012462  046040  047111  020113
 (3)  012470  042524  052123     000
 (2)          012476                                                            .EVEN
5433                                                .EVEN
5434
5441
5442
5443
```

```
 5445                                   .SBTTL          GLOBAL FORMAT STATEMENTS, MESSAGES, AND ASCII INFO
 5449
 5450  012476 041504 052114 000076 CLI$PM: .ASCIZ  /DCLT>/
 5451  012504 050122 037124 000040 CLI$RP: .ASCIZ  /RPT> /                    ;REV B BY EC
 5452  012512 047045 040445 044477 CLIERM: .ASCIZ  /%N%A?ILL CMD-BAD SYNTX?/
 5453  012542 047045 040445 044477 CLINUF: .ASCIZ  /%N%A?INCMPLTE CMD?/
 5454  012565    045 022516 037501 CLINBG: .ASCIZ  /%N%A?NUM TOO BIG?/
 5455  012607    045 022516 037501 CLIBRX: .ASCIZ  /%N%A?BAD RADIX?/
 5456  012627    045 022516 037501 CLIBDL: .ASCIZ  /%N%A?''LOOP'' VALID ONLY IN ACTIVE?/
 5457  012671    045 022516 037501 CLINPS: .ASCIZ  /%N%A?''ECHO'' VALID ONLY IN PASSIVE?/
 5458  012734 047045 040445 044477 CLIBCR: .ASCIZ  /%N%A?ILL CHR- ''A-Z,0-9,SP,TAB'' ONLY?/
 5459  013001    045 022516 037501 CLISEO: .ASCIZ  /%N%A?''SIZE=0'' NOT VALID?/
 5460  013032 047045 040445 052077 CLIPW:  .ASCIZ  /%N%A?TRANSMIT & EXPECT LIST MUST BE IDENTICAL FOR LOOP?/ ;REV B BY EC
 5461  013122 040523 052124 046105 DLLQ1:  .ASCIZ  /SATTELITE PASSWORD= /          ;REV B BY EC
 5462  013147    045 022516 052101 HLP0:  .ASCIZ  /%N%ATHIS IS DCLT. TYPE ''H'' OR ''?'' FOR DETAILS/
 5463  013225    045 022516 000124 HLPF:  .ASCIZ  /%N%T/
 5464  013232 041504 052114 041440 HLP1:  .ASCIZ  /DCLT CMDS:/
 5465  013245    040 046103 040505 HLP2:  .ASCII  / CLEAR OR SHOW  EXPECTLIST OR TRANSMITLIST/<15><12>
 5466  013321    040 051120 047111         .ASCII  / PRINT/<15><12>
 5467  013331    040 054105 052111         .ASCII  / EXIT/<15><12>  ;REV B BY EC
 5468  013340 042040 046525 020120         .ASCIZ  ? DUMP START-END/B?
 5469  013362 051440 052105 042440 HLP3:  .ASCIZ  ? SET EXPECTMSG OR TRANSMITMSG=TYPE/SIZE=N OR /COPY=N?
 5470  013447    040 042523 020124 HLP3A: .ASCIZ  / SET EXPECT=TRANSMIT/  ;REV B BY EC
 5471  013474 020040 052040 050131 HLP4:  .ASCIZ  ?   TYPE=ONES,ZEROES,1ALT,0ALT,ITEP,CCITT,ALPHA?
 5472  013553    040 020040 020040 HLP4A: .ASCIZ  /       OR 'OPR SPCD=A-Z,SP,TAB,0-9 IN QUOTES''/
 5473  013631    040 052522 020116 HLP5:  .ASCIZ  ? RUN MODE=MTYP/LOOP=LTYP/CHECK,STATUS,ECHO,MODEM,PASS=N?
 5474  013721    040 020040 052115 HLP6:  .ASCII  /   MTYP=TRAN,REC,ACT,PAS,TAL,LIS,DOWN/<15><12>
 5475  013770 020040 046040 054524         .ASCIZ  /   LTYP=INT,CAB,LOC,REM/
 5476  014020 047045 040445 054524 RHLP0:  .ASCIZ  /%N%ATYPE 'H' OR '?'' FOR HELP !/          ;REV B BY EC
 5477  014057    104 046103 020124 RHLP1:  .ASCIZ  /DCLT REPORT CMDS :/              ;REV B BY EC
 5478  014102 047514 020107 020055 RHLP2:  .ASCIZ  /LOG - PRINT DCLT EVENT LOG/              ;REV B BY EC
 5479  014135    105 044530 020124 RHLP3:  .ASCIZ  /EXIT - EXIT REPORT LEVEL/       ;REV B BY EC
 5480  014166 042510 050114 026440 RHLP4:  .ASCIZ  /HELP - PRINT THIS MESSAGE/      ;REV B BY EC
 5481  014220 040502 042523 042457 RHLP5:  .ASCIZ  !BASE/ERROR - PRINT ONLY ERRORS!          ;REV B BY EC
 5482  014257    102 051501 027505 RHLP6:  .ASCIZ  !BASE/FULL - PRINT ENTIRE TABLE!          ;REV B BY EC
 5483  014316 040502 042523 047457 RHLP7:  .ASCIZ  !BASE/OFFSET=NNN - PRINT SINGLE LOCATION!<15><12> ;REV B BY EC
 5484  014370 047045 040445 040502 RPTIV:  .ASCIZ  /%N%ABASE OFFSET=%03%A TOO BIG !/          ;REV B BY EC
 5485  014430 047045 040445 051515 SHMSG:  .ASCIZ  ?%N%AMSG: TYPE=%T%A/SIZE=%D3?
 5486  014464 042532 047522 051505 SHTYP0: .ASCIZ  /ZEROES/
 5487  014473    117 042516 000123 SHTYP1: .ASCIZ  /ONES/
 5488  014500 040461 052114    000 SHTYP2: .ASCIZ  /1ALT/
 5489  014505    060 046101 000124 SHTYP3: .ASCIZ  /0ALT/
 5490  014512 041503 052111 000124 SHTYP4: .ASCIZ  /CCITT/
 5491  014520 052111 050105    000 SHTYP5: .ASCIZ  /ITEP/
 5492  014525    101 050114 040510 SHTYP6: .ASCIZ  /ALPHA/
 5493  014533    117 051120 051440 SHTYP7: .ASCIZ  /OPR SPEC/
 5494  014544 042522 042503 053111 MO0:    .ASCIZ  /RECEIVE/
 5495  014554 051124 047101 046523 MO1:    .ASCIZ  /TRANSMIT/
 5496  014565    120 051501 044523 MO2:    .ASCIZ  /PASSIVE/
 5497  014575    101 052103 053111 MO3:    .ASCIZ  /ACTIVE/
 5498  014604 047504 047127 044514 MO4:    .ASCIZ  /DOWNLINELOAD/
 5499  014621    124 046101 000113 MO5:    .ASCIZ  /TALK/
 5500  014626 044514 052123 047105 MO6:    .ASCIZ  /LISTEN/
 5501  014635    000             LP0:    .ASCIZ  //
 5502  014636 046057 047517 036520 LP00:   .ASCIZ  ?/LOOP=?
 5503  014645    111 052116 051105 LP1:    .ASCIZ  ?INTERNAL?
```

```
5504  014656  040503  046102  000105  LP2:    .ASCIZ  ?CABLE?
5505  014664  047514  040503  046514  LP3:    .ASCIZ  ?LOCALMODEM?
5506  014677     122  046505  052117  LP4:    .ASCIZ  'REMOTEMODEM?
5507  014713     116     117          PNST:   .ASCII  /NO/
5508  014715     123  040524  052524  PST:    .ASCIZ  /STATUS/
5509  014724  047516                  PNCK:   .ASCII  /NO/
5510  014726  044103  041505  000113  PCK:    .ASCIZ  /CHECK/
5511  014734  047516                  PNEC:   .ASCII  /NO/
5512  014736  041505  047510     000  PEC:    .ASCIZ  /ECHO/
5513
5524  014743     116     117          PNMS:   .ASCII  /NO/                    ;ADDED BY EC
5525  014745     115  042117  046505  PMS:    .ASCIZ  /MODEM/                 ;ADDED BY EC
5526  014753     045  022516  046101  LISP:   .ASCIZ  /%N%ALIS>/
5527  014764  046124  037113     000  OPRMM:  .ASCIZ  /TLK>/
5528  014771     124  044510  020123  L5060:  .ASCIZ  /THIS A 50. OR 60. HZ. LSI-11:/
5529          015030                  .EVEN
5530
5531
5532
5533                                  ;
5534                                  ; FORMAT STATEMENTS USED IN PRINT CALLS
5535                                  ;
5536
5537  015030  047045  040445  047504  DLLCM:  .ASCIZ  /%N%ADOWN LINE LOAD COMPLETED SUCCESSFULLY/
5538  015102  047045  040445  040502  NOCLK:  .ASCIZ  /%N%ABAD CLOCK - PROGRAM WILL HANG ON 'TIMEOUT''!!/
5539  015163     115  054101  020056  TABEX:  .ASCIZ  /MAX. CHAR. MSG COUNT EXCEEDED -/
5540  015223     102  043125  042506  BUFEX:  .ASCIZ  /BUFFER FULL -/
5541  015241     045  022516  022524  MSGTRN: .ASCIZ  /%N%T%A MSG. NOT BUILT !!/
5542  015272  047045  040445  044103  MSGTRU: .ASCIZ  /%N%ACHAR. COUNT EXCEEDS BUFF LIMIT - MSG TRUNCATED/
5543  015355     045  022516  032523  SHF0:   .ASCIZ  ?%N%S5%AMODE=%T%T%T%A/PASS=%Z5?
5544
5550
5551  015413     045  022516  032523  SHF1:   .ASCIZ  ?%N%S5%S5%S5%A/%T%A/%T%A/%T%A/%T?
5552
5553  015453     045  032523  040445  EFM2:   .ASCIZ  /%S5%ATOTAL MISMATCHES IN MSG = %D5/
5554  015516  047045  051445  022463  PCPM:   .ASCIZ  /%N%S3%ACALLED FROM PC=%O6/
5555  015550  051445  022465  041501  EFM11:  .ASCIZ  /%S5%ACOMPARE COUNT=%D5%S3%ARECEIVE COUNT=%D5/
5556
5557
5558                                  ;EVENT DESCRIPTION MESSAGES
5559
5560  015625     124  040522  051516  EDTXQ:  .ASCIZ  /TRANSMIT MSG QUEUED/
5561  015651     124  040522  051516  EDTXC:  .ASCIZ  /TRANSMIT MSG COMPLETED/
5562  015700  042522  042503  053111  EDRXQ:  .ASCIZ  /RECEIVE SPACE QUEUED/
5563  015725     122  041505  044505  EDRXC:  .ASCIZ  /RECEIVE MSG COMPLETED/
5564  015753     104  053105  041511  EDDER:  .ASCIZ  /DEVICE ERROR/
5565  015770  040504  040524  041440  EDDCK:  .ASCIZ  /DATA COMPARISON STARTED/
5566  016020  042504  044526  042503  EDDVI:  .ASCIZ  /DEVICE INIT AND SETUP/
5567  016046  040504  040524  041440  EDDLE:  .ASCIZ  /DATA COMPARISON LENGTH ERROR/
5568  016103     104  052101  020101  EDDDE:  .ASCIZ  /DATA COMPARISON DATA ERROR/
5569  016136  047105  020104  043117  EDEOP:  .ASCIZ  /END OF PASS/
5570  016152  041101  047516  046522  EDMOS:  .ASCIZ  /ABNORMAL MODEM STATUS CHANGE/          ;ADDED BY EC
5571
5572                                  ;*****************************************************************
5573                                  ;THESE TWO STORAGE AREAS MUST NOT BE SEPERATED !!!!
5574
```

```
                                        ;EVENT REPORTING MESSAGES
016207      040  041040  051501  BASM1A: .ASCIZ  /  BASE TABLE/

016224  051445  022463  031517  BASM3:  .ASCIZ  /%S3%03/
016233      045  031523  047445  BASM2:  .ASCIZ  /%S3%06/
016242  047045  047445  000066  BASM1:  .ASCIZ  /%N%06/


016250  047045  040445  044124  NULEVT: .ASCIZ  ?%N%ATHE DCLT EVENT LOG IS EMPTY?
016310  047045  040445  037076  EVTF0:  .ASCIZ  ?%N%A>>>DCLT EVENT LOG ENTRY <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<?
016403      045  022516  032504  EVTF1:  .ASCIZ  /%N%D5%A:%Z2%A:%Z2%S3%T/
016432  047045  051445  022463  EVTF2:  .ASCIZ  /%N%S3%AADDR OF MSG=%06%S3%ABYTE COUNT=%D5/
016504  047045  051445  022463  EVTF3:  .ASCIZ  /%N%S3%T%N/
016516  051445  022463  033117  EVTF3C: .ASCIZ  /%S3%06%S3%06/
016533      045  031523  047445  EVTF3D: .ASCIZ  /%S3%06%S3%06%S3%T/
016555      045  022516  031523  EVTF4:  .ASCIZ  /%N%S3%AADDR OF MSG=%06%S3%ABYTE COUNT=%D5%S3%ANO. OF CMP ERRS=%D5/
016657      045  022516  031523  EVTF4A: .ASCIZ  /%N%S3%AADDR OF MSG=%06%S3%ARX BYTES=%D5%S3%ACOMPARE BYTES=%D5/
016755      045  022516  031523  EVTF4B: .ASCIZ  /%N%S3%APASS=%D5%S3%AERRORS=%D5%S3%ANOBUFFS=%D5/
017034  051445  022465  041101  EVTF5A: .ASCIZ  /%S5%ABYTE # IN MSG.=%D5%S3%AEXPTD=%03%S3%ARECVD=%03/

017120  047045  051445  022471  EVMOCG: .ASCIZ  /%N%S9%ACHANGED TO:/

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

017143      045  022516  034123  EVMOHD: .ASCIZ  /%N%S8%AMODEM STATUS: CTS DSR DCD RTS RI  SQD TM/
017223      045  022516  034523  EVMOST: .ASCII  /%N%S9%S9%S5%A/
017240      130     040     040  EVMCTS: .BYTE   'X,40,40,40
017244      130     040     040  EVMDSR: .BYTE   'X,40,40,40
017250      130     040     040  EVMDCD: .BYTE   'X,40,40,40
017254      130     040     040  EVMRTS: .BYTE   'X,40,40,40
017260      130     040     040  EVMRI:  .BYTE   'X,40,40,40
017264      130     040     040  EVMSQD: .BYTE   'X,40,40,40
017270      130     040     040  EVMTM:  .BYTE   'X,40,40,40
017274      000                  .BYTE   0
017276                           .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
```

```
5670                                    ;REV B BY EC
5671                                    .SBTTL  DMR BASE TABLE DESCRIPTION MESSAGES
5672  020322  047125  042504  044506  DMUNKN: .ASCIZ  /UNDEFINED DATA /      ;LOCATION UNDEFINED BY SPEC
5673  020342  040502  042523  052040  DMR000: .ASCIZ  /BASE TABLE UPDATE INDEX POINTER/
5674  020402  040502  042523  052040  DMR001: .ASCIZ  /BASE TABLE UPDATE LIMIT/
5675  020432  042502  044507  047116  DMR002: .ASCIZ  /BEGINNING OF BASE TABLE DATA/
5676  020467     116  045501  020123  DMR003: .ASCIZ  /NAKS RCVD..BUFFER TEMP UNAVAILABLE/
5677  020532  040516  051513  051040  DMR004: .ASCIZ  /NAKS RCVD..HEADER BCC ERROR/
5678  020566  040516  051513  051040  DMR005: .ASCIZ  /NAKS RCVD..DATA BCC ERROR/
5679  020620  040516  051513  051440  DMR006: .ASCIZ  /NAKS SENT..BUFFER TEMP UNAVAILABLE/
5680  020663     116  045501  020123  DMR007: .ASCIZ  /NAKS SENT..HEADER BCC ERROR/
5681  020717     116  045501  020123  DMR010: .ASCIZ  /NAKS SENT..DATA BCC ERROR/
5682  020751     122  050105  020123  DMR011: .ASCIZ  /REPS SENT..CUMUL REP SENT/
5683  021003     122  050105  020123  DMR012: .ASCIZ  /REPS RCVD..CUMUL REP RCVD/
5684  021035     116  045501  020123  DMR013: .ASCIZ  /NAKS RCVD..REP RESPONSE/
5685  021065     116  045501  020123  DMR014: .ASCIZ  /NAKS RCVD..RCV OVERRUN/
5686  021114  040516  051513  051040  DMR015: .ASCIZ  /NAKS RCVD..MSG HDR FORMAT/
5687  021146  040516  051513  051040  DMR016: .ASCIZ  /NAKS RCVD..MSG TOO LONG/
5688  021176  040516  051513  051440  DMR017: .ASCIZ  /NAKS SENT..REP RESPONSE/
5689  021226  040516  051513  051440  DMR020: .ASCIZ  /NAKS SENT..RCV OVERRUN/
5690  021255     116  045501  020123  DMR021: .ASCIZ  /NAKS SENT..MSG HDR FORMAT/
5691  021307     130  044515  020124  DMR022: .ASCIZ  /XMIT UNDERRUN COUNT/
5692  021333     103  046101  020114  DMR023: .ASCIZ  /CALL SET UP FAILURE COUNT/
5693  021365     101  052103  020123  DMR024: .ASCIZ  /ACTS FAILURE COUNT/
5694  021410  040503  051122  042511  DMR025: .ASCIZ  /CARRIER DETECT LOST COUNT(WHILE RECEIVING)/
5695  021463     122  041505  044505  DMR026:.ASCIZ   /RECEIVER INACTIVE COUNT/
5696  021513     123  051124  040505  DMR027: .ASCIZ  /STREAMING TIME-OUT COUNT/
5697  021544  046530  054502  024124  DMR030: .ASCIZ  /XMBYT(LSB) - TOTAL # BYTES XMITTFD, 32 BIT COUNTER/
5698  021627     130  041115  052131  DMR031: .ASCIZ  ?XMBYT(2/4) - # BYTES XMITTED?
5699  021664  046530  054502  024124  DMR032: .ASCIZ  ?XMBYT(3/4) - # BYTES XMITTED?
5700  021721     130  041115  052131  DMR033: .ASCIZ  ?XMBYT(MSB) - # BYTES XMITTED?
5701  021756  041522  054502  024124  DMR034: .ASCIZ  /RCBYT(LSB) - TOTAL # BYTES RECEIVED, 32 BIT COUNTER/
5702  022042  041522  054502  024124  DMR035: .ASCIZ  ?RCBYT(2/4) - # BYTES RECEIVED (CONT)?
5703  022107     122  041103  052131  DMR036: .ASCIZ  ?RCBYT(3/4) - # BYTES RECEIVED (CONT)?
5704  022154  041522  054502  024124  DMR037: .ASCIZ  /RCBYT(MSB) - # BYTES RECEIVED/
5705  022212  047111  047503  050115  DMR040: .ASCIZ  /INCOMPLETE SELECTION COUNT/
5706  022245     116  020117  042522  DMR041: .ASCIZ  /NO REPLY TO SELECTION COUNTER/
5707  022303     110  043511  042510  DMR042: .ASCIZ  /HIGHEST MESSAGE SUCCESSFULLY RECEIVED/
5708  022351     110  043511  042510  DMR043: .ASCIZ  /HIGHEST MESSAGE TRANSMITTED/
5709  022405     110  043511  042510  DMR044: .ASCIZ  /HIGHEST MESSAGE ACKNOWLEGED/
5710  022441     116  054105  020124  DMR045: .ASCIZ  /NEXT MESSAGE TO TRANSMIT/
5711  022472  040514  052123  046440  DMR046: .ASCIZ  /LAST MESSAGE TO COMPLETE TRANSMISSION/
5712  022540  052503  051122  047105  DMR047: .ASCIZ  /CURRENT MESSAGE BEING TRANSMITTED/
5713  022602  051124  047101  046523  DMR050: .ASCIZ  /TRANSMIT END OF QUEUE/
5714  022630  051124  047101  046523  DMR051: .ASCIZ  /TRANSMIT BEGINNING OF QUEUE/
5715  022664  042522  042503  053111  DMR052: .ASCIZ  /RECEIVE END OF QUEUE/
5716  022711     122  041505  044505  DMR053: .ASCIZ  /RECEIVE BEGINNING OF QUEUE/
5717  022744  040514  042524  052123  DMR054: .ASCIZ  /LATEST NAK REASON/
5718  022766  051120  043517  040522  DMR055: .ASCIZ  ?PROGRAMMABLE REP/SELECT-TIMER PRESET VALUE?
5719  023041     111  052123  052122  DMR056: .ASCIZ  ?ISTRT/ASTRT/REP/SELECT-TIMER COMPARE LEVEL?
5720  023114  041501  044524  042526  DMR057: .ASCIZ  /ACTIVE TIME COUNT/
5721  023136  044124  042522  044123  DMR060: .ASCIZ  /THRESHOLD LEVEL NAKS RCVD/
5722  023170  044124  042522  044123  DMR061: .ASCIZ  /THRESHOLD COUNT NAKS RCVD/
5723  023222  044124  042522  044123  DMR062: .ASCIZ  /THRESHOLD LEVEL NAKS SEND EXCEPT NO BUF/
5724  023272  044124  042522  044123  DMR063: .ASCIZ  /THRESHOLD COUNT NAKS SEND EXCEPT NO BUF/
5725  023342  044124  042522  044123  DMR064: .ASCIZ  /THRESHOLD LEVEL - REPS SENT/
```

```
 5726  023376  044124  042522  044123  DMR065: .ASCIZ  /THRESHOLD COUNT - REPS SENT/
 5727  023432  044124  042522  044123  DMR066: .ASCIZ  /THRESHOLD LEVEL - NO BUF AVAILABLE/
 5728  023475     124  051110  051505  DMR067: .ASCIZ  /THRESHOLD COUNT - NO BUF AVAILABLE/
 5729  023540  042523  020105  046504  DMR177: .ASCIZ  /SEE DMR TECH MANUAL FOR DESCRIPTION/
 5730
 5731
 5732
```

```
 5734                                       ;REV B BY EC
 5735                                       .SBTTL  DMC BASE TABLE DATA DESCRIPTION MESSAGES
 5736  023604  047101  020120  020055  DMC002: .ASCIZ  /ANP - CONSTANT 0/
 5737  023625     116  046124  020122  DMC003: .ASCIZ  /NTLR - NAKS RCVD..NO BUFFERS/
 5738  023662  044116  051104  026440  DMC004: .ASCIZ  /NHDR - NAKS RCVD..MSG HEADER BAD/
 5739  023723     104  052101  020122  DMC005: .ASCIZ  /DATR - NAKS RCVD..DATA BAD/
 5740  023756  052116  051514  026440  DMC006: .ASCIZ  /NTLS - NAKS SENT..NO BUFFERS/
 5741  024013     116  042110  020123  DMC007: .ASCIZ  /NHDS - NAKS SENT..BAD HEADER/
 5742  024050  040504  051524  026440  DMC010: .ASCIZ  /DATS - NAKS SENT..BAD DATA/
 5743  024103     122  050105  051503  DMC011: .ASCIZ  /REPCS - REPS SENT/
 5744  024125     122  050105  051103  DMC012: .ASCIZ  /REPCR - REPS RECD/
 5745  024147     102  051501  020105  DMC013: .ASCIZ  /BASE - CORE TABLE BASE ADDRESS/
 5746  024206  042523  020105  046504  DMC377: .ASCIZ  /SEE DMC TECH MANUAL FOR DESCRIPTION/
```

```
5755
5756                                    ;DEVICE ERROR MESSAGES
5757
5758  024252  044524  042515  047440  DVEM0:  .ASCII  /TIME OUT WAITING FOR RDI TO CLEAR/
5759  024313     015  020012  020040          .ASCIZ  <15><12>/   SEL0     SEL2  /
5760  024340  044524  042515  047440  DVEM1:  .ASCII  /TIME OUT WAITING FOR RDI TO SET/
5761  024377     015  020012  020040          .ASCIZ  <15><12>/   SEL0     SEL2  /
5762  024424  044524  042515  047440  DVEM3:  .ASCII  /TIME OUT WAITING FOR RUN TO SET/
5763  024463     015  020012  020040          .ASCIZ  <15><12>/   SEL0     SEL2  /
5764  024510  044524  042515  047440  DVEM4:  .ASCII  /TIME OUT WAITING FOR OUTPUT INTERRUPT/
5765  024555     015  020012  020040          .ASCIZ  <15><12>/   SEL0     SEL2  /
5766  024602  047111  052520  020124  DVEM5:  .ASCII  /INPUT INTERRUPT WHEN EXPECTING OUTPUT/
5767  024647     015  020012  020040          .ASCIZ  <15><12>/   SEL0     SEL2  /
5768  024674  046111  042514  040507  DVEM6:  .ASCII  /ILLEGAL OUTPUT INTERRUPT/
5769  024724  005015  020040  051440          .ASCIZ  <15><12>/   SEL2     SEL6  /
5770  024751     103  047117  051124  DVEM7:  .ASCII  /CONTROL OUT INSTEAD OF BA-CC OUT/
5771  025011     015  020012  020040          .ASCIZ  <15><12>/   SEL2     SEL6  /
5772
5773  025036  054124  041040  043125  DVEM8:  .ASCII  /TX BUFF COMPLETED AND SHOULD BE RX/
5774  025100  005015  020040  051440          .ASCIZ  <15><12>/   SEL4     SEL6  /
5775  025125     122  020130  052502  DVEM9:  .ASCII  /RX BUFF COMPLETED AND SHOULD BE TX/
5776  025167     015  020012  020040          .ASCIZ  <15><12>/   SEL4     SEL6  /
5777  025214  042040  053517  020116  DLLAB:  .ASCII  / DOWN LINE LOAD ABORTED/
5778  025243     015  020012  020040          .ASCIZ  <15><12>/   RXBUF    TXBUF /
5779
5780  025270  051120  041517  042105  PROEM:  .ASCIZ  /PROCEDURE ERROR/
5781  025310  047516  020116  054105  NXMM:   .ASCIZ  /NON EXIST MEM/
5782  025326  042104  046503  020120  DDCSRM: .ASCIZ  /DDCMP START REC/
5783  025346  044504  041523  047117  DISCOM: .ASCIZ  /DISCONNECT/
5784  025361     114  051517  020124  LOSDAM: .ASCIZ  /LOST DATA/
5785  025373     104  041504  050115  DDCMRM: .ASCIZ  /DDCMP MAINT REC/
5786  025413     124  046511  020105  TIMOM:  .ASCIZ  /TIME OUT/
5787  025424  040504  040524  041440  DATCKM: .ASCIZ  /DATA CHECK/
5788  025437     122  047125  051440  RUNSBM: .ASCIZ  /RUN SET ILLEGALLY/
5789  025461     122  020130  042111  RXIDM:  .ASCIZ  /RX IDLE/
5790  025471     103  020104  046107  CDGLM:  .ASCIZ  /CD GLITCHED/
5791  025505     103  051524  043040  CTSFM:  .ASCIZ  /CTS FALILED/
5792  025521     124  020130  047516  TXNC:   .ASCIZ  /TX NOT COMPLETE/
5793  025541     122  020130  047516  RXNC:   .ASCIZ  /RX NOT COMPLETE/
5794  025561     123  041505  051040  RXM1:   .ASCIZ  /SEC REQ ERR WORD 1/
5795  025604  042523  020103  042522  RXM2:   .ASCIZ  /SEC REQ ERR WORD 2/
5796          025630                          .EVEN
5797
5808
5809
5817
5818
5819
5820
```

```
5822                                    .SBTTL  GLOBAL ERROR REPORT SECTION
5823
5824                                    ;++
5825                                    ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
5826                                    ; USED BY MORE THAN TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
5827                                    ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
5828                                    ;--
5829
5830
5831
5832
5848
5849  025630                            BGNMSG  ERR1
 (3)  025630                                                                    ERR1::
5850  025630                            PRINTB  #EVTF5A,OFSET,<B,GOOD>,<B,BAD>  ;INDIVIDUAL DATA COMPARE ERROR
 (10) 025630  005046                                                                CLR     -(SP)
 (10) 025632  153716  007447                                                        BISB    BAD,(SP)
 (9)  025636  005046                                                                CLR     -(SP)
 (9)  025640  153716  007446                                                        BISB    GOOD,(SP)
 (8)  025644  013746  007424                                                        MOV     OFSET,-(SP)
 (7)  025650  012746  017034                                                        MOV     #EVTF5A,-(SP)
 (6)  025654  012746  000004                                                        MOV     #4,-(SP)
 (3)  025660  010600                                                                MOV     SP,R0
 (4)  025662  104414                                                                TRAP    C$PNTB
 (4)  025664  062706  000012                                                        ADD     #12,SP
5851  025670                            ENDMSG
 (3)  025670                                                                    L10001:
 (3)  025670  104423                                                                TRAP    C$MSG
5852
5853  025672                            BGNMSG  ERR2
 (3)  025672                                                                    ERR2::
5854  025672                            PRINTB  #EFM2,TEMP4                     ;TOTAL DATA COMPARE FAILS ERROR
 (8)  025672  013746  007436                                                        MOV     TEMP4,-(SP)
 (7)  025676  012746  015453                                                        MOV     #EFM2,-(SP)
 (6)  025702  012746  000002                                                        MOV     #2,-(SP)
 (3)  025706  010600                                                                MOV     SP,R0
 (4)  025710  104414                                                                TRAP    C$PNTB
 (4)  025712  062706  000006                                                        ADD     #6,SP
5855  025716                            ENDMSG
 (3)  025716                                                                    L10002:
 (3)  025716  104423                                                                TRAP    C$MSG
5856
5857  025720                            BGNMSG  ERR10
 (3)  025720                                                                    ERR10::
5858  025720                            PRINTB  #EFM11,R4,TEMP3
 (9)  025720  013746  007434                                                        MOV     TEMP3,-(SP)
 (8)  025724  010446                                                                MOV     R4,-(SP)
 (7)  025726  012746  015550                                                        MOV     #EFM11,-(SP)
 (6)  025732  012746  000003                                                        MOV     #3,-(SP)
 (3)  025736  010600                                                                MOV     SP,R0
 (4)  025740  104414                                                                TRAP    C$PNTB
 (4)  025742  062706  000010                                                        ADD     #10,SP
5859  025746                            ENDMSG
 (3)  025746                                                                    L10003:
 (3)  025746  104423                                                                TRAP    C$MSG
5860
```

```
 5870  025750                                BGNMSG  ERR8                                ERR8::
  (3)  025750
 5871  025750                                PRINTB  #EVTF3D,TEMP3,TEMP4,CONOTM
  (10) 025750  013746  007442                                                            MOV     CONOTM,-(SP)
  (9)  025754  013746  007436                                                            MOV     TEMP4,-(SP)
  (8)  025760  013746  007434                                                            MOV     TEMP3,-(SP)
  (7)  025764  012746  016533                                                            MOV     #EVTF3D,-(SP)
  (6)  025770  012746  000004                                                            MOV     #4,-(SP)
  (3)  025774  010600                                                                    MOV     SP,R0
  (4)  025776  104414                                                                    TRAP    C$PNTB
  (4)  026000  062706  000012                                                            ADD     #12,SP
 5872  026004                                PRINTB  #PCPM,PCADD
  (8)  026004  013746  007462                                                            MOV     PCADD,-(SP)
  (7)  026010  012746  015516                                                            MOV     #PCPM,-(SP)
  (6)  026014  012746  000002                                                            MOV     #2,-(SP)
  (3)  026020  010600                                                                    MOV     SP,R0
  (4)  026022  104414                                                                    TRAP    C$PNTB
  (4)  026024  062706  000006                                                            ADD     #6,SP
 5873  026030                                ENDMSG
  (3)  026030                                                                          L10004:
  (3)  026030  104423                                                                    TRAP    C$MSG
 5874
 5875  026032                                BGNMSG  ERR9                                ERR9::
  (3)  026032
 5876  026032                                PRINTB  #EVTF3C,TEMP3,TEMP4
  (9)  026032  013746  007436                                                            MOV     TEMP4,-(SP)
  (8)  026036  013746  007434                                                            MOV     TEMP3,-(SP)
  (7)  026042  012746  016516                                                            MOV     #EVTF3C,-(SP)
  (6)  026046  012746  000003                                                            MOV     #3,-(SP)
  (3)  026052  010600                                                                    MOV     SP,R0
  (4)  026054  104414                                                                    TRAP    C$PNTB
  (4)  026056  062706  000010                                                            ADD     #10,SP
 5877  026062                                PRINTB  #PCPM,PCADD
  (8)  026062  013746  007462                                                            MOV     PCADD,-(SP)
  (7)  026066  012746  015516                                                            MOV     #PCPM,-(SP)
  (6)  026072  012746  000002                                                            MOV     #2,-(SP)
  (3)  026076  010600                                                                    MOV     SP,R0
  (4)  026100  104414                                                                    TRAP    C$PNTB
  (4)  026102  062706  000006                                                            ADD     #6,SP
 5878  026106                                ENDMSG
  (3)  026106                                                                          L10005:
  (3)  026106  104423                                                                    TRAP    C$MSG
 5879
 5880  026110                                BGNMSG  ERR13                               ERR13::
  (3)  026110
 5881  026110                                PRINTB  #EVTF3C,TEMP3,TEMP4
  (9)  026110  013746  007436                                                            MOV     TEMP4,-(SP)
  (8)  026114  013746  007434                                                            MOV     TEMP3,-(SP)
  (7)  026120  012746  016516                                                            MOV     #EVTF3C,-(SP)
  (6)  026124  012746  000003                                                            MOV     #3,-(SP)
  (3)  026130  010600                                                                    MOV     SP,R0
  (4)  026132  104414                                                                    TRAP    C$PNTB
  (4)  026134  062706  000010                                                            ADD     #10,SP
 5882  026140                                ENDMSG
  (3)  026140                                                                          L10006:
  (3)  026140  104423                                                                    TRAP    C$MSG
```

```
 5883
 5884  026142                          BGNMSG  ERR14                          ERR14::
  (3)  026142                                                                 MOV     CONOTM,-(SP)
 5885  026142                          PRINTB  #EVTF3D,TEMP3,TEMP4,CONOTM     MOV     TEMP4,-(SP)
  (10) 026142  013746  007442                                                MOV     TEMP3,-(SP)
  (9)  026146  013746  007436                                                MOV     #EVTF3D,-(SP)
  (8)  026152  013746  007434                                                MOV     #4,-(SP)
  (7)  026156  012746  016533                                                MOV     SP,R0
  (6)  026162  012746  000004                                                TRAP    C$PNTB
  (3)  026166  010600                                                        ADD     #12,SP
  (4)  026170  104414
  (4)  026172  062706  000012
 5886  026176                          ENDMSG                         L10007:
  (3)  026176                                                                 TRAP    C$MSG
  (3)  026176  104423
 5887
 5888  026200                          EXIT    MSG                            .WORD   J$JMP
  (4)  026200  000167                                                        .WORD   L10007-2-.
  (3)  026202  177772
 5889
 5890
```

```
5892                                    .SBTTL  GLOBAL SUBROUTINES SECTION
5893
5894                                    ;++
5895                                    ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
5896                                    ; THAT ARE USED IN MORE THAN ONE TEST.
5897                                    ;--
5898
5975
5976                                    .SBTTL          CLOCK SETUP SUBROUTINE
5977
5978                                    ;++
5979                                    ; FUNCTIONAL DESCRIPTION:
5980                                    ;       THIS SUBROUTINE SETS UP THE CLOCK INFORMATION TABLE FOLLOWING A ''CLOCK''
5981                                    ;       CALL EXECUTED IN THE INITIALIZATION CODE. BUT SINCE THE ''CLOCK'' CALL
5982                                    ;       SAYS NOTHING ABOUT AN LSI-11'S CLOCK, THIS ROUTINE IS ONLY USED IF A
5983                                    ;       LINE OR P-CLOCK IS FOUND.
5984                                    ;
5985                                    ; INPUTS:
5986                                    ;       R1= POINTS TO SUPERVISOR SPACE WHERE CLOCK INFO WAS RETURNED
5987                                    ;       R2= POINTS TO ''CLK'' TABLE WHERE CLOCK INFO WILL BE KEPT
5988                                    ;
5989                                    ; IMPLICIT INPUTS:
5990                                    ;       THE SUPERVISOR  SPACE WHERE CLOCK INFO WAS RETURNED BY THE ''CLOCK'' CALL
5991                                    ;
5992                                    ; OUTPUTS:
5993                                    ;       ''CLKCSR'' GETS LOADED WITH THE CLOCK'S CSR ADDRESS
5994                                    ;       ''CLKBR'' GETS LOADED WITH THE CLOCK'S INTERRUPT LEVEL
5995                                    ;       ''CLKVEC'' GETS LOADED WITH THE CLOCK'S INTERRUPT VECTOR
5996                                    ;       ''CLKHZ'' GETS LOADED WITH THE LINE FREQ. (HERTZ RATE) WHICH DETERMINES
5997                                    ;               THE NUMBER OF TICKS IN A SECOND
5998                                    ;
5999                                    ; CALLING SEQUENCE:
6000                                    ;       JSR     PC,CLKSET               ;CALL CLOCK SETUP WITH R1 & R2 SETUP
6001                                    ;--
6002
6003  026204                            CLKSET:
6004  026204  012122                           MOV     (R1)+,(R2)+             ;LOAD CLOCK'S CSR ADDR. INTO ''CLKCSR''
6005  026206  012112                           MOV     (R1)+,(R2)              ;LOAD CLOCK'S INT. LEVEL INTO ''CLKBR''
6006  026210  006312                           ASL     (R2)                    ;ADJUST THE INT. LEVEL FOR LOADING INTO
6007  026212  006312                           ASL     (R2)                    ;    THE PSW WITH A ''SETVEC'' CALL
6008  026214  006312                           ASL     (R2)
6009  026216  006312                           ASL     (R2)
6010  026220  006322                           ASL     (R2)+
6011  026222  012122                           MOV     (R1)+,(R2)+             ;LOAD CLOCK'S INT. VECTOR INTO ''CLKVEC''
6012  026224  012122                           MOV     (R1)+,(R2)+             ;LOAD CLOCK'S HERTZ RATE INTO ''CLKHZ''
6013  026226  000207                           RTS     PC
6014
```

```
                                .SBTTL          CLOCK INTERRUPT SERVICE ROUTINE
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       THIS IS THE CLOCK INTERRUPT SERVICE ROUTINE WHICH TAKES CARE OF
                                ;       KEEPING THE 'TIME-SINCE-START' AND COUNTING DOWN ANY OF THE
                                ;       'EVENT' TIMERS. THE TIMERS ARE USED TO TIME COMPLETION OF  DEVICE
                                ;       REQUESTS. THE 'TIME-SINCE-START' IS USED TO BE LOGGED WITH EACH ENTRY
                                ;       INTO THE EVENT LOG.
                                ;
                                ; IMPLICIT INPUTS:
                                ;       TIMTCK: THE CURRENT NO. OF TICKS LEFT TO BE COUNTED UNTIL A SECOND
                                ;               HAS BEEN COUNTED OFF
                                ;       CLKHZ:  THE NO. OF TICKS IN A SECOND, DETERMINED BY THE SYS. LINE FREQ.
                                ;       TIMMIN & TIMSEC: CURRENT VALUE OF 'TIME-SINCE-START'
                                ;                        IN MINUTES & SECONDS
                                ;       TIMER 1,2, & S: CURRENT VALUES OF THE 'EVENT TIMERS'
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       NEW VALUE OF EVENT TIMER '1' DECREMENTED BY 1 TICK IF IT WAS NON-ZERO
                                ;       NEW VALUE OF EVENT TIMER '2' DECREMENTED BY 1 TICK IF IT WAS NON-ZERO
                                ;       NEW VALUE OF EVENT TIMER 'S' DECREMENTED BY 1 SECOND IF IT WAS NON-ZERO
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;       THE CLOCK IS DISABLED UPON ENTRY AND REENABLED WHEN LEAVING
                                ;
                                ; CALLING SEQUENCE:
                                ;       THIS ROUTINE IS CALLED WHEN THE CLOCK INTERRUPTS THRU 'CLKVEC'.
                                ;       THE ADDRESS OF THIS ROUTINE WAS LOADED INTO THE CLOCK'S INTERRUPT
                                ;       VECTOR WITH A SUPERVISOR 'SETVEC' CALL.
                                ;--

026230                                  BGNSRV  CLKINT
(3) 026230                                                              CLKINT::

026230  005077  161266                  CLR     @CLKCSR         ;DISABLE THE CLOCK FORM INTERRUPTING
026234  005337  007540                  DEC     TIMTCK          ;DECREMENT THE # OF TICKS/SEC.
026240  001015                          BNE     1$              ;GO CHECK TIMERS (1&2-TICKS, 3-SECONDS)
026242  013737  007530  007540          MOV     CLKHZ,TIMTCK    ;RESET THE # OF TICKS/SEC.
026250  005237  007536                  INC     TIMSEC          ;INC # OF SECS-SINCE-START
026254  022737  000074  007536          CMP     #60.,TIMSEC     ;SEE IF WE'VE COUNTED 60 SECS. YET
026262  001004                          BNE     1$              ;IF NOT, GO CHECK TIMERS
026264  005237  007534                  INC     TIMMIN          ; ELSE INC MINUTES-SINCE-START
026270  005037  007536                  CLR     TIMSEC          ; AND RESTART SECOND COUNTER

026274  005737  007542          1$:     TST     TIMER1          ;SEE IF TIMER #1, TIMING ANYTHING
026300  001402                          BEQ     2$              ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
026302  005337  007542                  DEC     TIMER1          ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
026306  005737  007544          2$:     TST     TIMER2          ;SEE IF TIMER #2, TIMING ANYTHING
026312  001402                          BEQ     3$              ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
026314  005337  007544                  DEC     TIMER2          ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
026320  005737  007546          3$:     TST     TIMERS          ;SEE IF TIMER #3, TIMING ANYTHING
026324  001406                          BEQ     4$              ; IF=0, NOTHING BEING TIMED, LEAVE
026326  023737  007530  007540          CMP     CLKHZ,TIMTCK    ;SEE IF A SECOND HAS BEEN COUNTED OFF
026334  001002                          BNE     4$              ; BR IF NO
026336  005337  007546                  DEC     TIMERS          ; ELSE DECREMENT THE TIMER VALUE (BY 1 SEC.)
```

```
  6071  026342  013777  007532  161152  4$:     MOV     CLKEN,@CLKCSR   ;REENABLE THE CLOCK TO INTERRUPT
  6072  026350                                  ENDSRV
   (3)  026350                                                                  L10010:
   (2)  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:
```

```
6130  026434  012737  020036  007430          MOV     #SDVE,TEMP1     ;SET UP MSG. TO PRINT
6131  026442  012737  000010  007426          MOV     #DER,TEMP       ;SET UP EVENT TYPE
6132  026450  000474                          BR      LOGS3           ;GO LOG EVENT AND TIME
6133
6134  026452                          LOGDVI:
6135  026452  012737  020060  007430          MOV     #SDVI,TEMP1     ;SET UP MSG. TO PRINT
6136  026460  012737  000012  007426          MOV     #DVI,TEMP       ;SET UP EVENT TYPE
6137  026466  113737  007470  007432          MOVB    MODTYP,TEMP2
6138  026474  113737  007472  007433          MOVB    MLTYP,TEMP2+1
6139  026502  013737  007500  007434          MOV     RPASS,TEMP3
6140  026510  013737  007476  007436          MOV     PARAM,TEMP4     ;SET UP EVNT ENTRIES
6141  026516  000451                          BR      LOGS3           ;GO LOG EVENT AND TIME
6142
6143  026520                          LOGCMP:
6144  026520  012737  020047  007430          MOV     #SCM,TEMP1      ;SET UP MSG. TO PRINT
6145  026526  012737  000014  007426          MOV     #DCK,TEMP       ;SET UP EVENT TYPE
6146  026534  000442                          BR      LOGS3
6147  026536                          LOGCML:
6148  026536  012737  020071  007430          MOV     #SCML,TEMP1
6149  026544  012737  000020  007426          MOV     #DLE,TEMP       ;SET UP MSG. AND TYPE
6150  026552  000433                          BR      LOGS3           ;GO LOG EVENT AND TIME
6151  026554                          LOGCMD:
6152  026554  012737  020102  007430          MOV     #SCMD,TEMP1
6153  026562  012737  000022  007426          MOV     #DDE,TEMP
6154  026570  000424                          BR      LOGS3           ;GO LOG MSG TYPE AND TIME
6155  026572                          LOGEOP:
6156  026572  012737  020113  007430          MOV     #SEOP,TEMP1
6157  026600  012737  000024  007426          MOV     #EOP,TEMP
6158  026606  000415                          BR      LOGS3           ;GO LOG MSG TYPE AND TIME
6159
6160  026610  013746  007400          LOGS1:  MOV     ERRCNT,-(SP)    ;SAVE CURRENT ERROR COUNT
6161  026614  004737  044004                  JSR     PC,DVMODS       ; GO GET MODEM STATUS
6162  026620  012604                          MOV     (SP)+,R4        ;GET SAVED ERRCNT VALUE
6163  026622  020437  007400                  CMP     R4,ERRCNT       ;WHERE ANY ERRORS FOUND
6164  026626  001402                          BEQ     1$              ; BR IF NONE
6165  026630  000137  027044                  JMP     LOGEX           ; ELSE, LEAVE WITHOUT LOGGING ANYTHING
6166                                                                  ;   BUT THE DEVICE ERROR FROM 'DVMODS'
6167  026634  013737  010456  007436  1$:     MOV     MODS,TEMP4      ;AND PUT IT IN TEMP4
6168
6169  026642                          LOGS3:
6170  026642  022737  000006  007426          CMP     #RXC,TEMP
6171  026650  001434                          BEQ     LOGS5           ;IF RXC DONT PRINT
6172  026652  032737  000001  007476          BIT     #STATB,PARAM
6173  026660  001430                          BEQ     LOGS5           ;IF NO STATUS SELECTED
6174                                                                  ;GO TO 5
6175
6176  026662  022737  000010  007372          CMP     #10,LNCNT       ;HAVE WE DONE 10?
6177  026670  001012                          BNE     LOGS4           ;IF NOT GO TO 4
6178  026672  005037  007372                  CLR     LNCNT           ;ESLE CLEAR IT
6179
6180  026676                                  PRINTF  #CR             ;ELSE PRINT CR
  (7) 026676  012746  020000                                                  MOV     #CR,-(SP)
  (6) 026702  012746  000001                                                  MOV     #1,-(SP)
  (3) 026706  010600                                                          MOV     SP,R0
  (4) 026710  104417                                                          TRAP    C$PNTF
  (4) 026712  062706  000004                                                  ADD     #4,SP
```

```
 6181  026716                                LOGS4:
 6182  026716  005237  007372                      INC     LNCNT           ;INC COUNTER OF # OF AWAKE MSGS
 6183  026722                                      PRINTF  TEMP1           ;PRINT OPERATOR AWAKE MSG.
   (7) 026722  013746  007430                                                          MOV     TEMP1,-(SP)
   (6) 026726  012746  000001                                                          MOV     #1,-(SP)
   (3) 026732  010600                                                                  MOV     SP,R0
   (4) 026734  104417                                                                  TRAP    C$PNTF
   (4) 026736  062706  000004                                                          ADD     #4,SP
 6184  026742  010346                      LOGS5:  MOV     R3,-(SP)        ;SAVE R3 ON THE STACK
 6185  026744  013703  007550                      MOV     EVTPTR,R3
 6186  026750  113723  007426                      MOVB    TEMP,(R3)+      ;LOG EVENT
 6187  026754  013737  007530  007426              MOV     CLKHZ,TEMP
 6188  026762  163737  007540  007426              SUB     TIMTCK,TEMP
 6189  026770  113723  007426                      MOVB    TEMP,(R3)+      ;LOG TIME SINCE START
 6190  026774  113723  007536                      MOVB    TIMSEC,(R3)+
 6191  027000  113723  007534                      MOVB    TIMMIN,(R3)+    ;TICKS,SECS AND MINS.
 6192  027004  013723  007432                      MOV     TEMP2,(R3)+     ;LOG EVNT ENTRY 3
 6193  027010  013723  007434                      MOV     TEMP3,(R3)+     ;LOG EVNT ENTRY 4
 6194  027014  013723  007436                      MOV     TEMP4,(R3)+     ;LOG EVNT ENTRY 5
 6195  027020  020327  010454                      CMP     R3,#EVTEND
 6196  027024  103404                            BLO     LOGS2           ;IF EVENT LOG FULL GO
 6197                                                                      ;CONTINUE;ELSE GO TO 2
 6198  027026  012713  177777                      MOV     #-1,(R3)        ;LOG A TABLE END
 6199  027032  012703  007552                      MOV     #EVTLOG,R3      ;PUT R3 TO START OF TABLE
 6200  027036  010337  007550              LOGS2:  MOV     R3,EVTPTR       ;RESTORE POINTER
 6201  027042  012603                            MOV     (SP)+,R3        ;RESTORE R3
 6202  027044  000207                      LOGEX:  RTS     PC
 6203
 6204
```

```
                        .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
(7) 027054  012746  014020                                              MOV     #RHLP0,-(SP)
(6) 027060  012746  000001                                              MOV     #1,-(SP)
(3) 027064  010600                                                      MOV     SP,R0
(4) 027066  104417                                                      TRAP    C$PNTF
(4) 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$RP,CMDBUF,A,0,1,72.,NO
(3) 027104  104443                                                      TRAP    C$GMAN
(3) 027106  000406                                                      BR      10000$
(4) 027110  003060                                                      .WORD   CMDBUF
(5) 027112  000142                                                      .WORD   T$CODE
(5) 027114  012504                                                      .WORD   CLI$RP
(5) 027116  000000                                                      .WORD   0
(5) 027120  000001                                                      .WORD   T$LOLIM
(5) 027122  000110                                                      .WORD   T$HILIM
(3) 027124                                                      10000$:
6220 027124  012737  003060  003544          MOV     #CMDBUF,P$BUFA  ;INPUT BUFFER
6230 027132  012737  027266  003546          MOV     #CLIRT,P$TREE   ;REPORT CLI TREE
6231 027140  012737  027464  003550          MOV     #CLIRAC,P$ACT   ;ACTION ROUTINES
6232 027146  005037  003204                  CLR     QUALFG
6233 027152  004737  032642                  JSR     PC,P$TRV        ;GO PARSE COMMAND LINE
6234 027156  105737  003561                  TSTB    P$GDBD          ;COMMAND OK ?
6235 027162  001412                          BEQ     1$              ;YES,BRANCH
5236 027164                                  PRINTF  #CLIERM         ;PRINT INVALID INPUT MESSAGE
(7) 027164  012746  012512                                                      MOV     #CLIERM,-(SP)
'6) 027170  012746  000001                                                      MOV     #1,-(SP)
(3) 027174  010600                                                              MOV     SP,R0
(4) 027176  104417                                                              TRAP    C$PNTF
(4) 027200  062706  000004                                                      ADD     #4,SP
6237 027204  000137  027074                  JMP     GETRCL          ;TRY AGAIN
6238
6239 027210  105737  003560          1$:     TSTB    P$NNUF          ;MORE COMMAND NEEDED ?
6240 027214  001412                          BEQ     10$             ;NO,BRANCH
6241 027216                                  PRINTF  #CLINUF         ;INCOMPLETE MESSAGE
(7) 027216  012746  012542                                                      MOV     #CLINUF,-(SP)
```

```
  (6)  027222  012746  000001                                                          MOV     #1,-(SP)
  (3)  027226  010600                                                                 MOV     SP,R0
  (4)  027230  104417                                                                 TRAP    C$PNTF
  (4)  027232  062706  000004                                                         ADD     #4,SP
6242   027236  000137  027074                JMP     GETRCL                  ;TRY AGAIN
6243
6244   027242  023727  003202  000002  10$:  CMP     KEYWD1,#RPEXT           ;EXIT COMMAND ?
6245   027250  001402                        BEQ     20$                     ;YES,BRANCH
6246   027252  000137  027074                JMP     GETRCL                  ;GET ANOTHER COMMAND
6247   027256  012604                  20$:  MOV     (SP)+,R4                ;RESTORE R4
6248   027260  012603                        MOV     (SP)+,R3                ;RESTORE R3
6249   027262  012602                        MOV     (SP)+,R2                ;RESTORE R2
6250   027264  000207                        RTS     PC                      ;RETURN
```

```
 6252                                .SBTTL  COMMAND LINE PARSING TREE FOR REPORT
 6253  027266                        CLIRT:  CLI CLISPA,0,R10$                ;SKIP SPACES IN COMMAND LINE
 6254  027272                        R10$:   CLI <'?>,RPHLP,R11$              ;IF INPUT = ? THEN PRINT HELP MESSAGE
 6255  027276                                CLI CLIEXI,0                     ;;AND EXIT PARSER
 6256  027300                        R11$:   CLI CLISTR,RPHLP,R12$,<'HELP'>   ;IF INPUT = 'HELP' THEN PRINT HELP
 6257  027314                                CLI CLIEXI,0                     ;;MESSAGE AND EXIT PARSER
 6258  027316                        R12$:   CLI CLISTR,RPEXT,R13$,<'EXIT'>   ;IF INPUT = 'EXIT' THEN SET KEYWORD =
 6259  027332                                CLI CLIEXI,0                     ;;RPEXT AND EXIT PARSER
 6260  027334                        R13$:   CLI CLISTR,RPLOG,R14$,<'LOG'>    ;IF INPUT = 'LOG' THEN GO PRINT EVENT
 6261  027346                                CLI CLIEXI,0                     ;;LOG AND EXIT PARSER
 6262  027350                        R14$:   CLI CLISTR,RNOTNF,R30$,<'BASE'>;IF INPUT = 'BASE' THEN MORE COMMAND
 6263  027364                                CLI CLIBR,0,R15$                 ;;LINE IS NEEDED
 6264  027370                        R15$:   CLI <'/>,RNOTNF,R125$            ;IF INPUT = '/' THEN LOOK FOR MORE
 6265  027374                                CLI CLISTR,RPSWE,R16$,<'ERROR'> ;IF INPUT = 'ERROR' THEN GO PRINT
 6266  027410                                CLI CLIEXI,0                     ;;ERROR INFORMATION
 6267  027412                        R16$:   CLI CLISTR,RPSWF,R17$,<'FULL'>   ;IF INPUT = 'FULL' THEN GO PRINT
 6268  027426                                CLI CLIEXI,0                     ;;ENTIRE BASE TABLE
 6269  027430                        R17$:   CLI CLISTR,RNOTNF,R30$,<'OFFSET'>;IF INPUT = 'OFFSET' THEN LOOK FOR
 6270  027446                                CLI <'=>,0,R30$                  ;;'='
 6271  027452                                CLI CLIOCT,RPSWO,R30$            ;IF INPUT = OCTAL VALUE THEN GO
 6272  027456                                CLI CLIEXI,0                     ;;PRINT SINGLE BASE TABLE ITEM
 6273  027460                        R30$:   CLI CLIERR,0
 6274  027462                        R125$:  CLI CLIEXI,0
```

```
6276                                    .SBTTL  CLI ACTION DISPATCHER AND ROUTINES
6277  027464  006302                    CLIRAC: ASL     R2              ;SET UP INDEX
6278  027466  016202  027502                    MOV     10$(R2),R2      ;
6279  027472  062702  027502                    ADD     #10$,R2         ;
6280  027476  004712                            JSR     PC,(R2)         ;GO DO ACTION
6281  027500  000207                            RTS     PC              ;RETURN
6282  027502  000026                    10$:    .WORD   ACTRNL-10$      ;NULL
6283  027504  000030                            .WORD   ACTRHL-10$      ;HELP ROUTINE
6284  027506  000074                            .WORD   ACTREX-10$      ;EXIT ROUTINE
6285  027510  000104                            .WORD   ACTRLG-10$      ;REPORT EVENT LOG ROUTINE
6286  027512  000120                            .WORD   ACTSWE-10$      ;REPORT ERRORS ROUTINE
6287  027514  000262                            .WORD   ACTSWF-10$      ;REPORT ENTIRE BASE TABLE
6288  027516  000374                            .WORD   ACTSWO-10$      ;REPORT SINGLE BASE ADDRESS
6289  027520  000020                            .WORD   ACTRNF-10$      ;MORE COMMAND NEEDED
6290
6291                                            ;MORE COMMAND NEEDED
6292  027522  112737  177777  003560    ACTRNF: MOVB    #-1,P$NNUF      ;MORE COMMAND NEEDED
6293  027530  000207                    ACTRNL: RTS     PC              ;NULL
6294
6295                                            ;PRINT HELP MESSAGE
6296  027532  012702  003230            ACTRHL: MOV     #RHLPTB,R2      ;INDEX FOR HELP MESSAGES
6297  027536                            1$:     PRINTF  #HLPF,(R2)+     ;PRINT IT
  (8) 027536  012246                                                    MOV     (R2)+,-(SP)
  (7) 027540  012746  013225                                            MOV     #HLPF,-(SP)
  (6) 027544  012746  000002                                            MOV     #2,-(SP)
  (3) 027550  010600                                                    MOV     SP,R0
  (4) 027552  104417                                                    TRAP    C$PNTF
  (4) 027554  062706  000006                                            ADD     #6,SP
6298  027560  020227  003246                    CMP     R2,#RHLPEN      ;LAST MESSAGE ?
6299  027564  001364                            BNE     1$              ;NO,BRANCH
6300  027566  012737  000001  003202            MOV     #RPHLP,KEYWD1   ;SET KEYWORD
6301  027574  000207                            RTS     PC              ;RETURN
6302
6303                                            ;EXIT REPORT LEVEL
6304  027576  012737  000002  003202    ACTREX: MOV     #RPEXT,KEYWD1   ;SET KEYWORD AND RETURN
6305  027604  000207                            RTS     PC
6306
6307                                            ;PRINT ERROR LOG
6308  027606  004737  030574            ACTRLG: JSR     PC,REPLOG       ;GO PRINT EVENT LOG
6309  027612  012737  000003  003202            MOV     #RPLOG,KEYWD1   ;SET KEYWORD
6310  027620  000207                            RTS     PC              ;RETURN
6311
6312                                            ;PRINT ONLY ERROR LOCATIONS
6313  027622  005737  012406            ACTSWE: TST     OPTYP           ;DMR ?
6314  027626  001026                            BNE     10$             ;YES,BRANCH
6315  027630  012737  003432  007450            MOV     #DMCIND,INDEX   ;SETUP DMC MESSAGES
6316  027636  062737  000006  007450            ADD     #6,INDEX        ;POINT TO CORRECT MESSAGE
6317  027644  012737  003462  007452            MOV     #DMCEND,INDEXE  ;LAST DMC ADDRESS
6318  027652  012737  017370  007454            MOV     #BASE,BEND      ;SET UP LAST ADDRESS
6319  027660  062737  000012  007454            ADD     #12,BEND        ;;TO BE PRINTED
6320  027666  012737  017370  007456            MOV     #BASE,BDATA     ;BASE TABLE START ADDRESS
6321  027674  062737  000003  007456            ADD     #3,BDATA        ;ERROR START ADDRESS
6322  027702  000425                            BR      20$             ;
6323  027704  012737  003250  007450    10$:    MOV     #DMRIND,INDEX   ;SETUP FOR DMR MESSAGES
6324  027712  062737  000006  007450            ADD     #6,INDEX        ;POINT TO FIRST ERROR MESSAGE
6325  027720  012737  003430  007452            MOV     #DMREND,INDEXE  ;LAST DMR MESSAGE
```

```
 6326  027726  012737  017370  007454          MOV     #BASE,BEND      ;SETUP LAST ADDRESS
 6327  027734  062737  000041  007454          ADD     #41,BEND        ;;TO BE PRINTED
 6328  027742  012737  017370  007456          MOV     #BASE,BDATA     ;START ADDRESS BASE TABLE
 6329  027750  062737  000003  007456          ADD     #3,BDATA        ;START ADDRESS ERRORS
 6330  027756  004737  030336          20$:    JSR     PC,RPBASE       ;GO PRINT DATA
 6331  027762  000207                          RTS     PC              ;RETURN
 6332
 6333                                          ;PRINT FULL BASE TABLE
 6334  027764  005737  012406          ACTSWF: TST     OPTYP           ;DMR ?
 6335  027770  001020                          BNE     10$             ;YES,BRANCH
 6336  027772  012737  003432  007450          MOV     #DMCIND,INDEX   ;SETUP DMC MESSAGES
 6337  030000  012737  003462  007452          MOV     #DMCEND,INDEXE  ;LAST MESSAGE
 6338  030006  012737  017370  007454          MOV     #BASE,BEND      ;TABLE START ADDRESS
 6339  030014  062737  000377  007454          ADD     #377,BEND       ;PRINT 256. BYTES OF DATA
 6340  030022  012737  017370  007456          MOV     #BASE,BDATA     ;FIRST ADDRESS TO PRINT
 6341  030030  000417                          BR      20$
 6342  030032  012737  003250  007450  10$:    MOV     #DMRIND,INDEX   ;SETUP DMR MESSAGES
 6343  030040  012737  003430  007452          MOV     #DMREND,INDEXE  ;LAST DMR MESSAGE
 6344  030046  012737  017370  007454          MOV     #BASE,BEND      ;TABLE START ADDRESS
 6345  030054  062737  000177  007454          ADD     #177,BEND       ;PRINT 128. BYTES OF DATA
 6346  030062  012737  017370  007456          MOV     #BASE,BDATA     ;FIRST ADDRESS TO PRINT
 6347  030070  004737  030336          20$:    JSR     PC,RPBASE       ;GO PRINT DATA
 6348  030074  000207                          RTS     PC              ;RETURN
 6349
 6350                                          ;PRINT SINGLE TABLE LOCATION
 6351  030076  105037  003560          ACTSWO: CLRB    P$NMUF          ;INIT NOT ENOUGH FLAG
 6352  030102  005737  012406                  TST     OPTYP           ;DMR?
 6353  030106  001004                          BNE     5$              ;YES,BRANCH
 6354  030110  012737  000377  007454          MOV     #377,BEND       ;BASE TABLE FOR DMC = 256 BYTES
 6355  030116  000403                          BR      7$              ;BRANCH
 6356  030120  012737  000177  007454  5$:     MOV     #177,BEND       ;BASE TABLE FOR DMR = 128 BYTES
 6357  030126  023737  003554  007454  7$:     CMP     P$NUM,BEND      ;DMC = 256 BYTES   DMR = 128 BYTES
 6358  030134  101416                          BLOS    10$             ;YES,BRANCH
 6359  030136                                  PRINTF  #RPTIV,P$NUM    ;PRINT ILLEGAL VALUE
   (8) 030136  013746  003554                                                  MOV     P$NUM,-(SP)
   (7) 030142  012746  014370                                                  MOV     #RPTIV,-(SP)
   (6) 030146  012746  000002                                                  MOV     #2,-(SP)
   (3) 030152  010600                                                          MOV     SP,R0
   (4) 030154  104417                                                          TRAP    C$PNTF
   (4) 030156  062706  000006                                                  ADD     #6,SP
 6360  030162  112737  177777  003561          MOVB    #-1,P$GDBD      ;SET BAD DATA
 6361  030170  000461                          BR      30$             ;RETURN
 6362  030172  013701  003554          10$:    MOV     P$NUM,R1        ;OFFSET VALUE
 6363  030176  006301                          ASL     R1              ;MULTIPLY BY 2
 6364  030200  005737  012406                  TST     OPTYP           ;DMR ?
 6365  030204  001025                          BNE     15$             ;YES,BRANCH
 6366  030206  012737  003432  007450          MOV     #DMCIND,INDEX   ;DMC MESSAGES
 6367  030214  060137  007450                  ADD     R1,INDEX        ;GET RIGHT MESSAGE
 6368  030220  012737  003462  007452          MOV     #DMCEND,INDEXE  ;LAST DMC MESSAGE
 6369  030226  012737  017370  007454          MOV     #BASE,BEND      ;TABLE ADDRESS
 6370  030234  063737  003554  007454          ADD     P$NUM,BEND      ;LAST ADDRESS
 6371  030242  012737  017370  007456          MOV     #BASE,BDATA     ;BASE ADDRESS
 6372  030250  063737  003554  007456          ADD     P$NUM,BDATA     ;ADD OFFSET
 6373  030256  000424                          BR      20$             ;GO PRINT DATA
 6374  030260  012737  003250  007450  15$:    MOV     #DMRIND,INDEX   ;SETUP FOR DMR MESSAGES
 6375  030266  060137  007450                  ADD     R1,INDEX        ;GET CORRECT MESSAGE
```

```
 6376  030272  012737  003430  007452          MOV     #DMREND,INDEXE  ;LAST DMR MESSAGE
 6377  030300  012737  017370  007454          MOV     #BASE,BEND      ;TABLE ADDRESS
 6378  030306  063737  003554  007454          ADD     P$NUM,BEND      ;LAST ADDRESS
 6379  030314  012737  017370  007456          MOV     #BASE,BDATA     ;TABLE ADDRESS
 6380  030322  063737  003554  007456          ADD     P$NUM,BDATA     ;ADD OFFSET
 6381  030330  004737  030336          20$:    JSR     PC,RPBASE       ;GO PRINT SINGLE LOCATION
 6382  030334  000207                  30$:    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
  (7)   030342  012746  030476                                          MOV     #BTHEAD,-(SP)
  (6)   030346  012746  000001                                          MOV     #1,-(SP)
  (3)   030352  010600                                                  MOV     SP,R0
  (4)   030354  104417                                                  TRAP    C$PNTF
  (4)   030356  062706  000004                                          ADD     #4,SP
030362  013702  007450                  MOV     INDEX,R2        ;POINTER TO MESSAGES
030366  013701  007456                  MOV     BDATA,R1        ;ADDRESS OF DATA
030372  010137  007434          10$:    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$:    MOV     (R2)+,TEMP2     ;READ MESSAGE ADDRESS
030420                          PRINTF  #DMFMT,TEMP3,<B,TEMP1>,TEMP2    ;PRINT DATA AND MESSAGE
  (10)  030420  013746  007432                                          MOV     TEMP2,-(SP)
  (9)   030424  005046                                                  CLR     -(SP)
  (9)   030426  153716  007430                                          BISB    TEMP1,(SP)
  (8)   030432  013746  007434                                          MOV     TEMP3,-(SP)
  (7)   030436  012746  030547                                          MOV     #DMFMT,-(SP)
  (6)   030442  012746  000004                                          MOV     #4,-(SP)
  (3)   030446  010600                                                  MOV     SP,R0
  (4)   030450  104417                                                  TRAP    C$PNTF
  (4)   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  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
```

```
                                .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.
 (7)    030616  012746  016250                                    MOV     #NULEVT,-(SP)
 (6)    030622  012746  000001                                    MOV     #1,-(SP)
 (3)    030626  010600                                            MOV     SP,R0
 (4)    030630  104416                                            TRAP    C$PNTS
 (4)    030632  062706  000004                                    ADD     #4,SP
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
 (7)    030712  012746  016310                                    MOV     #EVTF0,-(SP)
 (6)    030716  012746  000001                                    MOV     #1,-(SP)
 (3)    030722  010600                                            MOV     SP,R0
 (4)    030724  104416                                            TRAP    C$PNTS
 (4)    030726  062706  000004                                    ADD     #4,SP
030732  112203                  MOVB    (R2)+,R3        ;PUT EVENT TYPE INTO R3
030734  112237  010612          MOVB    (R2)+,EVTTCK
030740  112237  010606          MOVB    (R2)+,EVTSEC    ;PUT EVENT TIME (TICKS,SECS,MINS IN TEMP LOC.S)
030744  112237  010610          MOVB    (R2)+,EVTMIN
030750                          PRINTS  #EVTF1,EVTMIN,EVTSEC,EVTTCK,EVTLST(R3)  ;PRINT EVENT TIME AND DESCRIPT.
 (11)   030750  016346  010514                                    MOV     EVTLST(R3),-(SP)
 (10)   030754  013746  010612                                    MOV     EVTTCK,-(SP)
 (9)    030760  013746  010606                                    MOV     EVTSEC,-(SP)
 (8)    030764  013746  010610                                    MOV     EVTMIN,-(SP)
 (7)    030770  012746  016403                                    MOV     #EVTF1,-(SP)
 (6)    030774  012746  000005                                    MOV     #5,-(SP)
 (3)    031000  010600                                            MOV     SP,R0
 (4)    031002  104416                                            TRAP    C$PNTS
 (4)    031004  062706  000014                                    ADD     #14,SP
031010  000173  010622          JMP     @RPTDSP(R3)     ;DISPATCH TO DECODING SECTION FOR SPECIFIC TYPE

031014  012237  010614  RPTTXQ: MOV     (R2)+,EVTADD    ;STORE MESSAGE ADDRESS FOR PRINTING
031020  012237  010616          MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT FOR PRINTING
031024  012203                  MOV     (R2)+,R3        ;STORE MODEM STATUS FOR PRINTING
```

```
 6463  031026                                  PRINTS  #EVTF2,EVTADD,EVTBCT    ;PRINT ADDR,BYTE CNT
  (9)  031026  013746  010616                                                   MOV     EVTBCT,-(SP)
  (8)  031032  013746  010614                                                   MOV     EVTADD,-(SP)
  (7)  031036  012746  016432                                                   MOV     #EVTF2,-(SP)
  (6)  031042  012746  000003                                                   MOV     #3,-(SP)
  (3)  031046  010600                                                           MOV     SP,R0
  (4)  031050  104416                                                           TRAP    C$PNTS
  (4)  031052  062706  000010                                                   ADD     #10,SP
 6464  031056  004737  031442                  JSR     PC,RPTMSB       ;GO PRINT MODEM STATUS
 6465  031062  000137  030642                  JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
 6466
 6467  031066  012237  010620          RPTDER: MOV     (R2)+,EVTTMP    ;GET ADDRESS OF DEVICE INFO MESSAGE
 6468  031072  012237  010650                  MOV     (R2)+,DEV1      ;STORE DEVICE REG CONTENTS FOR PRINTING
 6469  031076  012237  010652                  MOV     (R2)+,DEV2
 6470  031102                                  PRINTS  #EVTF3,EVTTMP   ;PRINT DEVICE REG CONTENTS.
  (8)  031102  013746  010620                                                   MOV     EVTTMP,-(SP)
  (7)  031106  012746  016504                                                   MOV     #EVTF3,-(SP)
  (6)  031112  012746  000002                                                   MOV     #2,-(SP)
  (3)  031116  010600                                                           MOV     SP,R0
  (4)  031120  104416                                                           TRAP    C$PNTS
  (4)  031122  062706  000006                                                   ADD     #6,SP
 6471  031126                                  PRINTS  #EVTF3C,DEV1,DEV2
  (9)  031126  013746  010652                                                   MOV     DEV2,-(SP)
  (8)  031132  013746  010650                                                   MOV     DEV1,-(SP)
  (7)  031136  012746  016516                                                   MOV     #EVTF3C,-(SP)
  (6)  031142  012746  000003                                                   MOV     #3,-(SP)
  (3)  031146  010600                                                           MOV     SP,R0
  (4)  031150  104416                                                           TRAP    C$PNTS
  (4)  031152  062706  000010                                                   ADD     #10,SP
 6472  031156  000137  030642                  JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
 6473
 6474  031162  005037  010650          RPTDVI: CLR     DEV1
 6475  031166  005037  010652                  CLR     DEV2            ;CLEAR UPPER BYTES OF DEV1 & DEV2 BEFORE USE
 6476  031172  112237  010650                  MOVB    (R2)+,DEV1      ;STORE SETUP OPERATION PARAMETERS FOR PRINTING
 6477  031176  112237  010652                  MOVB    (R2)+,DEV2
 6478  031202  012237  010654                  MOV     (R2)+,DEV3
 6479  031206  012237  010656                  MOV     (R2)+,DEV4
 6480  031212  010246                          MOV     R2,-(SP)        ;SAVE R2 ON THE STACK
 6481  031214  004737  032340                  JSR     PC,SHWOP        ;GO PRINT MODE, MAINT-LOOP TYPE, PARAMTERS.
 6482  031220  012602                          MOV     (SP)+,R2        ;RESTORE R2
 6483  031222  000137  030642                  JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
 6484  031226  012237  010614          RPTEOP: MOV     (R2)+,EVTADD
 6485  031232  012237  010616                  MOV     (R2)+,EVTBCT
 6486  031236  012237  010620                  MOV     (R2)+,EVTTMP
 6487  031242                                  PRINTS  #EVTF4B,EVTADD,EVTBCT,EVTTMP    ;PRINT ADDR,RXBYTES,CMPBYTES.
 (10)  031242  013746  010620                                                   MOV     EVTTMP,-(SP)
  (9)  031246  013746  010616                                                   MOV     EVTBCT,-(SP)
  (8)  031252  013746  010614                                                   MOV     EVTADD,-(SP)
  (7)  031256  012746  016755                                                   MOV     #EVTF4B,-(SP)
  (6)  031262  012746  000004                                                   MOV     #4,-(SP)
  (3)  031266  010600                                                           MOV     SP,R0
  (4)  031270  104416                                                           TRAP    C$PNTS
  (4)  031272  062706  000012                                                   ADD     #12,SP
 6488
 6489  031276  000137  030642                  JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY
 6490
```

```
                                RPTDDE: MOV     (R2)+,EVTADD    ;STORE MESSAGE ADDRESS FOR PRINTING
                                        MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT FOR PRINTING
                                        MOV     (R2)+,EVTTMP    ;STORE TOTAL # OF CMP ERRORS
                                        PRINTS  #EVTF4,EVTADD,EVTBCT,EVTTMP     ;PRINT ADDR, BYTE CNT, # CMP ERRS
  (10) 031316  013746  010620                                                           MOV     EVTTMP,-(SP)
   (9) 031322  013746  010616                                                           MOV     EVTBCT,-(SP)
   (8) 031326  013746  010614                                                           MOV     EVTADD,-(SP)
   (7) 031332  012746  016555                                                           MOV     #EVTF4,-(SP)
   (6) 031336  012746  000004                                                           MOV     #4,-(SP)
   (3) 031342  010600                                                                   MOV     SP,R0
   (4) 031344  104416                                                                   TRAP    C$PNTS
   (4) 031346  062706  000012                                                           ADD     #12,SP
6496 031352  000137  030642                    JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY
6497
6498 031356                             RPTDLE:
6499 031356  012237  010614             RPTDCK: MOV     (R2)+,EVTADD    ;STORE MSG ADDR FOR PRINT
6500 031362  012237  010616                     MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT
6501 031366  012237  010620                     MOV     (R2)+,EVTTMP    ;STORE BYTE COUNT COMP
6502 031372                                     PRINTS  #EVTF4A,EVTADD,EVTBCT,EVTTMP    ;PRINT ADDR,RXBYTES,CMPBYTES.
  (10) 031372  013746  010620                                                           MOV     EVTTMP,-(SP)
   (9) 031376  013746  010616                                                           MOV     EVTBCT,-(SP)
   (8) 031402  013746  010614                                                           MOV     EVTADD,-(SP)
   (7) 031406  012746  016657                                                           MOV     #EVTF4A,-(SP)
   (6) 031412  012746  000004                                                           MOV     #4,-(SP)
   (3) 031416  010600                                                                   MOV     SP,R0
   (4) 031420  104416                                                                   TRAP    C$PNTS
   (4) 031422  062706  000012                                                           ADD     #12,SP
6503
6504 031426  000137  030642                    JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY
6505
6506 031432  012604                     ENDEVT: MOV     (SP)+,R4        ;RESTORE R4,R3,R2
6507 031434  012603                             MOV     (SP)+,R3
6508 031436  012602                             MOV     (SP)+,R2
6509 031440  000207                             RTS     PC              ;RETURN TO CALLING ROUTINE
6510
6511
6512                                    ;REPORT MODEM STATUS SUBROUTINE
6513                                    ;       PART OF STATISICAL REPORTING (DUMPING EVENT LOG)
6514
6515 031442                             RPTMSB: PRINTS  #EVMOHD                 ;PRINT MODEM STATUS HEADER
   (7) 031442  012746  017143                                                           MOV     #EVMOHD,-(SP)
   (6) 031446  012746  000001                                                           MOV     #1,-(SP)
   (3) 031452  010600                                                                   MOV     SP,R0
   (4) 031454  104416                                                                   TRAP    C$PNTS
   (4) 031456  062706  000004                                                           ADD     #4,SP
6516 031462  012704  010460                     MOV     #MOBITS,R4      ;MAKE R4 A POINTER TO MODEM SIG. BIT DEF. TABLE
6517 031466  012705  010476                     MOV     #MOMSGS,R5      ;MAKE R5 A POINTER TO MODEM MSG. POSITION TABLE
6518 031472  005714                     6$:    TST     (R4)            ;SEE IF BIT AVAIABLE FROM DEVICE
6519 031474  001004                             BNE     7$              ;BR IF THAT MODEM SIG. AVAIABLE
6520 031476  112735  000130                     MOVB    #'X,@(R5)+      ;ELSE PUT 'X' IN REPORT IF SIGNAL NOT AVAILABLE
6521 031502  005724                             TST     (R4)+           ;BUMP R4 TO POINT TO NEXT BIT DEFINITION
6522 031504  000407                             BR      9$              ;GO SEE IF CHECKED ALL MODEM SIGNALS
6523 031506  032403                     7$:    BIT     (R4)+,R3        ;IF THERE, SEE IF THAT BIT IN DEVICE'S ENTRY=1
6524 031510  001403                             BEQ     8$              ;BR IF BIT (SIGNAL) VALUE =0
6525 031512  112735  000061                     MOVB    #'1,@(R5)+      ;IF=1, PUT '1' IN REPORT MESSAGE
```

```
 6526  031516  000402                        BR      9$              ;GO SEE IF ALL MODEM SIGNALS CHECKED
 6527  031520  112735  000060        8$:     MOVB    #'0,@(R5)+      ;IF BIT(SIGNAL)=0, PUT '0' IN REPORT MESSAGE
 6528  031524  020427  010476        9$:     CMP     R4,#MOBITE      ;SEE IF ALL BITS(SIGNALS) CHECKED
 6529  031530  002760                        BLT     6$              ;LOOP UNTIL ALL SIGNALS(BITS) CHECKED
 6530  031532                                PRINTS  #EVMOST         ;THEN PRINT MODEM SIGNAL VALUE MESSAGE
   (7) 031532  012746  017223                                                    MOV     #EVMOST,-(SP)
   (6) 031536  012746  000001                                                    MOV     #1,-(SP)
   (3) 031542  010600                                                            MOV     SP,R0
   (4) 031544  104416                                                            TRAP    C$PNTS
   (4) 031546  062706  000004                                                    ADD     #4,SP
 6531  031552  000207                        RTS     PC              ;RETURN TO EVENT DECODING
 6532
 6533
```

```
                                .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
(8)     031562  010246                                          MOV     R2,-(SP)
(7)     031564  012746  016242                                  MOV     #BASM1,-(SP)
(6)     031570  012746  000002                                  MOV     #2,-(SP)
(3)     031574  010600                                          MOV     SP,R0
(4)     031576  104417                                          TRAP    C$PNTF
(4)     031600  062706  000006                                  ADD     #6,SP
031604  005737  007406  DUM3:   TST     BYTBIT          ;IS THIS BYTE OR WORD
031610  001416          BEQ     DUM1            ;BR IF WORD
031612  112237  007426  MOVB    (R2)+,TEMP      ;MOV BYTE TO TEMP
031616                  PRINTF  #BASM3,<B,TEMP> ;PRINT BYTE
(8)     031616  005046                                          CLR     -(SP)
(8)     031620  153716  007426                                  BISB    TEMP,(SP)
(7)     031624  012746  016224                                  MOV     #BASM3,-(SP)
(6)     031630  012746  000002                                  MOV     #2,-(SP)
(3)     031634  010600                                          MOV     SP,R0
(4)     031636  104417                                          TRAP    C$PNTF
(4)     031640  062706  000006                                  ADD     #6,SP
031644  000411          BR      DUM2
031646          DUM1:   PRINTF  #BASM2,(R2)+    ;PRINT WORD
(8)     031646  012246                                          MOV     (R2)+,-(SP)
(7)     031650  012746  016233                                  MOV     #BASM2,-(SP)
(6)     031654  012746  000002                                  MOV     #2,-(SP)
(3)     031660  010600                                          MOV     SP,R0
(4)     031662  104417                                          TRAP    C$PNTF
(4)     031664  062706  000006                                  ADD     #6,SP
031670  020237  007404  DUM2:   CMP     R2,ENADD        ;COMPARE FOR LAST ADD
```

```
 6572  031674  003005                        BGT     DUMEX           ;IF DONE EXIT
 6573  031676  005203                        INC     R3              ;ELSE BUMP R3
 6574  031700  022703  000010                CMP     #8.,R3          ;HAVE WE PRINTED 8 ACCROSS
 6575  031704  001725                        BEQ     DUM4            ;IF SO GO BACK TO 4
 6576  031706  000736                        BR      DUM3            ;ELSE GO BACK AND PRINT ANOTHER
 6577                                                                ;BYTE OR WORD
 6578  031710  000207              DUMEX:    RTS     PC              ;RETURN TO CALLER
 6579
```

```
6581                                    .SBTTL          UPDATE TOTAL CHAR. COUNT SUBROUTINE
6582
6583                                    ;++
6584                                    ; FUNCTIONAL DESCRIPTION:
6585                                    ;               UPDATES TOTAL CHAR. COUNT TOTCC BASED ON CURCC.
6586                                    ;               LAST MESSAGE IS TRUNCATED TO FIT INTO THE
6587                                    ;               BUFFER IF TOTAL CHAR. COUNT EXCEEDS 'BUFLIM'' A MESSAGE
6588                                    ;               IS PRINTED TELLING THE OPERATOR THE TRUNCATION OCCURED.
6589                                    ;
6590                                    ; INPUTS:
6591                                    ;               CURCC=  CHAR. COUNT OF MESSAGE BEING ADDED
6592                                    ;               TOTCC=  TOTAL CHAR COUNT OF BUFFER ITS BEING ADDED TO
6593                                    ;
6594                                    ; OUTPUTS:
6595                                    ;               MESSAGE TO OPERATOR IF MESSAGE TRUNCATED TO FIT
6596                                    ;
6597                                    ; FUNCTIONAL SIDE EFFECTS:
6598                                    ;               LOCATION ''TEMP'' USED FOR CALCULATIONS
6599                                    ; CALLING SEQUENCE:
6600                                    ;               JSR     PC,ADCC         ;UPDATED TOTAL CHAR. COUNT
6601                                    ;--
6602
6603
6604  031712  063737  007412  007422  ADDCC:  ADD     CURCC,TOTCC     ;ADD CURRENT TO TOTAL
6605  031720  022737  001000  007422          CMP     #BUFLIM,TOTCC   ; COMPARE TO 'BUFLIM''
6606  031726  103027                          BHIS    ADDC1           ;IF NOT MORE THEN 'BUFLIM'' EXIT
6607
6608                                          ; PRINT MESSAGE AND TRUNCATE COUNT
6609
6610  031730                                  PRINTF  #MSGTRU
 (7)  031730  012746  015272                                                    MOV     #MSGTRU,-(SP)
 (6)  031734  012746  000001                                                    MOV     #1,-(SP)
 (3)  031740  010600                                                            MOV     SP,R0
 (4)  031742  104417                                                            TRAP    C$PNTF
 (4)  031744  062706  000004                                                    ADD     #4,SP
6611  031750  163737  007412  007422          SUB     CURCC,TOTCC     ;SUB CURRENT FROM TOTAL
6612  031756  012737  001000  007426          MOV     #BUFLIM,TEMP    ;MOV 'BUFLIM'' TO TEMP
6613  031764  163737  007422  007426          SUB     TOTCC,TEMP      ;SUB TOTAL FROM 'BUFLIM''
6614  031772  013737  007426  007412          MOV     TEMP,CURCC      ;AND ESTABLISH NEW CURRENT
6615  032000  063737  007412  007422          ADD     CURCC,TOTCC     ;ADD ''ADJUSTED CURRENT'' TO TOTAL CHAR. CNT.
6616  032006  000207                  ADDC1:  RTS     PC              ;RETURN TO CALLER
6617
```

```
                        .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
```

```
6727
6728  032134                                  FACSIMILE:
6729  032134  010146                                MOV     R1,-(SP)                ;SAVE R1
6730  032136  005001                                CLR     R1                      ;INIT R1
6731  032140  116161  003562  005562  10$:         MOVB    TXBUF(R1),CMPBUF(R1)    ;COPY TX BUFFER TO EXPECTED BUFFER
6732  032146  005201                                INC     R1                      ;BUMP INDEX
6733  032150  020127  001000                       CMP     R1,#BUFLIM              ;ALL DATA COPIED ?
6734  032154  001371                                BNE     10$                     ;NO,BRANCH
6735
6736  032156  012701  000017          20$:         MOV     #MSGLIM,R1              ;MESSAGE LIMIT
6737  032162  006301                                ASL     R1                      ;MULTIPLY BY 2
6738  032164  006301                                ASL     R1                      ;MULTIPLY BY 2
6739  032166  012737  006562  007336                MOV     #PTRTAB,CMPPTR          ;TOP OF POINTER TABLE
6740  032174  060137  007336                       ADD     R1,CMPPTR               ;START OF EXPECTED POINTER TABLE
6741  032200  005001                                CLR     R1                      ;INIT R1
6742
6743                                                ;SET UP WHILE - DO LOOP
6744  032202  012737  006562  007334                MOV     #PTRTAB,TXPTR           ;TX POINTER NOW AT TOP OF TABLE
6745  032210  012737  005562  007344                MOV     #CMPBUF,CCURAD          ;TRANSFER ADDRESS OF 1ST MESSAGE
6746  032216  062737  000002  007334                ADD     #2,TXPTR                ;BUMP POINTER
6747  032224  017737  155104  007342                MOV     @TXPTR,CTOTCC           ;BYTE COUNTER 1ST MESSAGE
6748  032232  005037  007340                       CLR     CMPTOT                  ;INIT EXPECTED MESSAGE COUNT
6749
6750                                                ;WHILE TX MESSAGE TOTAL <> EXPECTED MESSAGE TOTAL DO
6751  032236  023737  007354  007340  30$:         CMP     TXMTOT,CMPTOT           ;ALL MESSAGES COPIED ?
6752  032244  001430                                BEQ     40$                     ;YES,BRANCH
6753  032246  013777  007344  155062                MOV     CCURAD,@CMPPTR          ;TRANSFER ADDRESS OF MESSAGE
6754  032254  062737  000002  007336                ADD     #2,CMPPTR               ;BUMP POINTER
6755  032262  013777  007342  155046                MOV     CTOTCC,@CMPPTR          ;BYTE COUNT OF MESSAGE
6756  032270  062737  000004  007334                ADD     #4,TXPTR                ;BUMP TX MESSAGE POINTER
6757  032276  063737  007342  007344                ADD     CTOTCC,CCURAD           ;CALC. TRANSFER ADDRESS
6758  032304  017737  155024  007342                MOV     @TXPTR,CTOTCC           ;BYTE COUNT NEXT MESSAGE
6759  032312  062737  000002  007336                ADD     #2,CMPPTR               ;BUMP POINTER
6760  032320  005237  007340                       INC     CMPTOT                  ;INCREMENT MESSAGE COUNT
6761  032324  000744                                BR      30$                     ;DO IT AGAIN
6762                                                ;END WHILE - DO
6763  032326  013737  007356  007342  40$:         MOV     TTOTCC,CTOTCC           ;COPY TOTAL CHARACTER COUNT
6764
6765                                                ;END ROUTINE
6766  032334  012601                                MOV     (SP)+,R1                ;RESTORE R1
6767  032336  000207                                RTS     PC                      ;RETURN
6768
6769
```

```
                                .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  010650          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  010652                  MOV     DEV2,R2         ;GET MAINTENANCE LOOP TYPE
032360  006302                          ASL     R2
032362  012737  014636  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  014635  007434          MOV     #LP0,TEMP3      ;IF NO LOOP THEN DON'T PRINT ''/LOOP=''
032402  016237  003532  007430  10$:    MOV     LOOPS(R2),TEMP1 ;GET ADDRESS OF LOOP-IN-ASCII
032410  013737  010654  007432          MOV     DEV3,TEMP2      ;GET NUMBER OF PASSES
032416                          PRINTS  #SHF0,TEMP,TEMP3,TEMP1,TEMP2
 (11)   032416  013746  007432                                                  MOV     TEMP2,-(SP)
 (10)   032422  013746  007430                                                  MOV     TEMP1,-(SP)
 (9)    032426  013746  007434                                                  MOV     TEMP3,-(SP)
 (8)    032432  013746  007426                                                  MOV     TEMP,-(SP)
 (7)    032436  012746  015355                                                  MOV     #SHF0,-(SP)
 (6)    032442  012746  000005                                                  MOV     #5,-(SP)
 (3)    032446  010600                                                          MOV     SP,R0
 (4)    032450  104416                                                          TRAP    C$PNTS
 (4)    032452  062706  000014                                                  ADD     #14,SP

032456  005002                          CLR     R2              ;NOW SET UP FOR QUALIFIERS IN ASCII
032460  012737  014715  007426          MOV     #PST,TEMP
032466  032737  000001  010656          BIT     #STATB,DEV4     ;SEE IF /STATUS OR /NOSTATUS
032474  001003                          BNE     1$              ;BR IF /STATUS
032476  012737  014713  007426          MOV     #PNST,TEMP
032504  012737  014726  007430  1$:     MOV     #PCK,TEMP1
032512  032737  000002  010656          BIT     #DATCKB,DEV4    ;SEE IF /CHECK OR /NOCHECK
032520  001003                          BNE     2$              ;BR IF /CHECK
032522  012737  014724  007430          MOV     #PNCK,TEMP1
032530  012737  014736  007432  2$:     MOV     #PEC,TEMP2
032536  032737  000004  010656          BIT     #ECHOB,DEV4     ;SEE IF /ECHO OR /NOECHO
032544  001003                          BNE     3$              ;BR IF /ECHO
032546  012737  014734  007432          MOV     #PNEC,TEMP2
```

```
 6818  032554  012737  014745  007440  3$:     MOV     #PMS,TEMP5      ;ASSUME /MODEM ;REV B EC
 6819  032562  032737  000010  010656          BIT     #MOCHK,DEV4     ;MODEM CHECK ? ;REV B EC
 6820  032570  001003                          BNE     5$              ;YES,BRANCH ;REV B EC
 6821  032572  012737  014743  007440          MOV     #PNMS,TEMP5     ;''/NOMODEM'' MESSAGE ;REV B EC
 6822
 6840
 6841  032600                          5$:     PRINTS  #SHF1,TEMP,TEMP1,TEMP2,TEMP5    ;,TEMP3,TEMP4 **;SEE NOTE ABOVE
  (11) 032600  013746  007440                                                                  MOV     TEMP5,-(SP)
  (10) 032604  013746  007432                                                                  MOV     TEMP2,-(SP)
  (9)  032610  013746  007430                                                                  MOV     TEMP1,-(SP)
  (8)  032614  013746  007426                                                                  MOV     TEMP,-(SP)
  (7)  032620  012746  015413                                                                  MOV     #SHF1,-(SP)
  (6)  032624  012746  000005                                                                  MOV     #5,-(SP)
  (3)  032630  010600                                                                          MOV     SP,R0
  (4)  032632  104416                                                                          TRAP    C$PNTS
  (4)  032634  062706  000014                                                                  ADD     #14,SP
 6842  032640  000207                          RTS     PC              ;RETURN
 6843
 6844
```

```
 6846                                   .SBTTL          TRAVERSE COMMAND LINE SUBROUTINES
 6847
 6848                                   ;++
 6849                                   ;       P$TRV SUBROUITNE
 6850                                   ;
 6851                                   ;PARSE THE COMMAND LINE SUBROUTINE
 6852                                   ;TAKE ACTIONS (VIA ACTION TREE) AS PARSING LINE
 6853                                   ;PARSING DIRECTIONS FROM ''CLI PARSING NODES''
 6854                                   ;   REGS USED:
 6855                                   ;
 6856                                   ;       R1,R5=SCRATCH                           P$NUM=NUMERIC CODE FROM DATA
 6857                                   ;       R2=ACTION CODE PARAMETER FROM TREE
 6858                                   ;       R3=PARSE TREE POINTER
 6859                                   ;       R4=INPUT STRING POINTER
 6860                                   ; CALLING SEQUENCE:
 6861                                   ;       JSR     PC,P$TRV
 6862                                   ;--
 6863
 6864  032642                           P$TRV:
 6865  032642  013704  003544                   MOV     P$BUFA,R4
 6866  032646  013703  003546                   MOV     P$TREE,R3
 6867  032652  105714                   P$TR5:  TSTB    (R4)                    ;SEE IF ANY CHARS LEFT IN INPUT STRING
 6868  032654  001441                           BEQ     P$EXIT                  ;BR IF NO
 6869  032656  121327  000013                   CMPB    (R3),#11.               ;SEE IF SPECIAL CLI CHAR CODE OR ASCII
 6870  032662  003023                           BGT     20$                     ;BR IF REGULAR ASCII CHAR.
 6871  032664  111305                           MOVB    (R3),R5                 ;GET SPECIAL CHAR CODE INTO R5
 6872  032666  006305                           ASL     R5
 6873  032670  016505  032704                   MOV     10$(R5),R5              ;BUILD TRAVERSE ROUTINE ADDRESS
 6874  032674  062705  032704                   ADD     #10$,R5
 6875  032700  004715                           JSR     PC,(R5)                 ;JSR TO SPECIAL CLI TRAVERSE ROUTINE
 6876  032702  000763                           BR      P$TR5                   ;GO SEE IF MORE OF STRING LEFT
 6877
 6878                                           ;TRAVERSE TABLE FOR 'CLI FUNTIONS'
 6879  032704  000114                   10$:    .WORD   TRVERR-10$              ;TAKE ERROR ACTION
 6880  032706  000134                           .WORD   TRVEXI-10$              ;TAKE EXIT ACTION
 6881  032710  000152                           .WORD   TRVBR-10$               ;TAKE BRANCH ACTION
 6882  032712  000162                           .WORD   TRVBIF-10$              ;TEST P$GDBD & TAKE BRANCH
 6883  032714  000204                           .WORD   TRVSPA-10$              ;SKIP SPACES OR TABS IN CMD LINE
 6884  032716  000270                           .WORD   TRVNUM-10$              ;TRAVERSE NUMERIC FIELD
 6885  032720  000604                           .WORD   TRVALP-10$              ;TRAVERSE ALPHABETICS
 6886  032722  000650                           .WORD   TRVALN-10$              ;TRAVERSE ALPHANUMERICS
 6887  032724  000270                           .WORD   TRVOCT-10$              ;SAME AS TRVNUM
 6888  032726  000256                           .WORD   TRVDEC-10$              ;SAME AS CLINUM BUT DECIMAL
 6889  032730  000736                           .WORD   TRVSTR-10$              ;FIND ASCIZ MATCH IN CMD LINE
 6890
 6891                                   ;NOT A SPECIAL CODE
 6892
 6893  032732  121314                   20$:    CMPB    (R3),(R4)               ;SEE IF FIRST CHAR OF STRING IS A MATCH
 6894  032734  001403                           BEQ     22$                     ;BR IF A MATCH
 6895  032736  004737  033002                   JSR     PC,TRVBRC               ;IF NOT A MATCH, GO TAKE MISS BRANCH
 6896  032742  000743                           BR      P$TR5                   ; THEN GO BACK PT'G TO MISS NODE
 6897  032744  004737  032762           22$:    JSR     PC,TRVACT               ;IF A MATCH, GO DO ACTION DEFINED BY
 6898  032750  062703  000004                   ADD     #4,R3                   ; ACTION CODE IN CLI NODE, THEN
 6899                                                                           ; ADJUST PTR TO NEXT CLI NODE
 6900  032754  005204                           INC     R4                      ;ADJUST BUF PTR TO NEXT CHAR IF MATCH
 6901  032756  000735                           BR      P$TR5
```

```
6902
6903  032760  000207                  P$EXIT: RTS     PC                      ;RETURN FROM PARSER
6904
6905                                  ;-------------------------------------------------------------------------
6906
6907                                  ;GOTO USER ACTION ROUTINE
6908  032762  116302  000001          TRVACT: MOVB    1(R3),R2                ;GET ACTION CODE FROM CLI NODE
6909  032766  042702  177400                  BIC     #177400,R2              ;CLEAR ANY SIGN EXTENSION
6910  032772  013705  003550                  MOV     P$ACT,R5                ;GET ADDRESS OF CLI ACTION ROUTINE
6911  032776  004715                          JSR     PC,(R5)                 ;GO DO ACTION DEFINED BY CODE
6912  033000  000207                          RTS     PC                      ;RETURN TO CALLING CODE
6913
6914                                  ;TAKE BRANCH IN TREE
6915  033002  016305  000002          TRVBRC: MOV     2(R3),R5                ;GET BRANCH DISPLACEMENT FROM TREE
6916  033006  060503                          ADD     R5,R3                   ; AND POINT R3 TO THE 'MISS' NODE
6917  033010  000207                          RTS     PC                      ; RETURN TO P$TRV
6918
6919                                  ;NO BRANCH TAKEN
6920  033012  062703  000004          TRVNOB: ADD     #4,R3                   ;THINGS OK, UPDATE R3 TO POINT TO NEXT
6921  033016  000207                          RTS     PC                      ; NODE AND RETURN TO P$TRV
6922
6923                                  ;-------------------------------------------------------------------------
6924  033020  004737  032762          TRVERR: JSR     PC,TRVACT               ;TAKE ERROR ACTION
6925  033024  112737  177777  003561          MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
6926  033032  005726                          TST     (SP)+                   ;GET RID OF "JSR PUSH TO TRVERR"
6927  033034  000137  032760                  JMP     P$EXIT                  ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
6928
6929  033040  004737  032762          TRVEXI: JSR     PC,TRVACT               ;TAKE EXIT ACTION
6930  033044  105037  003561                  CLRB    P$GDBD                  ;SET GOOD/BAD FLAG TO "SUCCESS (0)"
6931  033050  005726                          TST     (SP)+                   ;GET RID OF "JSR PUSH TO TRVEXI"
6932  033052  000137  032760                  JMP     P$EXIT                  ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
6933
6934  033056  004737  032762          TRVBR:  JSR     PC,TRVACT               ;GO TAKE BRANCH ACTION
6935  033062  000137  033002                  JMP     TRVBRC
6936
6937  033066  004737  032762          TRVBIF: JSR     PC,TRVACT
6938  033072  105737  003561                  TSTB    P$GDBD                  ;SEE IF P$GDBD SET OR CLEARED BY ACTION
6939  033076  001402                          BEQ     1$                      ;IF CLEAR FALL THRU TO NEXT NODE
6940  033100  000137  033002                  JMP     TRVBRC                  ;ELSE TAKE THE "MISS" BRANCH
6941  033104  000137  033012          1$:     JMP     TRVNOB                  ;JUST UPDATE TO NEXT NODE IF THINGS OK
6942
6943  033110  005005                  TRVSPA: CLR     R5                      ;CLEAR "SPACE OR TAB FOUND" FLAG
6944  033112  121427  000011          1$:     CMPB    (R4),#11                ;SEE IF CHAR. IN CMD LINE= TAB
6945  033116  001003                          BNE     2$                      ;BR IF NO, NOT A TAB
6946  033120  005204                          INC     R4                      ;INC INPUT STRING POINTER
6947  033122  005205                          INC     R5                      ;INDICATE A TAB FOUND
6948  033124  000772                          BR      1$                      ;GO CHECK NEXT CHAR
6949
6950  033126  121427  000040          2$:     CMPB    (R4),#40                ;SEE IF CHAR. IN CMD LINE= SPACE
6951  033132  001003                          BNE     10$                     ;BR IF NO, NON-SPACE OR NON-TAB CHAR.
6952  033134  005204                          INC     R4                      ;INC INPUT STRING POINTER
6953  033136  005205                          INC     R5                      ;INDICATE A SPACE FOUND
6954  033140  000764                          BR      1$                      ;GO CHECK NEXT CHAR
6955  033142  005705                  10$:    TST     R5                      ;SEE IF ANY SPACES OR TABS FOUND
6956  033144  001404                          BEQ     15$                     ;BR IF NO, TAKE NO ACTION
6957  033146  004737  032762                  JSR     PC,TRVACT               ;GO TAKE ACTION IF ANY FOUND
```

```
 6958  033152  000137  033012                 JMP     TRVNOB                  ;JUST GO UPDATE R3 TO NEXT NODE IF OK
 6959  033156  000137  033002          15$:   JMP     TRVBRC                  ;TAKE BRANCH (MISS) IF NONE FOUND
 6960
 6961
 6962  033162  012737  000012  003556 TRVDEC: MOV     #10.,P$RADX             ;USE DECIMAL AS RADIX AND ASSUME +
 6963  033170  000137  033202                 JMP     TRVNMA
 6964  033174                         TRVOCT: ;(SAME AS TRVNUM SINCE DEFAULT RADIX IS OCTAL)
 6965  033174  012737  000010  003556 TRVNUM: MOV     #8.,P$RADX              ;USE OCTAL AS RADIX AND ASSUME +
 6966  033202  005005                 TRVNMA: CLR     R5                      ;CLEAR DIGIT COUNTER
 6967  033204  121427  000053                 CMPB    (R4),#'+                ;SEE IF THERE'S A + SIGN THERE
 6968  033210  001001                         BNE     10$                     ; BR IF NO
 6969  033212  000406                         BR      11$                     ; ELSE P$RADX ALREADY SAYS +, JUST BR
 6970  033214  121427  000055          10$:   CMPB    (R4),#'-                ;SEE IF THERE'S A - SIGN THERE
 6971  033220  001004                         BNE     1$                      ; BR IF NO
 6972  033222  112737  177777  003557         MOVB    #-1,P$RADX+1            ;SET 'MINUS FLAG' (HI BYTE OF P$RADX)
 6973  033230  005204                 11$:   INC     R4                      ;BUMP R4 TO POINT TO FIRST CHAR
 6974
 6975  033232  121427  000060          1$:    CMPB    (R4),#60                ;SEE IF CHAR. LESS THAN A '0'
 6976  033236  002434                         BLT     2$                      ;BR IF YES (NOT NUMERIC)
 6977  033240  121427  000067                 CMPB    (R4),#67                ;SEE IF CHAR. GREATER THAN A '7'
 6978  033244  003426                         BLE     13$                     ; BR IF YES
 6979  033246  123727  003556  000012         CMPB    P$RADX,#10.             ;SEE IF IN DECIMAL MODE
 6980  033254  001417                         BEQ     12$                     ; BR IF YES (CAN USE HIGHER LIMIT)
 6981  033256  121427  000071                 CMPB    (R4),#71                ;SEE IF DIGIT WAS A 8 OR 9
 6982  033262  003022                         BGT     2$                      ;BR IF NON-NUMERIC
 6983  033264                                 PRINTF  #CLIBRX                 ;ELSE WAS A 8 OR 9 WHEN IN OCTAL RADIX
   (7) 033264  012746  012607                                                                 MOV     #CLIBRX,-(SP)
   (6) 033270  012746  000001                                                                 MOV     #1,-(SP)
   (3) 033274  010600                                                                         MOV     SP,R0
   (4) 033276  104417                                                                         TRAP    C$PNTF
   (4) 033300  062706  000004                                                                 ADD     #4,SP
 6984  033304  112737  177777  003561         MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
 6985  033312  000474                         BR      5$                      ; PRINT ERROR AND TAKE MISS
 6986
 6987  033314  121427  000071          12$:   CMPB    (R4),#71                ;SEE IF CHAR. GREATER THAN A '9'
 6988  033320  003003                         BGT     2$                      ;BR IF YES (NOT NUMERIC)
 6989  033322  005204                 13$:   INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR.
 6990  033324  005205                         INC     R5                      ;INDICATE A NUMERIC FOUND
 6991  033326  000741                         BR      1$                      ;GO LOOK AT NEXT CHAR.
 6992
 6993  033330  005705                 2$:    TST     R5                      ;SEE IF FOUND ANY NUMERICS
 6994  033332  001464                         BEQ     5$                      ;BR IF NO, TAKE 'MISS' BRANCH
 6995  033334  010401                         MOV     R4,R1                   ;GET POINTER TO START OF NUMERIC STRING
 6996  033336  160501                         SUB     R5,R1
 6997  033340  005037  003554                 CLR     P$NUM                   ;CLEAR LOC. WHERE VALUE WILL BE STORED
 6998  033344  112102                 3$:    MOVB    (R1)+,R2                ;GET ASCII CHAR AND CONVERT IT TO A #
 6999  033346  162702  000060                 SUB     #60,R2
 7000  033352  006337  003554                 ASL     P$NUM                   ;SHIFT CURRENT VALUE TO MAKE ROOM
 7001  033356  103437                         BCS     7$                      ;ERROR IF NUMBER TOO BIG
 7002  033360  013737  003554  003552         MOV     P$NUM,P$CNT             ;SAVE FOR LATER IN CASE DECIMAL RADIX
 7003  033366  006337  003554                 ASL     P$NUM
 7004  033372  103431                         BCS     7$                      ;ERROR IF NUMBER TOO BIG
 7005  033374  006337  003554                 ASL     P$NUM
 7006  033400  103426                         BCS     7$                      ;ERROR IF NUMBER TOO BIG
 7007  033402  123727  003556  000012         CMPB    P$RADX,#10.             ;SEE IF DECIMAL RADIX
 7008  033410  001004                         BNE     4$                      ;BR IF NOT EQUAL
```

```
 7009  033412  063737  003552  003554          ADD     P$CNT,P$NUM
 7010  033420  103416                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
 7011  033422  060237  003554          4$:     ADD     R2,P$NUM
 7012  033426  103413                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
 7013  033430  005305                          DEC     R5
 7014  033432  001344                          BNE     3$
 7015  033434  105737  003557                  TSTB    P$RADX+1                ;SEE IF NUM WAS PRECEDED BY A - SIGN
 7016  033440  001402                          BEQ     15$                     ; BR IF NO
 7017  033442  005437  003554                  NEG     P$NUM                   ; ELSE NEGATE THE NUMBER BEFORE LEAVING
 7018  033446  004737  032762          15$:    JSR     PC,TRVACT               ;SINCE NUMERIC FOUND, GO TAKE ACTION
 7019  033452  000137  033012                  JMP     TRVNOB                  ;GO POINT R3 TO NEXT NODE
 7020
 7021  033456                          7$:     PRINTF  #CLINBG                 ;PRINT NUMBER TOO BIG ERROR
   (7) 033456  012746  012565                                                          MOV     #CLINBG,-(SP)
   (6) 033462  012746  000001                                                          MOV     #1,-(SP)
   (3) 033466  010600                                                                  MOV     SP,R0
   (4) 033470  104417                                                                  TRAP    C$PNTF
   (4) 033472  062706  000004                                                          ADD     #4,SP
 7022  033476  112737  177777  003561          MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
 7023  033504  000137  033002          5$:     JMP     TRVBRC                  ;TAKE 'MISS' BRANCH
 7024
 7025
 7026  033510  005005                  TRVALP: CLR     R5                      ;CLEAR ALPHA FOUND FLAG
 7027  033512  121427  000101          1$:     CMPB    (R4),#101               ;SEE IF CHAR. LESS THAN A 'A'
 7028  033516  002406                          BLT     2$                      ;BR IF YES (NOT ALPHA)
 7029  033520  121427  000132                  CMPB    (R4),#132               ;SEE IF CHAR. GREATER THAN A 'Z'
 7030  033524  003003                          BGT     2$                      ;BR IF YES (NOT ALPHA)
 7031  033526  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR
 7032  033530  005205                          INC     R5                      ;INDICATE AN ALPHA WAS FOUND
 7033  033532  000767                          BR      1$                      ;GO LOOK AT NEXT CHAR.
 7034  033534  005705                  2$:     TST     R5                      ;SEE IF ANY ALPHA'S WERE FOUND
 7035  033536  001404                          BEQ     3$                      ;BR IF NO
 7036  033540  004737  032762                  JSR     PC,TRVACT               ;IF ANY FOUND TAKE ACTION
 7037  033544  000137  033012                  JMP     TRVNOB                  ;THEN UPDATE R3 TO NEXT NODE -NO BRANCH
 7038  033550  000137  033002          3$:     JMP     TRVBRC                  ;NONE FOUND, TAKE MISS BRANCH
 7039
 7040  033554  005005                  TRVALN: CLR     R5                      ;CLEAR ALPHANUM FOUND FLAG
 7041  033556  121427  000060          10$:    CMPB    (R4),#60                ;SEE IF CHAR. LESS THAN A '0'
 7042  033562  002417                          BLT     2$                      ;BR IF YES (NOT NUMERIC OR ALPHA)
 7043  033564  121427  000072                  CMPB    (R4),#72                ;SEE IF CHAR. GREATER THAN A '9'
 7044  033570  003003                          BGT     1$                      ;BR IF YES (NOT NUMERIC)
 7045  033572  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR.
 7046  033574  005205                          INC     R5                      ;INDICATE A NUMERIC FOUND
 7047  033576  000767                          BR      10$                     ;GO LOOK AT NEXT CHAR.
 7048  033600  121427  000101          1$:     CMPB    (R4),#101               ;SEE IF CHAR. LESS THAN A 'A'
 7049  033604  002406                          BLT     2$                      ;BR IF YES (NOT ALPHA)
 7050  033606  121427  000132                  CMPB    (R4),#132               ;SEE IF CHAR. GREATER THAN A 'Z'
 7051  033612  003003                          BGT     2$                      ;BR IF YES (NOT ALPHA)
 7052  033614  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR
 7053  033616  005205                          INC     R5                      ;INDICATE AN ALPHA FOUND
 7054  033620  000756                          BR      10$                     ;GO LOOK AT NEXT CHAR.
 7055  033622  005705                  2$:     TST     R5                      ;SEE IF ANY ALPHANUM'S WERE FOUND
 7056  033624  001404                          BEQ     3$                      ;BR IF NO
 7057  033626  004737  032762                  JSR     PC,TRVACT               ;IF ANY FOUND TAKE ACTION
 7058  033632  000137  033012                  JMP     TRVNOB                  ;THEN UPDATE R3 TO NEXT NODE -NO BRANCH
 7059  033636  000137  033002          3$:     JMP     TRVBRC                  ;NONE FOUND, TAKE MISS BRANCH
```

```
7060
7061
7062
7063  033642  010401              TRVSTR: MOV     R4,R1           ;POINT R1 TO CMD STRING
7064  033644  010305                      MOV     R3,R5
7065  033646  062705  000006              ADD     #6,R5           ;POUNT R5 TO MATCH STRING FROM CLI NODE
7066  033652  005037  003552              CLR     P$CNT           ;CLEAR CHAR MATCH COUNT
7067  033656  105715              2$:     TSTB    (R5)            ;SEE IF END OF MATCH STRING YET
7068  033660  001411                      BEQ     10$             ;BR IF YES
7069  033662  105711                      TSTB    (R1)            ;SEE IF END OF CMD LINE YET
7070  033664  001407                      BEQ     10$             ;BR IF YES
7071  033666  121115                      CMPB    (R1),(R5)       ;SEE IF CHARACTERS MATCH
7072  033670  001005                      BNE     10$             ;BR IF NO
7073  033672  005237  003552              INC     P$CNT           ;MATCH -INCREMENT MATCH COUNT
7074  033676  005201                      INC     R1              ;UPDATE STRING POINTERS
7075  033700  005205                      INC     R5
7076  033702  000765                      BR      2$              ;BR TO CONTINUE CHECKING CHARS.
7077
7078  033704  005737  003552      10$:    TST     P$CNT           ;WHEN DONE SEE IF ANY MATCHES FOUND
7079  033710  001406                      BEQ     15$             ;BR IF NO, GO TAKE THE MISS BRANCH
7080  033712  010104                      MOV     R1,R4           ;POINT CMD POINTER TO END OF STRING &
7081  033714  004737  032762              JSR     PC,TRVACT       ;IF A MATCH FOUND, GO DO MATCH ACTION
7082  033720  066303  000004              ADD     4(R3),R3        ;UPDATE R3 TO NEXT NODE (NO BRANCH)
7083  033724  000207                      RTS     PC              ; (NO RETURN THRU TRVNOB SINCE DIFFERNT
7084                                                              ;  DISPLACEMENT DUE TO MATCH STRING)
7085  033726  000137  033002      15$:    JMP     TRVBRC          ; GO TAKE BRANCH
7086
7087                                                              ; (PARSED OK), -1 IF ILL CMD.....
7088                              ;-----------------------------------------------------------------
7089
```

```
 7091                                   .SBTTL  REPORT CODING SECTION
 7092
 7093
 7094                                   ;++
 7095                                   ; THE REPORT CODING SECTION CONTAINS THE
 7096                                   ; 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
 7097                                   ;--
 7098
 7099  033732                                   BGNRPT
  (3)  033732                                                                    L$RPT::
 7100
 7112
 7113  033732  004737  027046                   JSR     PC,REPORT               ;CALL SUBROUTINE TO DUMP EVENT LOG
 7114                                                                            ; AND BASE TABLE
 7115
 7122
 7123
 7130
 7131  033736                                   ENDRPT
  (3)  033736                                                                    L10011:
  (3)  033736  104425                                                                   TRAP    C$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
```

```
7162                                    .SBTTL  INITIALIZE SECTION
7163
7164                                    ;++
7165                                    ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
7166                                    ; AT THE BEGINNING OF EACH PASS.
7167                                    ;--
7168
7169  033746                                    BGNINIT
 (3)  033746                                                                                L$INIT::
7170
7194                                    ;;ADDED BY EC
7195                                    ;;REVISION CHANGE B
7196  033746  005737  007464           TST     DCLFLG                  ;CLEANUP & EXIT ?
7197  033752  001403                   BEQ     INIT1                   ;NO,BRANCH
7198  033754  005037  007464           CLR     DCLFLG                  ;CLEAR FLAG
7199  033760                           DOCLN                           ;GO CLEANUP
 (3)  033760  104444                                                                       TRAP    C$DCLN
7200
7201  033762  012737  177777  007466  INIT1:  MOV     #-1,RESFLG              ;SET RESTART FLAG
7202  033770                           READEF  #EF.START               ;IF HERE CAUSE OF START,DO SOME INIT
 (3)  033770  012700  000040                                                               MOV     #EF.START,R0
 (3)  033774  104447                                                                       TRAP    C$REFG
7203  033776                           BCOMPLETE       START
 (2)  033776  103417                                                                       BCS     START
7204  034000                           READEF  #EF.RESTART             ;IF HERE CAUSE OF RESTART, DO SOME INIT
 (3)  034000  012700  000037                                                               MOV     #EF.RESTART,R0
 (3)  034004  104447                                                                       TRAP    C$REFG
7205  034006                           BCOMPLETE       RESTRT
 (2)  034006  103515                                                                       BCS     RESTRT
7206  034010                           READEF  #EF.CONTINUE            ;SEE IF WE'RE HERE CAUSE OF A CONTINUE
 (3)  034010  012700  000036                                                               MOV     #EF.CONTINUE,R0
 (3)  034014  104447                                                                       TRAP    C$REFG
7207  034016                           BNCOMPLETE      S1              ;BR IF NOT HERE CAUSE OF CONITNUE
 (2)  034016  103002                                                                       BCC     S1
7208  034020  000137  034530           JMP     ENDIT                   ;JMP IF HERE CAUSE OF A CONTINUE
7209  034024                   S1:     READEF  #EF.NEW                 ;SEE IF THIS IS A 'NEW PASS''
 (3)  034024  012700  000035                                                               MOV     #EF.NEW,R0
 (3)  034030  104447                                                                       TRAP    C$REFG
7210  034032                           BCOMPLETE       NEW             ;IF YES, BR AROUND LOGUNIT # SETUP
 (2)  034032  103523                                                                       BCS     NEW
7211  034034  000525                   BR      GETPRM
7212
7213  034036  005037  007466   START:  CLR     RESFLG                  ;CLEAR RESTART FLAG SINCE HERE ON START
7214  034042  005037  007526           CLR     CLKVEC                  ;CLEAR CLK VECTOR PTR. AS A FLAG IN
7215                                                                    ; NO CLOCK IS FOUND.
7216  034046  012702  007522           MOV     #CLKCSR,R2              ;SETUP R2 AS A PTR. TO CLOCK INFO BLOCK
7217  034052                           CLOCK   L,R1                    ;LOOK FOR A LINE CLOCK
 (3)  034052  012700  000114                                                               MOV     #'L,R0
 (3)  034056  104462                                                                       TRAP    C$CLCK
 (3)  034060  010001                                                                       MOV     R0,R1
7218  034062                           BNCOMPLETE      S2              ; IF NONE THERE GO LOOK FOR A P-CLOCK
 (2)  034062  103006                                                                       BCC     S2
7219  034064  004737  026204           JSR     PC,CLKSET               ; GO SET UP CLOCK INFO TABLE & CLK VEC.
7220  034070  012737  000100  007532   MOV     #LCLKEN,CLKEN           ;SETUP THE ENABLE LINE CLOCK DATA
7221  034076  000461                   BR      RESTRT
7222
```

```
7223  034100                          S2:     CLOCK   P,R1                    ;LOOK FOR A P-CLOCK SINCE NO LINE CLOCK
 (3)  034100  012700  000120                                                                MOV     #'P,R0
 (3)  034104  104462                                                                        TRAP    C$CLCK
 (3)  034106  010001                                                                        MOV     R0,R1
7224  034110                                  BNCOMPLETE      S3              ; IF NONE THERE GO SEE IF THIS IS LSI
 (2)  034110  103017                                                                        BCC     S3
7225  034112  004737  026204                  JSR     PC,CLKSET               ; ELSE GO SET UP CLOCK INFO & VECTOR
7226  034116  062737  000002  007522          ADD     #2,CLKCSR               ;POINT CLKCSR TO P-CLK COUNT SET REG.
7227  034124  012777  001600  153370          MOV     #PCLKCT,@CLKCSR         ;LOAD CLK SET REG. WITH COUNT VALUE
7228  034132  162737  000002  007522          SUB     #2,CLKCSR               ;POINT CLKCSR BAC TO P-CLK CSR
7229  034140  012737  000111  007532          MOV     #PCLKEN,CLKEN           ;SETUP THE ENABLE THE P-CLK DATA
7230  034146  000435                          BR      RESTRT
7231
7232  034150                          S3:     READBUS                         ;READ BUS TYPE TO SEE IF ON AN LSI
 (3)  034150  104407                                                                        TRAP    C$RDBU
7233  034152                                  BNCOMPLETE      S4              ;BR IF NOT, NO CHANCE OF A CLOCK
 (2)  034152  103021                                                                        BCC     S4
7234  034154  012737  000100  007526          MOV     #100,CLKVEC             ;LOAD 100 AS CLK VECTOR
7235  034162  005037  007524                  CLR     CLKBR                   ;LOAD 0 AS CLK INT. LEVEL
7236  034166  012737  007532  007522          MOV     #CLKEN,CLKCSR           ;KLUDGE UP THE CSR & ENABLE DATA LOCS
7237  034174                                  GMANID  L5060,CLKHZ,D,377,50.,60.,YES
 (3)  034174  104443                                                                        TRAP    C$GMAN
 (3)  034176  000406                                                                        BR      10000$
 (4)  034200  007530                                                                        .WORD   CLKHZ
 (5)  034202  000052                                                                        .WORD   T$CODE
 (5)  034204  014771                                                                        .WORD   L5060
 (5)  034206  000377                                                                        .WORD   377
 (5)  034210  000062                                                                        .WORD   T$LOLIM
 (5)  034212  000074                                                                        .WORD   T$HILIM
 (3)  034214                                                                    10000$:
7238  034214  000412                          BR      RESTRT
7239
7240  034216                          S4:     PRINTF  #NOCLK                  ;INFORM OPR. NO CLOCK, & EXIT INIT
 (7)  034216  012746  015102                                                                MOV     #NOCLK,-(SP)
 (6)  034222  012746  000001                                                                MOV     #1,-(SP)
 (3)  034226  010600                                                                        MOV     SP,R0
 (4)  034230  104417                                                                        TRAP    C$PNTF
 (4)  034232  062706  000004                                                                ADD     #4,SP
7241  034236                                  EXIT    INIT
 (3)  034236  104432                                                                        TRAP    C$EXIT
 (3)  034240  000404                                                                        .WORD   L10013-.
7242
7243  034242  005037  007534          RESTRT: CLR     TIMMIN                  ;CLEAR TIME SINCE START LOCATIONS
7244  034246  005037  007536                  CLR     TIMSEC
7245  034252  013737  007530  007540          MOV     CLKHZ,TIMTCK            ;LOAD TICKS/SEC
7246  034260  012702  007552                  MOV     #EVTLOG,R2              ;INIT EVENT TABLE TO ALL 1'S AFTER EACH
7247  034264  010237  007550                  MOV     R2,EVTPTR               ; START OR RES AND INIT TABLE POINTER
7248  034270  012722  177777          1$:     MOV     #-1,(R2)+
7249  034274  020227  010454                  CMP     R2,#EVTEND              ;SEE IF REACHED END OF TABLE
7250  034300  001373                          BNE     1$                      ;LOOP UNTIL DONE
7251
7252  034302  012737  177777  007460  NEW:    MOV     #-1,LOGUNT              ;INITIALIZE LOGICAL UNIT #
7253
7254  034310  005237  007460          GETPRM: INC     LOGUNT                  ;POINT TO NEXT LOGICAL UNIT
7255  034314  023737  007460  002012          CMP     LOGUNT,L$UNIT           ;SEE IF PAST MAX. LOG. UNIT #
7256  034322  002367                          BGE     NEW                     ;BR IF YES, AND START OVER
```

```
7257
7258  034324                                  GPHARD  LOGUNT,R1               ;GET THE P-TABLE FOR THIS LOG. UNIT
 (3)  034324  013700  007460                                                          MOV     LOGUNT,R0
 (3)  034330  104442                                                                  TRAP    C$GPHRD
 (3)  034332  010001                                                                  MOV     R0,R1
7259  034334                                  BNCOMPLETF      GETPRM          ;IF NO P-TABLE AVAIL., GO GET NEXT ONE
 (2)  034334  103365                                                                  BCC     GETPRM
7260
7261  034336  011137  007474                  MOV     (R1),FHDPLX             ;PUT FULL OR HALF DUPLEX ANSWER IN LOC.
7262
7273
7274                                          ;DEVICE DEPENDENT PART OF GETTING INFO FROM P-TABLE
7275
7276  034342  016137  000002  012360          MOV     2(R1),SEL0              ;STORE AWAY CSR ADDRESSES
7277  034350  016137  000002  012362          MOV     2(R1),BSEL1
7278  034356  005237  012362                  INC     BSEL1
7279  034362  016137  000002  012364          MOV     2(R1),SEL2
7280  034370  062737  000002  012364          ADD     #2,SEL2
7281  034376  016137  000002  012366          MOV     2(R1),BSEL3
7282  034404  062737  000003  012366          ADD     #3,BSEL3
7283  034412  016137  000002  012370          MOV     2(R1),SEL4
7284  034420  062737  000004  012370          ADD     #4,SEL4
7285  034426  016137  000002  012372          MOV     2(R1),BSEL5
7286  034434  062737  000005  012372          ADD     #5,BSEL5
7287  034442  016137  000002  012374          MOV     2(R1),SEL6
7288  034450  062737  000006  012374          ADD     #6,SEL6
7289  034456  016137  000002  012376          MOV     2(R1),BSEL7
7290  034464  062737  000007  012376          ADD     #7,BSEL7
7291
7292  034472  016137  000004  012400          MOV     4(R1),INVEC             ;STORE AWAY INPUT INTERRUPT VECTOR
7293  034500  016137  000004  012402          MOV     4(R1),OUTVEC
7294  034506  062737  000004  012402          ADD     #4,OUTVEC               ;BUILD OUTPUT INTERRUPT VECTOR
7295  034514  016137  000006  012404          MOV     6(R1),INTPRI            ;STORE AWAY INTERRUPT PRIORITY
7296  034522  016137  000012  012406          MOV     12(R1),OPTYP            ;STORE AWAY DEVICE OPTION TYPE
7297
7298  034530                          ENDIT:
7299  034530                                  SETVEC  CLKVEC,#CLKINT,#340     ;SETUP CLOCK VECTOR
 (7)  034530  012746  000340                                                          MOV     #340,-(SP)
 (6)  034534  012746  026230                                                          MOV     #CLKINT,-(SP)
 (5)  034540  013746  007526                                                          MOV     CLKVEC,-(SP)
 (4)  034544  012746  000003                                                          MOV     #3,-(SP)
 (3)  034550  104437                                                                  TRAP    C$SVEC
 (2)  034552  062706  000010                                                          ADD     #10,SP
7300
7301                                      ;DEVICE DEPENDENT VECTOR SETUP
7302
7311  034556                                  SETVEC  INVEC,#DVINS,INTPRI     ;SETUP INPUT INTERRUPT VECTOR
 (7)  034556  013746  012404                                                          MOV     INTPRI,-(SP)
 (6)  034562  012746  044620                                                          MOV     #DVINS,-(SP)
 (5)  034566  013746  012400                                                          MOV     INVEC,-(SP)
 (4)  034572  012746  000003                                                          MOV     #3,-(SP)
 (3)  034576  104437                                                                  TRAP    C$SVEC
 (2)  034600  062706  000010                                                          ADD     #10,SP
7312  034604                                  SETVEC  OUTVEC,#DVOUTS,INTPRI   ;SETUP OUTPUT INTERRUPT VECTOR
 (7)  034604  013746  012404                                                          MOV     INTPRI,-(SP)
 (6)  034610  012746  044630                                                          MOV     #DVOUTS,-(SP)
```

```
   (5)  034614  013746  012402                                                    MOV     OUTVEC,-(SP)
   (4)  034620  012746  000003                                                    MOV     #3,-(SP)
   (3)  034624  104437                                                            TRAP    C$SVEC
   (2)  034626  062706  000010                                                    ADD     #10,SP
  7313
  7314  034632                        SETPRI  #PRI00          ;SET THE 'RUN' PRIORITY TO 0
   (3)  034632  012700  000000                                                    MOV     #PRI00,R0
   (3)  034636  104441                                                            TRAP    C$SPRI
  7315  034640                        EXIT    INIT
   (3)  034640  104432                                                            TRAP    C$EXIT
   (3)  034642  000002                                                            .WORD   L10013-.
  7316
  7328
  7329                                .EVEN
  7330
  7331  034644                        ENDINIT
   (3)  034644                                                          L10013:
   (3)  034644  104411                                                            TRAP    C$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
(3) 034646                                                              L$AUTO::

034646                                  ENDAUTO
(3) 034646                                                              L10014:
(3) 034646  104461                                                              TRAP    C$AUTO
```

```
 7353                                   .SBTTL  CLEANUP CODING SECTION
 7354
 7355                                   ;++
 7356                                   ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
 7357                                   ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
 7358                                   ;--
 7359
 7360  034650                           BGNCLN
  (3)  034650                                                                  L$CLEAN::
 7361
 7370  034650  004737  045714           JSR     PC,DVBTUP       ;GO UPDATE BASE TABLE
 7371  034654  005077  152642           CLR     @CLKCSR         ;DISABLE CLOCK
 7372  034660                           SETPRI  #PRI07          ;SET PROCESSOR PRIORITY BACK TO 7
  (3)  034660  012700  000340                                                  MOV     #PRI07,R0
  (3)  034664  104441                                                          TRAP    C$SPRI
 7373
 7374  034666                           EXIT    CLN
  (3)  034666  104432                                                          TRAP    C$EXIT
  (3)  034670  000002                                                          .WORD   L10015-.
 7375
 7387
 7388                                   .EVEN
 7389
 7390  034672                           ENDCLN
  (3)  034672                                                                  L10015:
  (3)  034672  104412                                                          TRAP    C$CLEAN
```

```
 7392                                   .SBTTL  DROP UNIT SECTION
 7393
 7394                                   ;++
 7395                                   ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
 7396                                   ; TO NO LONGER BE TESTED.
 7397                                   ;--
 7398
 7399  034674                                   BGNDU
  (3)  034674                                                                  L$DU::
 7400
 7409
 7410  034674                                   EXIT    DU
  (4)  034674  000167                                                                 .WORD   J$JMP
  (3)  034676  000000                                                                 .WORD   L10016-2-.
 7411
 7423
 7424                                   .EVEN
 7425
 7426  034700                                   ENDDU
  (3)  034700                                                                  L10016:
  (3)  034700  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.
                                ;--

034702                                  BGNAU
034702                                                                  L$AU::

                                        EXIT    AU
034702  000167                                                                .WORD   J$JMP
034704  000000                                                                .WORD   .L10017-2-.

                                        .EVEN

034706                                  ENDAU
034706                                                                  L10017:
034706  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.
                                        ;--


034710                                          BGNTST
(3) 034710                                                                              T1::

                                        .SBTTL          PROGRAM SETUP SECTION

034710  013777  007532  152604          MOV     CLKEN,@CLKCSR   ;ENABLE THE CLOCK

034716                                  GTXRXB:
034716  005001                          GTRA2:  CLR     R1
034720  012737  000001  007542                  MOV     #1,TIMER1       ;SET TIMER TO COUNT 1 TICK
034726  005737  007542                  1$:     TST     TIMER1          ;CHECK FOR IT TO BE COUNTED OFF
034732  001412                                  BEQ     GTRA3           ;BRANCH IF CLOCK EXISTS (COUNTED A TICK)
034734  005301                                  DEC     R1
034736  001373                                  BNE     1$              ;KEEP CHECKING UNTIL R1 DOES FULL COUNTDWN
034740                                          PRINTF  #NOCLK          ;PRINT BAD CLK MSG AND WARN OF HANG IF TIMEOUT
(7) 034740  012746  015102                                                              MOV     #NOCLK,-(SP)
(6) 034744  012746  000001                                                              MOV     #1,-(SP)
(3) 034750  010600                                                                      MOV     SP,R0
(4) 034752  104417                                                                      TRAP    C$PNTF
(4) 034754  062706  000004                                                              ADD     #4,SP

034760  005737  007466          GTRA3:  TST     RESFLG          ;SEE IF HERE AFTER A RESTART.
034764  001117                          BNE     GTRA5           ;BR IF HERE CAUSE OF A RESTART

                                        ; CLEAR COUNTS AND SET UP DEFAULTS

034766  005037  007422          GTRA4:  CLR     TOTCC           ;CLEAR TOTAL CHAR. COUNT TEMP. LOC.
034772  005037  007356                  CLR     TTOTCC          ; CLEAR TOTAL CHAR. COUNT FOR TX BUFF
034776  005037  007342                  CLR     CTOTCC          ; CLEAR TOTAL CHAR. COUNT FOR CMP BUFF
035002  012701  006562                  MOV     #PTRTAB,R1      ;INIT TRANSMIT MESSAGE POINTER
035006  010137  007334                  MOV     R1,TXPTR
035012  005037  007332                  CLR     RXPTR           ; ZERO RX POINTER
035016  012702  000017                  MOV     #MSGLIM,R2
035022  006302                          ASL     R2
035024  006302                          ASL     R2
035026  010137  007336                  MOV     R1,CMPPTR
035032  060237  007336                  ADD     R2,CMPPTR       ;INIT COMPARE MESSAGE POINTER

035036  012737  000005  007410          MOV     #5,MSGTYP       ;SET UP DEFAULT MSG TYPE (QUICK FOX - ITEP MSG)
035044  013737  002162  007412          MOV     MSG5C,CURCC     ;SET UP DEFAULT CHAR COUNT
035052  012737  003562  007360          MOV     #TXBUF,TCURAD   ;SET UP CURRENT ADD TO START OF TX BUFFER
035060  012737  005562  007344          MOV     #CMPBUF,CCURAD  ;SET UP CURRENT ADD TO START OF CMP BUFFER

035066  013737  007360  007420          MOV     TCURAD,CURADD   ;SETUP CURRENT ADDR TO START OF TXBUF
```

```
 7533  035074  013737  007334  007416          MOV     TXPTR,CPTR      ;SETUP CURRENT POINTER TABLE POINTER FOR TXBUF
 7534  035102  004737  032010                  JSR     PC,BLDBUF       ; GO BUILD POINTER TABLE AND BUFFER
 7535  035106  012737  000001  007354          MOV     #1,TXMTOT       ;BUMP TOTAL MESSAGE COUNT
 7536
 7537  035114  013737  007336  007416          MOV     CMPPTR,CPTR     ;SET UP START OF COMPARE POINTER TABLE
 7538  035122  013737  007344  007420          MOV     CCURAD,CURADD   ;SET UP CURRENT ADDR. TO START OF CMPBUF
 7539  035130  012737  000005  007410          MOV     #5,MSGTYP
 7540  035136  013737  002162  007412          MOV     MSG5C,CURCC
 7541  035144  004737  032010                  JSR     PC,BLDBUF       ;PUT DEFAULT MESSAGE INTO CMPBUF
 7542  035150  012737  000001  007340          MOV     #1,CMPTOT       ;BUMP THE COMP MESG COUNT
 7543  035156  012737  000003  007470          MOV     #ACT,MODTYP     ;SET DEFAULT MODE= ACTIVE
 7544  035164  005037  007472                  CLR     MLTYP           ;SET DEFAULT MAINTENANCE LOOP MODE =NONE
 7545  035170  012737  000001  007500          MOV     #1,RPASS        ;SET UP DEFAULT 'RUN PASS' COUNT TO 1
 7546  035176  012737  000002  007476          MOV     #2,PARAM        ;SET UP PROG. PARAMETERS - DATACHECKING ENABLED
 7547                                                                  ;                  OPERATOR STATUS MSGS. PRINT OFF
 7548  035204                                  PRINTF  #HLP0
  (7)  035204  012746  013147                                                             MOV     #HLP0,-(SP)
  (6)  035210  012746  000001                                                             MOV     #1,-(SP)
  (3)  035214  010600                                                                     MOV     SP,R0
  (4)  035216  104417                                                                     TRAP    C$,NTF
  (4)  035220  062706  000004                                                             ADD     #4,SP
 7549  035224  013737  007470  010650  GTRA5:  MOV     MODTYP,DEV1
 7550  035232  013737  007472  010652          MOV     MLTYP,DEV2
 7551  035240  013737  007500  010654          MOV     RPASS,DEV3
 7552  035246  013737  007476  010656          MOV     PARAM,DEV4
 7553  035254  004737  032340                  JSR     PC,SHWOP        ;PRINT TO OPERATOR THE CURRENT MODE.....
 7554
 7555  035260                                  MANUAL                  ;SEE IF MANUAL INTERVENTION ALLOWED
  (3)  035260  104450                                                                     TRAP    C$MANI
 7556  035262                                  BCOMPLETE       GETCL   ;  BR IF YES (UAM=0 AND NOT CHAINED)
  (2)  035262  103412                                                                     BCS     GETCL
 7557  035264  005737  007500                  TST     RPASS           ;SEE IF THIS IS FIRST 'DCLT PASS''
 7558  035270  001002                          BNE     1$              ; BR IF NOT COMPLETED 1 PASS
 7559  035272                                  EXIT    TST             ; IF DONE 1 PASS IN UNATTENDED MODE - EXIT
  (3)  035272  104432                                                                     TRAP    C$EXIT
  (3)  035274  010560                                                                     .WORD   L10020-.
 7560  035276  012737  000001  007472  1$:     MOV     #TTL,MLTYP      ;SET UP DEFAULT FOR UNATTENDED MODE
 7561  035304  000137  040334                  JMP     GTR9            ;  'R M=ACT/LO=I/PAS=1/NOST/CH'' AND RUN
 7562
 7563                          .SBTTL          COMMAND LINE FETCH & INTERPRETATION SECTION
 7564
 7565  035310  105037  003561  GETCL:  CLRB    P$GDBD          ;CLEAR CMD LINE PARSING ERROR FLAGS
 7566  035314  105037  003560          CLRB    P$NNUF
 7567  035320                          GMANID  CLI$PM,CMDBUF,A,0,1,72.,NO      ;GET A COMMAND LINE FROM OPR.
  (3)  035320  104443                                                                     TRAP    C$GMAN
  (3)  035322  000406                                                                     BR      10000$
  (4)  035324  003060                                                                     .WORD   CMDBUF
  (5)  035326  000142                                                                     .WORD   T$CODE
  (5)  035330  012476                                                                     .WORD   CLI$PM
  (5)  035332  000000                                                                     .WORD   0
  (5)  035334  000001                                                                     .WORD   T$LOLIM
  (5)  035336  000110                                                                     .WORD   T$HILIM
  (3)  035340                                                                     10000$:
 7568  035340  012737  003060  003544          MOV     #CMDBUF,P$BUFA
 7569  035346  012737  010660  003546          MOV     #CLITRE,P$TREE
 7570  035354  012737  036302  003550          MOV     #CLIACT,P$ACT
```

```
 7571  035362  005037  003204                  CLR     QUALFG                  ;CLEAR QUALIFIER FLAG LOCATION
 7572  035366  004737  032642                  JSR     PC,P$TRV                ;GO PARSE COMMAND LINE
 7573  035372  105737  003561                  TSTB    P$GDBD                  ;SEE IF PARSED OK OR AN ERROR
 7574  035376  001412                          BEQ     1$
 7575  035400                                  PRINTF  #CLIERM
   (7) 035400  012746  012512                                                          MOV     #CLIERM,-(SP)
   (6) 035404  012746  000001                                                          MOV     #1,-(SP)
   (3) 035410  010600                                                                  MOV     SP,R0
   (4) 035412  104417                                                                  TRAP    C$PNTF
   (4) 035414  062706  000004                                                          ADD     #4,SP
 7576  035420  000137  035310          1$:     JMP     GETCL
 7577  035424  105737  003560                  TSTB    P$NNUF                  ;SEE IF INCOMPLETE COMMAND TYPED
 7578  035430  001412                          BEQ     10$
 7579  035432                                  PRINTF  #CLINUF
   (7) 035432  012746  012542                                                          MOV     #CLINUF,-(SP)
   (6) 035436  012746  000001                                                          MOV     #1,-(SP)
   (3) 035442  010600                                                                  MOV     SP,R0
   (4) 035444  104417                                                                  TRAP    C$PNTF
   (4) 035446  062706  000004                                                          ADD     #4,SP
 7580  035452  000137  035310                  JMP     GETCL
 7581
 7582                                          ;REV B BY EC
 7583  035456  023727  003202  000060  10$:    CMP     KEYWD1,#SETET           ;WAS ''SET EXPECT = TRANMIT'' ENTERED?
 7584  035464  001711                          BEQ     GETCL                   ;YES,BRANCH
 7585
 7586  035466  023727  003202  000005          CMP     KEYWD1,#HLP             ;SEE IF HELP WAS TYPED
 7587  035474  001705                          BEQ     GETCL                   ;GO GET CMD AGAIN IF YES
 7588  035476  023727  003202  000055          CMP     KEYWD1,#PRNT            ;SEE IF PRINT WAS TYPED
 7589  035504  001701                          BEQ     GETCL                   ; GO GET CMD AGAIN IF YES
 7590  035506  023727  003202  000004          CMP     KEYWD1,#RUN             ;SEE IF RUN WAS TYPED
 7591  035514  001002                          BNE     11$                     ; BR IF NO
 7592  035516  000137  040334                  JMP     GTR9                    ; START EXEC. IF YES
 7593  035522  023727  003202  000052  11$:    CMP     KEYWD1,#DMPS            ;SEE IF DUMP WAS TYPED
 7594  035530  001004                          BNE     12$                     ; BR IF NO
 7595  035532  004737  031554                  JSR     PC,DUMPSR               ; ELSE, DUMP PART OF MEMORY
 7596  035536  000137  035310                  JMP     GETCL                   ; THEN RETURN TO GET ANOTHER CMD.
 7597                                          ;;EXIT COMMAND IS A REVISION B CHANGE BY EC
 7598  035542  023727  003202  000057  12$:    CMP     KEYWD1,#EXIT            ;EXIT COMMAND ?
 7599  035550  001005                          BNE     13$                     ;NO,BRANCH
 7600  035552  012737  000001  007464          MOV     #1,DCLFLG               ;SET CLEANUP & EXIT FLAG
 7601  035560                                  EXIT    TST                     ;GO BACK TO INIT ROUTINE
   (3) 035560  104432                                                                  TRAP    C$EXIT
   (3) 035562  010272                                                                  .WORD   L10020-.
 7602  035564  023727  003202  000001  13$:    CMP     KEYWD1,#CLEAR           ;SEE IF CLEAR WAS TYPED
 7603  035572  001646                          BEQ     GETCL                   ; IF YES, BACK TO GET ANOTHER CMD.
 7604  035574  023727  003202  000002          CMP     KEYWD1,#SHOW            ;SEE IF SHOW WAS TYPED
 7605  035602  001642                          BEQ     GETCL                   ; IF YES, BACK TO GET ANOTHER CMD.
 7606  035604  023727  003202  000010  4$:     CMP     KEYWD1,#SETEXP          ;SEE IF SET EXPECTED
 7607  035612  001512                          BEQ     2$                      ; BR IF YES (A SETEXP WAS TYPED)
 7608  035614  013737  007356  007422  5$:     MOV     TTOTCC,TOTCC
 7609  035622  023727  007422  001000          CMP     TOTCC,#BUFLIM           ;SEE IF BUFFER ALREADY FULL
 7610  035630  002414                          BLT     15$                     ; BR IF NOT FULL (BUFLIM # OF CHARS.)
 7611  035632                                  PRINTF  #MSGTRN,#BUFEX          ; ELSE TELL OPR. AND DON'T BUILD MSG.
   (8) 035632  012746  015223                                                          MOV     #BUFEX,-(SP)
   (7) 035636  012746  015241                                                          MOV     #MSGTRN,-(SP)
   (6) 035642  012746  000002                                                          MOV     #2,-(SP)
```

```
  (3)  035646  010600                                                    MOV     SP,R0
  (4)  035650  104417                                                    TRAP    C$PNTF
  (4)  035652  062706  000006                                            ADD     #6,SP
7612   035656  000137  035310                JMP     GETCL           ; THEN GO GET A NEW COMMAND
7613   035662  005737  007356          15$:  TST     TTOTCC          ;IF FIRST ''SET'' THEN GET RID OF DEFAULT
7614   035666  001002                        BNE     6$
7615   035670  005037  007354                CLR     TXMTOT
7616   035674  012737  006562  007334  6$:   MOV     #PTRTAB,TXPTR   ;GET POSITION OF END OF TX LIST
7617   035702  013701  007354                MOV     TXMTOT,R1
7618   035706  020127  000017                CMP     R1,#MSGLIM      ;SEE IF MSG COUNT EXCEEDED.
7619   035712  002414                        BLT     17$             ; BR IF NO
7620   035714                                PRINTF  #MSGTRN,#TABEX  ; ELSE TELL OPR. AND DON'T BUILD MSG.
  (8)  035714  012746  015163                                                    MOV     #TABEX,-(SP)
  (7)  035720  012746  015241                                                    MOV     #MSGTRN,-(SP)
  (6)  035724  012746  000002                                                    MOV     #2,-(SP)
  (3)  035730  010600                                                    MOV     SP,R0
  (4)  035732  104417                                                    TRAP    C$PNTF
  (4)  035734  062706  000006                                            ADD     #6,SP
7621   035740  000137  035310                JMP     GETCL           ; THEN GO GET A NEW COMMAND.
7622   035744  006301                  17$:  ASL     R1              ;# OF MSGS *4 = NEXT FREE PTR BLOCK
7623   035746  006301                        ASL     R1
7624   035750  060137  007334                ADD     R1,TXPTR
7625   035754  013737  007334  007416        MOV     TXPTR,CPTR      ;SETUP CHAR. COUNT, CURRENT ADDR, & PTR
7626   035762  013737  007360  007420        MOV     TCURAD,CURADD
7627   035770  004737  031712                JSR     PC,ADDCC        ;ADD IN CHAR. COUNT AND CHECK TOTAL
7628   035774  004737  032010                JSR     PC,BLDBUF       ;GO BUILD MESSAGE IN BUFFER AND PTRS.
7629   036000  013737  007416  007334        MOV     CPTR,TXPTR
7630   036006  013737  007422  007356        MOV     TOTCC,TTOTCC    ;UPDATE CHAR. COUNT, CURR ADDR, & PTR
7631   036014  013737  007420  007360        MOV     CURADD,TCURAD
7632   036022  005237  007354                INC     TXMTOT
7633   036026  005337  003206                DEC     QUALVL          ;DLC THE COPY COUNT
7634   036032  001270                        BNE     5$
7635   036034  000137  035310                JMP     GETCL
7636
7637   036040  013737  007342  007422  2$:   MOV     CTOTCC,TOTCC    ;SETUP CHAR. COUNT, CURR. ADDR. & PTR
7638   036046  023727  007422  001000        CMP     TOTCC,#BUFLIM   ;SEE IF BUFFER ALREADY FULL
7639   036054  002414                        BLT     16$             ; BR IF NOT FULL (BUFLIM # OF CHARS.)
7640   036056                                PRINTF  #MSGTRN,#BUFEX  ; ELSE TELL OPR. AND DON'T BUILD MSG.
  (8)  036056  012746  015223                                                    MOV     #BUFEX,-(SP)
  (7)  036062  012746  015241                                                    MOV     #MSGTRN,-(SP)
  (6)  036066  012746  000002                                                    MOV     #2,-(SP)
  (3)  036072  010600                                                    MOV     SP,R0
  (4)  036074  104417                                                    TRAP    C$PNTF
  (4)  036076  062706  000006                                            ADD     #6,SP
7641   036102  000137  035310                JMP     GETCL           ; THEN GO GET A NEW COMMAND
7642   036106  005737  007342          16$:  TST     CTOTCC          ;IF FIRST ''SET'' THEN GET RID OF DEFAULT
7643   036112  001002                        BNE     7$
7644   036114  005037  007340                CLR     CMPTOT
7645   036120  012701  006562          7$:   MOV     #PTRTAB,R1
7646   036124  012702  000017                MOV     #MSGLIM,R2
7647   036130  006302                        ASL     R2
7648   036132  006302                        ASL     R2
7649   036134  010137  007336                MOV     R1,CMPPTR
7650   036140  060237  007336                ADD     R2,CMPPTR       ;INIT COMPARE MESSAGE POINTER
7651   036144  013701  007340                MOV     CMPTOT,R1
7652   036150  020127  000017                CMP     R1,#MSGLIM      ;SEE IF MSG COUNT EXCEEDED.
```

```
 7653  036154  002414                          BLT     18$                     ; BR IF NO
 7654  036156                                  PRINTF  #MSGTRN,#TABEX          ; ELSE TELL OPR. AND DON'T BUILD MSG.
  (8)  036156  012746  015163                                                            MOV     #TABEX,-(SP)
  (7)  036162  012746  015241                                                            MOV     #MSGTRN,-(SP)
  (6)  036166  012746  000002                                                            MOV     #2,-(SP)
  (3)  036172  010600                                                                    MOV     SP,R0
  (4)  036174  104417                                                                    TRAP    C$PNTF
  (4)  036176  062706  000006                                                            ADD     #6,SP
 7655  036202  000137  035310                  JMP     GETCL                   ; THEN GO GET A NEW COMMAND.
 7656  036206  006301                  18$:    ASL     R1                      ;# OF MSGS *4 = NEXT FREE PTR BLOCK
 7657  036210  006301                          ASL     R1
 7658  036212  060137  007336                  ADD     R1,CMPPTR
 7659  036216  013737  007336  007416          MOV     CMPPTR,CPTR
 7660  036224  013737  007344  007420          MOV     CCURAD,CURADD
 7661  036232  004737  031712                  JSR     PC,ADDCC                ;ADD IN XHAR. COUNT AND CHECK TOTAL
 7662  036236  004737  032010                  JSR     PC,BLDBUF
 7663  036242  013737  007416  007336          MOV     CPTR,CMPPTR
 7664  036250  005237  007340                  INC     CMPTOT
 7665  036254  013737  007420  007344          MOV     CURADD,CCURAD           ;UPDATE CHAR. COUNT, CURR ADDRR. & PTR
 7666  036262  013737  007422  007342          MOV     TOTCC,CTOTCC
 7667  036270  005337  003206                  DEC     QUALVL                  ;IF COPY WAS GIVEN, PUT MSG IN BUFF
 7668  036274  001261                          BNE     2$                      ; AGAIN
 7669  036276  000137  035310                  JMP     GETCL                   ;GO BACK UNTIL GET A 'RUN''
 7670
 7671
 7672
 7673
 7674
```

```
                                        .SBTTL          ACTION TABLE AND ROUTINES
                                        ;       USER MUST CLEAR/SET P$GDBD IF USE "CLIBIF" IN CONNECTION WITH ACTION
                                        ;       R2 WILL HOLD ACTION CODE FROM PARSING (CLI) NODE
                                        ;
036302                                  CLIACT:
036302  006302                                  ASL     R2                      ;MULTIPLY ACTION CODE BY 2
036304  016202  036320                          MOV     10$(R2),R2              ;OFFSET VALUE
036310  062702  036320                          ADD     #10$,R2                 ;ADD BASE VALUE
036314  004712                                  JSR     PC,(R2)                 ;GO DO ACTION
036316  000207                                  RTS     PC                      ;RETURN TO TRVACT:

                                                                                ;BRIEF DESCRIPTION OF ACTIONS TAKEN
036320  000150                          10$:    .WORD   ACTNUL-10$              ;NULL
036322  000152                                  .WORD   ACTCLR-10$              ;CLEAR
036324  000162                                  .WORD   ACTSHO-10$              ;SHOW
036326  001604                                  .WORD   ACTCHK-10$              ;CHECK
036330  000262                                  .WORD   ACTRUN-10$              ;RUN
036332  000172                                  .WORD   ACTHLP-10$              ;HELP
036334  000306                                  .WORD   ACTCSE-10$              ;CLEAR OR SHOW EXPECTED
036336  000434                                  .WORD   ACTCST-10$              ;CLEAR OR SHOW TRANSMIT
036340  000756                                  .WORD   ACTSTE-10$              ;SET EXPECTED
036342  000766                                  .WORD   ACTSTT-10$              ;SET TRANSMIT
036344  001004                                  .WORD   ACTSZE-10$              ;SIZE
036346  001014                                  .WORD   ACTCOP-10$              ;COPY
036350  001024                                  .WORD   ACTNUM-10$              ;NUMERIC VALUE FOR SIZE OR COPY
036352  001116                                  .WORD   ACTQPM-10$              ;QUOTED MESSAGE FROM USER
036354  001612                                  .WORD   ACTSTS-10$              ;STATUS
036356  001136                                  .WORD   ACTEQQ-10$              ;END OF QUOTED MESSAGE FROM USER
036360  001216                                  .WORD   ACTMS0-10$              ;ONES
036362  001224                                  .WORD   ACTMS1-10$              ;ZEROS
036364  001234                                  .WORD   ACTMS2-10$              ;1ALT
036366  001244                                  .WORD   ACTMS3-10$              ;0ALT
036370  001254                                  .WORD   ACTMS4-10$              ;ITEP
036372  001264                                  .WORD   ACTMS5-10$              ;CCITT
036374  001302                                  .WORD   ACTMS6-10$              ;ALPHA
036376  001370                                  .WORD   ACTATV-10$              ;ACTIVE MODE
036400  001400                                  .WORD   ACTPAS-10$              ;PASSIVE MODE
036402  001420                                  .WORD   ACTREC-10$              ;RECEIVE MODE
036404  001426                                  .WORD   ACTLIS-10$              ;LISTEN MODE
036406  001436                                  .WORD   ACTDLL-10$              ;DOWNLINE LOAD
036410  001446                                  .WORD   ACTTRA-10$              ;TRANSMIT MODE
036412  001456                                  .WORD   ACTTAL-10$              ;TALK MODE
036414  001504                                  .WORD   ACTNO-10$               ;NO
036416  001514                                  .WORD   ACTECH-10$              ;ECHO
036420  001620                                  .WORD   ACTCRC-10$              ;SET CRC BIT
036422  001634                                  .WORD   ACTPRO-10$              ;SET PROTOCOL BIT
036424  001666                                  .WORD   ACTRPS-10$              ;STATUS
036426  001676                                  .WORD   ACTMOP-10$              ;REMOTE STATION IN MAINTENANCE LOOP MODE
036430  001706                                  .WORD   ACTTLP-10$              ;INTERNALTTL
036432  001716                                  .WORD   ACTCLP-10$              ;CABLE LOOP
036434  001726                                  .WORD   ACTLLP-10$              ;LOCAL MODEM LOOP
036436  001736                                  .WORD   ACTRLP-10$              ;REMOTE MODEM LOOP
036440  000142                                  .WORD   ACTNUF-10$              ;MORE COMMAND LINE NEEDED
036442  001174                                  .WORD   ACTBCR-10$              ;BAD CHARACTER IN OPERATOR MESSAGE
036444  000712                                  .WORD   ACTDMS-10$              ;DUMP MEMORY START ADDRESS
036446  000742                                  .WORD   ACTDME-10$              ;DUMP MEMORY END ADDRESS
```

```
 7732  036450  000734                .WORD   ACTDMQ-10$      ;DUMP WORD
 7733  036452  000246                .WORD   ACTPRT-10$      ;PRINT
 7734  036454  001626                .WORD   ACTMOS-10$      ;MODEM ACTION REV B BY EC
 7735  036456  000236                .WORD   ACTEXT-10$      ;EXIT ACTION  REV B BY EC
 7736  036460  001326                .WORD   ACTSEX-10$      ;SET E=T ACTION REV B BY EC NPI
 7737
```

```
 7739
 7740  036462  112737  177777  003560  ACTNUF: MOVB    #-1,P$NNUF      ;SET FLAG TO SAY NEED MORE OF COMMAND
 7741  036470  000207                  ACTNUL: RTS     PC              ;RETURN TO PARSER
 7742
 7743  036472  012737  000001  003202  ACTCLR: MOV     #CLEAR,KEYWD1   ;SET LOC TO SAY A CLEAR WAS TYPED
 7744  036500  000207                          RTS     PC
 7745
 7746  036502  012737  000002  003202  ACTSHO: MOV     #SHOW,KEYWD1    ;SET LOC. TO SAY A SHOW WAS TYPED
 7747  036510  000207                          RTS     PC
 7748
 7749  036512  012702  003210          ACTHLP: MOV     #HLPTAB,R2      ;SETUP R2 AS A POINTER TO HELP MSG TABLE
 7750  036516                          1$:     PRINTF  #HLPF,(R2)+     ;PRINT HELP INFORMATION MESSAGES
  (8)  036516  012246                                                                  MOV     (R2)+,-(SP)
  (7)  036520  012746  013225                                                          MOV     #HLPF,-(SP)
  (6)  036524  012746  000002                                                          MOV     #2,-(SP)
  (3)  036530  010600                                                                  MOV     SP,R0
  (4)  036532  104417                                                                  TRAP    C$PNTF
  (4)  036534  062706  000006                                                          ADD     #6,SP
 7751  036540  020227  003230                  CMP     R2,#HLPEND      ;SEE IF ALL INFO PRINTED YET
 7752  036544  001364                          BNE     1$              ;IF NO KEEP PRINTING
 7753  036546  012737  000005  003202          MOV     #HLP,KEYWD1     ;SET LOC. TO SAY A HELP WAS TYPED
 7754  036554  000207                          RTS     PC
 7755  036556  012737  000057  003202  ACTEXT: MOV     #EXIT,KEYWD1    ;EXIT COMMAND
 7756  036564  000207                          RTS     PC
 7757  036566  012737  000055  003202  ACTPRT: MOV     #PRNT,KEYWD1    ;SET LOC. TO SAY A HELP WAS TYPED
 7758  036574  004737  027046                  JSR     PC,REPORT       ;CALL ROUTINE TO PRINT EVENT LOG AND BASE TABLE
 7759  036600  000207                          RTS     PC
 7760
 7761  036602  012737  000004  003202  ACTRUN: MOV     #RUN,KEYWD1     ;SET RUN FLAG
 7762  036610  112737  177777  003560          MOVB    #-1,P$NNUF      ;SET FLAG TO SAY NEED MORE OF COMMAND
 7763  036616  012737  000001  007500          MOV     #1,RPASS        ;SET DEFAULT RUN 'PASS'' TO 1
 7764  036624  000207                          RTS     PC
 7765
 7766  036626  012701  006562          ACTCSE: MOV     #PTRTAB,R1
 7767  036632  012702  000017                  MOV     #MSGLIM,R2
 7768  036636  006302                          ASL     R2
 7769  036640  006302                          ASL     R2
 7770  036642  010137  007336                  MOV     R1,CMPPTR
 7771  036646  060237  007336                  ADD     R2,CMPPTR       ;INIT COMPARE MESSAGE POINTER
 7772  036652  013701  007336                  MOV     CMPPTR,R1
 7773
 7774  036656  013702  007340                  MOV     CMPTOT,R2
 7775  036662  105037  003560                  CLRB    P$NNUF          ;FLAG THAT HAVE VALID COMMAND AT THIS PT.
 7776  036666  023727  003202  000002          CMP     KEYWD1,#SHOW    ;SEE IF A CLEAR OR SHOW WAS TYPED
 7777  036674  001500                          BEQ     ACTSHW          ;BR IF A SHOW WAS TYPED
 7778  036676  012737  000001  007340          MOV     #1,CMPTOT       ;CLEAR COMPARE MESSAGE COUNT, CHAR. COUNT
 7779  036704  005037  007342                  CLR     CTOTCC          ; AND RESET POINTER
 7780
 7781  036710  012701  006562                  MOV     #PTRTAB,R1
 7782  036714  012702  000017                  MOV     #MSGLIM,R2
 7783  036720  006302                          ASL     R2
 7784  036722  006302                          ASL     R2
 7785  036724  010137  007336                  MOV     R1,CMPPTR
 7786  036730  060237  007336                  ADD     R2,CMPPTR       ;INIT COMPARE MESSAGE POINTER
 7787  036734  013737  007336  007416          MOV     CMPPTR,CPTR     ;SET UP TO FILL IN DEFAULT MESSAGE
 7788  036742  012701  005562                  MOV     #CMPBUF,R1
```

```
 7789  036746  010137  007344                 MOV     R1,CCURAD
 7790  036752  000431                         BR      ACTCLB
 7791
 7792  036754  012701  006562         ACTCST: MOV     #PTRTAB,R1
 7793  036760  013702  007354                 MOV     TXMTOT,R2
 7794  036764  105037  003560                 CLRB    P$NMUF          ;FLAG THAT HAVE VALID COMMAND AT THIS PT.
 7795  036770  023727  003202  000002         CMP     KEYWD1,#SHOW    ;SEE IF A CLEAR OR SHOW WAS TYPED
 7796  036776  001437                         BEQ     ACTSHW          ;BR IF A SHOW WAS TYPED
 7797  037000  012737  000001  007354         MOV     #1,TXMTOT       ;CLEAR TRANSMIT MESSAGE COUNT, CHAR. COUNT
 7798  037006  005037  007356                 CLR     TTOTCC          ; AND RESET POINTER
 7799  037012  012737  006562  007334         MOV     #PTRTAB,TXPTR
 7800  037020  013737  007334  007416         MOV     TXPTR,CPTR
 7801  037026  012701  003562                 MOV     #TXBUF,R1
 7802  037032  010137  007360                 MOV     R1,TCURAD
 7803
 7804  037036  012702  001000         ACTCLB: MOV     #BUFLIM,R2
 7805  037042  010137  007420                 MOV     R1,CURADD       ;SET UP TO PUT DEFAULT MSG IN LIST AFTER 033'S
 7806  037046  012737  000005  007410         MOV     #5,MSGTYP
 7807  037054  013737  002162  007412         MOV     MSG5C,CURCC
 7808  037062  105021                 1$:     CLRB    (R1)+           ;FILL EXPT OR TRAN BUFFER WITH 0'S IF A CLEAR
 7809  037064  005302                         DEC     R2              ;DO 'BUFLIM' NUMBER OF BYTE LOCATIONS
 7810  037066  001375                         BNE     1$
 7811  037070  004737  032010                 JSR     PC,BLDBUF       ;''CLEAR'' REALLY MEANS TO PUT DEFAULT MSG IN
 7812  037074  000207                         RTS     PC              ;WHEN DONE, RETURN TO PARSER
 7813
 7814
 7815  037076  012705  003504         ACTSHW: MOV     #SHTAB,R5
 7816  037102  122571  000000         5$:     CMPB    (R5)+,@(R1)     ;LOOK AT FIRST BYTE OF MSG TO DECIPHER TYPE
 7817  037106  001404                         BEQ     6$
 7818  037110  020527  003513                 CMP     R5,#SHTEND      ;SEE IF LOOKED AT ALL OF DEFAULTS YET
 7819  037114  001372                         BNE     5$
 7820  037116  005205                         INC     R5              ;MUST BE OPR. SPEC'D THEN
 7821  037120  162705  003505         6$:     SUB     #SHTAB+1,R5
 7822  037124  006305                         ASL     R5
 7823  037126  016137  000002  007426         MOV     2(R1),TEMP
 7824  037134                                 PRINTF  #SHMSG,SHTYTB(R5),TEMP  ;PRINT MSG SIZE & TYPE
  (9)  037134  013746  007426                                                 MOV     TEMP,-(SP)
  (8)  037140  016546  003464                                                 MOV     SHTYTB(R5),-(SP)
  (7)  037144  012746  014430                                                 MOV     #SHMSG,-(SP)
  (6)  037150  012746  000003                                                 MOV     #3,-(SP)
  (3)  037154  010600                                                         MOV     SP,R0
  (4)  037156  104417                                                         TRAP    C$PNTF
  (4)  037160  062706  000010                                                 ADD     #10,SP
 7825  037164  062701  000004                 ADD     #4,R1           ;BUMP R1 TO NEXT SET OF POINTERS
 7826  037170  005302                         DEC     R2
 7827  037172  001341                         BNE     ACTSHW
 7828  037174  013737  007470  010650         MOV     MODTYP,DEV1
 7829  037202  013737  007472  010652         MOV     MLTYP,DEV2
 7830  037210  013737  007500  010654         MOV     RPASS,DEV3
 7831  037216  013737  007476  010656         MOV     PARAM,DEV4
 7832  037224  004737  032340                 JSR     PC,SHWOP        ;SHOW THE OPERATOR THE CURRENT MODE..... ALSO
 7833  037230  000207                         RTS     PC
 7834
 7835  037232  013737  003554  007402 ACTDMS: MOV     P$NUM,STADD     ;SETUP STARTING ADDRESS FOR DUMP
 7836  037240  005037  007406                 CLR     BYTBIT          ;SET DEFAULT OF WORD DUMP
 7837  037244  012737  000052  003202         MOV     #DMPS,KEYWD1    ;FLAG THAT A DUMP WAS TYPED
```

```
 7838  037252  000403                          BR      ACTDME
 7839
 7840  037254  012737  177777  007406  ACTDMQ: MOV     #-1,BYTBIT      ;SET DUMP FLAG TO 'DUMP-WORD''
 7841  037262  013737  003554  007404  ACTDME: MOV     P$NUM,ENADD     ;SETUP END ADDRESS FOR DUMP (=START IF NO 'EEE''
 7842  037270  105037  003560          ACTDMX: CLRB    P$NNUF          ;CLEAR NOT-ENOUGH FLAG, 'DUMP N-N/B'' IS VALID
 7843  037274  000207                          RTS     PC
 7844
```

```
 7846
 7847
 7848  037276  012737  000010  003202  ACTSTE: MOV     #SETEXP,KEYWD1
 7849  037304  000403                          BR      ACTSTX
 7850
 7851  037306  012737  000011  003202  ACTSTT: MOV     #SETTRN,KEYWD1
 7852  037314  012737  000001  003206  ACTSTX: MOV     #1,QUALVL       ;SET UP DEFAULT COPY TO 1 (/COPY=0)
 7853  037322  000207                          RTS     PC
 7854
 7855  037324  012737  000012  003204  ACTSZE: MOV     #SIZE,QUALFG
 7856  037332  000207                          RTS     PC
 7857
 7858  037334  012737  000013  003204  ACTCOP: MOV     #QCOPY,QUALFG
 7859  037342  000207                          RTS     PC
 7860
 7861  037344  023727  003204  000012  ACTNUM: CMP     QUALFG,#SIZE    ;SEE IF A SIZE OR COPY TYPED
 7862  037352  001023                          BNE     1$              ;BR IF IT WAS A COPY
 7863  037354  005737  003554                  TST     P$NUM           ;CHECK TO BE SURE DIDN'T TRY SIZE=0
 7864  037360  001014                          BNE     3$              ; BR IF NO
 7865  037362                                  PRINTF  #CLISEO
  (7)  037362  012746  013001                                                    MOV     #CLISEO,-(SP)
  (6)  037366  012746  000001                                                    MOV     #1,-(SP)
  (3)  037372  010600                                                            MOV     SP,R0
  (4)  037374  104417                                                            TRAP    C$PNTF
  (4)  037376  062706  000004                                                    ADD     #4,SP
 7866  037402  112737  177777  003561          MOVB    #-1,P$GDBD      ;SEE ERROR-IN-CMD FLAG
 7867  037410  000411                          BR      2$
 7868  037412  013737  003554  007412  3$:     MOV     P$NUM,CURCC     ;IF A SIZE LOAD CURCC WITH BYTE COUNT
 7869  037420  000405                          BR      2$
 7870  037422  013737  003554  003206  1$:     MOV     P$NUM,QUALVL    ;IF A COPY, LOAD COPY COUNT
 7871  037430  005237  003206                  INC     QUALVL          ;INCREMENT SO FIRST DEC MAKES IT REAL #
 7872  037434  000522                  2$:     BR      ACTMEX
 7873
 7874  037436  012737  000007  007410  ACTOPM: MOV     #7,MSGTYP
 7875  037444  010437  007426                  MOV     R4,TEMP         ;KEEP TRACK OF START OF QUOTED TEXT
 7876  037450  005237  007426                  INC     TEMP            ; SO CAN CALC OPCNT AT END OF QUOTES
 7877  037454  000207                          RTS     PC
 7878
 7879  037456  010402                  ACTEQO: MOV     R4,R2
 7880  037460  163702  007426                  SUB     TEMP,R2
 7881  037464  010237  007412                  MOV     R2,CURCC        ;CALC BYTE COUNT FOR QUOTED TEXT
 7882  037470  010237  002166                  MOV     R2,OPCNT
 7883  037474  013701  007426                  MOV     TEMP,R1
 7884  037500  012705  002524                  MOV     #OPBUF,R5
 7885  037504  112125                  1$:     MOVB    (R1)+,(R5)+     ;COPY QUOTED TEXT TO OPBUF
 7886  037506  005302                          DEC     R2
 7887  037510  001375                          BNE     1$
 7888  037512  000473                          BR      ACTMEX
 7889
 7890  037514                          ACTBCR: PRINTF  #CLIBCR         ;BAD CHAR. IN OPR. QUOTED STRING
  (7)  037514  012746  012734                                                    MOV     #CLIBCR,-(SP)
  (6)  037520  012746  000001                                                    MOV     #1,-(SP)
  (3)  037524  010600                                                            MOV     SP,R0
  (4)  037526  104417                                                            TRAP    C$PNTF
  (4)  037530  062706  000004                                                    ADD     #4,SP
 7891  037534  000207                          RTS     PC
```

```
 7892                                                ;SET THE MESSAGE TYPE AS PER COMMAND LINE
 7893  037536  005037  007410          ACTMS0: CLR     MSGTYP
 7894  037542  000435                          BR      ACTME1
 7895  037544  012737  000001  007410  ACTMS1: MOV     #1,MSGTYP       ;ALL ONES
 7896  037552  000431                          BR      ACTME1
 7897  037554  012737  000002  007410  ACTMS2: MOV     #2,MSGTYP       ;ONES & ZEROS
 7898  037562  000425                          BR      ACTME1
 7899  037564  012737  000003  007410  ACTMS3: MOV     #3,MSGTYP       ;ZEROS & ONES
 7900  037572  000421                          BR      ACTME1
 7901  037574  012737  000004  007410  ACTMS4: MOV     #4,MSGTYP       ;CCITT
 7902  037602  000415                          BR      ACTME1
 7903  037604  012737  000005  007410  ACTMS5: MOV     #5,MSGTYP       ;QUICK FOX
 7904  037612  013737  002162  007412          MOV     MSG5C,CURCC             ;SETUP DEFAULT SIZE FOR THIS TYPE
 7905  037620  000430                          BR      ACTMEX
 7906  037622  012737  000006  007410  ACTMS6: MOV     #6,MSGTYP       ;ALPHA/NUM
 7907  037630  013737  002164  007412          MOV     MSG6C,CURCC             ;SETUP DEFAULT SIZE FOR THIS TYPE
 7908
 7909  037636  012737  000100  007412  ACTME1: MOV     #64.,CURCC              ;SETUP DEFAULT SIZE FOR MSG0-4
 7910  037644  000416                          BR      ACTMEX                  ;GO TO EXIT
 7911
 7912                                                ;REV B BY EC
 7913  037646  022737  000010  003202  ACTSEX: CMP     #SETEXP,KEYWD1          ;DID WE GET HERE FROM 'SET E =' COMMAND?
 7914  037654  001404                          BEQ     10$                     ;YES,BRANCH
 7915  037656  112737  177777  003561          MOVB    #-1,P$GDBD              ;SET ERROR FLAG
 7916  037664  000406                          BR      ACTMEX                  ;GO TO EXIT
 7917  037666  004737  032134          10$:    JSR     PC,FACSIMILE            ;GO COPY TRANMIT BUFFER TO EXPECT BUFFER
 7918  037672  012737  000060  003202          MOV     #SETET,KEYWD1           ;SET FLAG TO BE USED IN T1::
 7919  037700  000400                          BR      ACTMEX                  ;GO TO EXIT
 7920
 7921  037702  105037  003560          ACTMEX: CLRB    P$NNUF                  ;CLEAR NOT-ENOUGH FLAG
 7922  037706  000207                          RTS     PC
 7923
```

```
 7925  037710  012737  000003  007470  ACTATV: MOV     #ACT,MODTYP
 7926  037716  000432                  BR      ACTM2X
 7927
 7928  037720  012737  000002  007470  ACTPAS: MOV     #PAS,MODTYP
 7929  037726  105037  003560          CLRB    P$NNUF          ;CLEAR NOT-ENOUGH FLAG
 7930  037732  005037  007472          CLR     MLTYP           ;CLEAR MAINT LOOP TYPE
 7931  037736  000207                  RTS     PC
 7932
 7933  037740  005037  007470          ACTREC: CLR     MODTYP
 7934  037744  000417                  BR      ACTM2X
 7935
 7936  037746  012737  000006  007470  ACTLIS: MOV     #LIS,MODTYP
 7937  037754  000413                  BR      ACTM2X
 7938
 7939  037756  012737  000004  007470  ACTDLL: MOV     #DOW,MODTYP
 7940  037764  000407                  BR      ACTM2X
 7941
 7942  037766  012737  000001  007470  ACTTRA: MOV     #TRA,MODTYP
 7943  037774  000403                  BR      ACTM2X
 7944
 7945  037776  012737  000005  007470  ACTTAL: MOV     #TAL,MODTYP
 7946
 7947  040004  042737  000004  007476  ACTM2X: BIC     #ECHOB,PARAM    ;DISABLE /ECHO (ALL BUT PASSIVE MODE)
 7948  040012  105037  003560          CLRB    P$NNUF          ;CLEAR NOT-ENOUGH FLAG
 7949  040016  005037  007472          CLR     MLTYP           ;CLEAR MAINT LOOP TYPE
 7950  040022  000207                  RTS     PC
 7951
```

```
 7953  040024  012737  000036  003204  ACTNO:  MOV     #NO,QUALFG
 7954  040032  000207                  RTS     PC
 7955
 7956  040034  022737  000036  003204  ACTECH: CMP     #NO,QUALFG
 7957  040042  001422                  BEQ     1$
 7958  040044  052737  000004  007476          BIS     #ECHOB,PARAM
 7959  040052  022737  000002  007470          CMP     #PAS,MODTYP             ;BE SURE IN PASSIVE MODE IF
 7960  040060  001416                  BEQ     2$                              ;IF TRYING TO SET /ECHO
 7961  040062                          PRINTF  #CLINPS
  (7)  040062  012746  012671                                                  MOV     #CLINPS,-(SP)
  (6)  040066  012746  000001                                                  MOV     #1,-(SP)
  (3)  040072  010600                                                          MOV     SP,R0
  (4)  040074  104417                                                          TRAP    C$PNTF
  (4)  040076  062706  000004                                                  ADD     #4,SP
 7962  040102  112737  177777  003561          MOVB    #-1,P$GDBD
 7963  040110  042737  000004  007476  1$:     BIC     #ECHOB,PARAM
 7964  040116  005037  003204          2$:     CLR     QUALFG                  ;CLEAR 'NO' OUT OF QUALIFIER FLAG
 7965  040122  000501                  BR      ACTLXX
 7966
 7967  040124  012701  000002          ACTCHK: MOV     #DATCKB,R1              ;SET DATA CHECK BIT
 7968  040130  000413                  BR      ACTQFG
 7969
 7970  040132  012701  000001          ACTSTS: MOV     #STATB,R1       ;SET THE STATUS BIT
 7971  040136  000410                  BR      ACTQFG
 7972
 7973  040140  012701  000020          ACTCRC: MOV     #CRCB,R1        ;SET THE CRC BIT
 7974  040144  000405                  BR      ACTQFG
 7975
 7976  040146  012701  000010          ACTMOS: MOV     #MOCHK,R1       ;MODEM BIT      ADDED BY EC
 7977  040152  000402                  BR      ACTQFG
 7978
 7979  040154  012701  000040          ACTPRO: MOV     #PROTOB,R1      ;SET THE PROTOCOL BIT
 7980
 7981  040160  050137  007476          ACTQFG: BIS     R1,PARAM
 7982  040164  022737  000036  003204          CMP     #NO,QUALFG
 7983  040172  001002                  BNE     1$
 7984  040174  040137  007476          BIC     R1,PARAM
 7985  040200  005037  003204          1$:     CLR     QUALFG                  ;CLEAR 'NO' OUT OF QUALIFIER FLAG
 7986  040204  000450                  BR      ACTLXX
 7987
 7988  040206  013737  003554  007500  ACTRPS: MOV     P$NUM,RPASS             ;GET NUMBER OF 'RUN PASSES'
 7989  040214  000444                  BR      ACTLXX
 7990
 7991  040216  012737  000005  007472  ACTMOP: MOV     #5,MLTYP
 7992  040224  000417                  BR      ACTLPX
 7993  040226  012737  000001  007472  ACTTLP: MOV     #1,MLTYP
 7994  040234  000413                  BR      ACTLPX
 7995  040236  012737  000002  007472  ACTCLP: MOV     #2,MLTYP
 7996  040244  000407                  BR      ACTLPX
 7997  040246  012737  000003  007472  ACTLLP: MOV     #3,MLTYP
 7998  040254  000403                  BR      ACTLPX
 7999  040256  012737  000004  007472  ACTRLP: MOV     #4,MLTYP
 8000
 8001  040264  022737  000003  007470  ACTLPX: CMP     #ACT,MODTYP     ;BE SURE IN ACTIVE IF TRYING TO SET LOOP
 8002  040272  001415                  BEQ     ACTLXX          ; BR IF IN ACTIVE
 8003  040274  112737  177777  003561          MOVB    #-1,P$GDBD
```

```
 8004  040302  005037  007472          CLR     MLTYP           ;CLEAR ANY LOOP TYPE THAT MAY HAVE GOT SET
 8005  040306                          PRINTF  #CLIBDL
  (7)  040306  012746  012627                                                  MOV     #CLIBDL,-(SP)
  (6)  040312  012746  000001                                                  MOV     #1,-(SP)
  (3)  040316  010600                                                          MOV     SP,R0
  (4)  040320  104417                                                          TRAP    C$PNTF
  (4)  040322  062706  000004                                                  ADD     #4,SP
 8006  040326  105037  003560  ACTLXX: CLRB    P$NNUF          ;CLEAR NOT-ENOUGH FLAG
 8007  040332  000207                  RTS     PC
 8008
```

```
 8010
 8011                                          ;REV B BY EC
 8012  040334  005737  007472        GTR9:    TST     MLTYP           ;LOOP MODE ?
 8013  040340  001422                         BEQ     10$             ;NO,BRANCH
 8014  040342  032737  000002  007476         BIT     #DATCKB,PARAM   ;DATA CHECK ?
 8015  040350  001416                         BEQ     10$             ;NO,BRANCH
 8016  040352  023737  007340  007354         CMP     CMPTOT,TXMTOT   ;TX & CX EQUAL
 8017  040360  001412                         BEQ     10$             ;YES,BRANCH
 8018  040362                                 PRINTF  #CLIPW          ;PRINT WARNING
   (7) 040362  012746  013032                                                    MOV     #CLIPW,-(SP)
   (6) 040366  012746  000001                                                    MOV     #1,-(SP)
   (3) 040372  010600                                                            MOV     SP,R0
   (4) 040374  104417                                                            TRAP    C$PNTF
   (4) 040376  062706  000004                                                    ADD     #4,SP
 8019  040402  000137  035310                 JMP     GETCL           ;TRY AGAIN
 8020
 8021                                          ;RX ALLOCATE CODE
 8022  040406  012701  006562        10$:     MOV     #PTRTAB,R1      ;INIT TRANSMIT MESSAGE POINTER
 8023  040412  010137  007334                 MOV     R1,TXPTR
 8024  040416  012702  000017                 MOV     #MSGLIM,R2
 8025  040422  006302                         ASL     R2
 8026  040424  006302                         ASL     R2
 8027  040426  010137  007336                 MOV     R1,CMPPTR
 8028  040432  060237  007336                 ADD     R2,CMPPTR       ;INIT COMPARE MESSAGE POINTER
 8029  040436  013701  007336                 MOV     CMPPTR,R1
 8030  040442  012702  000017                 MOV     #MSGLIM,R2
 8031  040446  006302                         ASL     R2
 8032  040450  006302                         ASL     R2
 8033  040452  010137  007332                 MOV     R1,RXPTR
 8034  040456  060237  007332                 ADD     R2,RXPTR        ;INIT RECEIVE MESSAGE POINTER
 8035
 8036  040462  013737  007340  007370         MOV     CMPTOT,RXMTOT   ;MAKE COMPARE AND RX MESSAGE COUNTS EQUAL
 8037
 8038
 8039  040470  005037  007502        GTREX:   CLR     FLAG            ;CLEAR FLAG
 8040  040474  005037  007374                 CLR     NOBUF           ;CLEAR NO BUFFER COUNTER
 8041  040500  005037  007376                 CLR     PSCNT           ;CLEAR PASS COUNT
 8042  040504  005037  007400                 CLR     ERRCNT          ;CLEAR ERROR COUNT
 8043  040510  005037  007372                 CLR     LNCNT           ;CLEAR LINE COUNTER
 8044
 8045  040514  004737  026452                 JSR     PC,LOGDVI       ;LOG ABOUT TO INIT DEVICE
 8046  040520  004737  043320                 JSR     PC,DVINIT       ;INIT DEVICE
 8047
 8048  040524  012737  001000  007412  GTRX2:  MOV     #BUFLIM,CURCC   ;SET CHAR COUNT TO 'BUFLIM'' NO. OF BYTES
 8049  040532  012737  004562  007420         MOV     #RXBUF,CURADD   ;SET UP RX BUFFER AS CURRRENT ADD.
 8050  040540  013737  007332  007416         MOV     RXPTR,CPTR
 8051  040546  012737  000010  007410         MOV     #10,MSGTYP      ;SET UP FOR 33 TO FILL RX BUFFERS
 8052  040554  004737  032010                 JSR     PC,BLDBUF       ;CLEAR RX BUFFER
 8053  040560  013702  007470                 MOV     MODTYP,R2
 8054  040564  006302                         ASL     R2
 8055  040566  000172  007504                 JMP     @MODE(R2)       ;MODE DISPATCH
 8056
```

```
                                        .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
                                        ;--

040572                                  RXONLY:
040572  013737  007332  007414          RXON2:  MOV     RXPTR,CPTRR
040600  013737  007370  007366                  MOV     RXMTOT,DVRCT    ;SET UP MESSAGE COUNT
040606  052737  000104  007502                  BIS     #QRX+#ERX,FLAG  ;SET UP RX QUE
040614  005037  007416                          CLR     CPTR            ;CLEAR THE TX POINTER
040620  000137  040762                          JMP     ALLTR           ;GO RX.
```

```
                                        .SBTTL          TRANSMIT MODE SECTION

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

040624  042737  000002  007476  TXONLY: BIC     #DATCKB,PARAM   ;SET NOCHECK
040632  013737  007334  007416  TXON2:  MOV     TXPTR,CPTR
040640  013737  007354  007352          MOV     TXMTOT,DVTCT    ;COPY COUNTER FOR THIS PASS
040646  052737  000210  007502          BIS     #QTX+#ETX,FLAG  ;SET THE QUE TX FLAG
040654  005037  007414                  CLR     CPTRR           ;CLEAR RX POINTER
040660  000137  040762                  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
                                ;--

040664                          PLCK:
040664  013737  007354  007352 PLCK2:  MOV     TXMTOT,DVTCT    ;SET UP  THE TRANSMIT COUNT
040672  013737  007334  007416         MOV     TXPTR,CPTR      ;SET UP CPTR TO TRANSMIT POINTER
040700  013737  007332  007414 PLCK3:  MOV     RXPTR,CPTRR     ;SET UP CPTRR TO REC POINTER
040706  052737  000104  007502         BIS     #QRX+#ERX,FLAG  ;SET UP Q AND EXPECT RX
040714  000137  040762                 JMP     ALLTR           ;AND GO RX FIRST MSG.
```

```
 8131                                   .SBTTL          ACTIVE MODE SECTION
 8132
 8133                                   ;++
 8134                                   ; FUNCTIONAL DESCRIPTION:
 8135                                   ;       ACTIVE MODE SECTION
 8136                                   ;       IN THIS MODE OF TESTING A LIST OF MESSAGES IS TRANSMITTED AND
 8137                                   ;       MESSAGES ARE EXPECTED TO BE RECEIVED.  RECEIVED DATA CAN BE COMPARED
 8138                                   ;       AGAINST "EXPECTED" DATA IF DATA-CHECKING IS ENABLED.
 8139                                   ;               NOTE: IF BOTH ENDS OF THE LINK ARE IN ACTIVE MODE, THEN THE
 8140                                   ;                       LINK MUST BE A FULL DUPLEX LINK!
 8141                                   ;
 8142                                   ; SUBORDINATE ROUTINES USED:
 8143                                   ;
 8144                                   ;               "ALLTR"
 8145                                   ;
 8146                                   ; CALLING SEQUENCE:
 8147                                   ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
 8148                                   ;--
 8149
 8150  040720  013737  007354  007352  ALCK:   MOV     TXMTOT,DVTCT
 8151  040726  013737  007334  007416          MOV     TXPTR,CPTR      ;SET UP TX COUNTS
 8152  040734  013737  007370  007366          MOV     RXMTOT,DVRCT    ;SET UP COUNTS
 8153  040742  013737  007332  007414          MOV     RXPTR,CPTRR
 8154  040750  052737  000314  007502          BIS     #QRX+#QTX+#ETX+#ERX,FLAG
 8155  040756  000137  040762                  JMP     ALLTR
 8156
 8157
 8158
```

```
                                .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
                                ;--

040762                          ALLTR:
040762  032737  000004  007502  ALCK5:  BIT     #QRX,FLAG
040770  001420                  BEQ     ALCK1           ;IF NOT RX GO TO TX'S
040772  013702  007414          MOV     CPTRR,R2
040776  011237  007432          MOV     (R2),TEMP2
041002  012237  007362          MOV     (R2)+,DVRXA
041006  011237  007434          MOV     (R2),TEMP3
041012  011237  007364          MOV     (R2),DVRCC
041016  010237  007414          MOV     R2,CPTRR
041022  004737  044040          JSR     PC,DVRXQ        ;GO QUE DEVICE
041026  004737  026406          JSR     PC,LOGRXQ       ;LOG REC QUED
041032  032737  000010  007502  ALCK1:  BIT     #QTX,FLAG
```

```
 8216  041040  001416                          BEQ     ALCK2           ;IF NO TX'S GO TO 2
 8217  041042  013702  007416                  MOV     CPTR,R2
 8218  041046  011237  007432                  MOV     (R2),TEMP2
 8219  041052  012237  007346                  MOV     (R2)+,DVTXA
 8220  041056  011237  007434                  MOV     (R2),TEMP3
 8221  041062  012237  007350                  MOV     (R2)+,DVTCC
 8222  041066  010237  007416                  MOV     R2,CPTR
 8223  041072  004737  026352                  JSR     PC,LOGTXQ
 8224
 8225  041076  004737  044120          ALCK2:  JSR     PC,DVTXRX       ;GO TO TX AND RX SUB ROUT.
 8226
 8227  041102  032737  000004  007502          BIT     #QRX,FLAG       ;CHECK FOR REC. MSG.
 8228  041110  001514                          BEQ     ALCK3
 8229  041112  013737  007362  007432          MOV     DVRXA,TEMP2
 8230  041120  013737  007364  007434          MOV     DVRCC,TEMP3
 8231  041126  004737  026424                  JSR     PC,LOGRXC       ;LOG REC COMPLETE
 8232  041132  032737  000004  007476  UPTABL: BIT     #ECHOB,PARAM    ;IS THIS ECHO MODE(PASSIVE)
 8233  041140  001406                          BEQ     UPTA4           ;IF NOT GO TO 4
 8234  041142  013702  007416                  MOV     CPTR,R2         ;ELSE SET R2 TO PRESENT TX TABL
 8235  041146  013722  007432                  MOV     TEMP2,(R2)+     ;STORE OFF RX ADD
 8236  041152  013712  007434                  MOV     TEMP3,(R2)      ;AND CC
 8237  041156  032737  000002  007476  UPTA4:  BIT     #DATCKB,PARAM   ;IS DATA CHECKING ASKED FOR
 8238  041164  001015                          BNE     UPTA1           ;IF SO GO TO 1
 8239  041166  012737  000001  007366          MOV     #01,DVRCT       ;ELSE SET DVRCT TO A 1
 8240  041174  013737  007332  007414          MOV     RXPTR,CPTRR     ;RESET POINTER
 8241  041202  022737  000003  007470          CMP     #ACT,MODTYP     ;IS THIS ACTIVE
 8242  041210  001002                          BNE     UPTA3
 8243  041212  005237  007366                  INC     DVRCT           ;IF YES BUMP COUNT
 8244  041216  000424                  UPTA3:  BR      UPTEX
 8245  041220  013702  007414          UPTA1:  MOV     CPTRR,R2
 8246  041224  011237  007426                  MOV     (R2),TEMP       ;LOAD TEMP WITH PREV. COUNT
 8247  041230  163737  007434  007426          SUB     TEMP3,TEMP      ;LOAD TEMP WITH PREV.COUNT-CURRENT
 8248  041236  013722  007434                  MOV     TEMP3,(R2)+
 8249  041242  063737  007434  007432          ADD     TEMP3,TEMP2
 8250  041250  013722  007432                  MOV     TEMP2,(R2)+     ;STORE OF NEW ADD
 8251  041254  013712  007426                  MOV     TEMP,(R2)       ;AND NEW CC
 8252  041260  162702  000002                  SUB     #2,R2           ;PUT POINTER BACK TO ADDR.
 8253  041264  010237  007414                  MOV     R2,CPTRR        ;AND RESTORE IT.
 8254  041270                          UPTEX:
 8255  041270  022737  000002  007470          CMP     #PAS,MODTYP
 8256  041276  001007                          BNE     ALCK2A          ;IF NOT PASSIVE LOOP THEN GO TO 2A
 8257  041300  042737  000104  007502          BIC     #QRX+#ERX,FLAG  ;CLEAR BOTH EXPECTED AND COMPLETED FLAGS
 8258  041306  052737  000210  007502          BIS     #QTX+#ETX,FLAG  ;SET THE TX FLAGS
 8259  041314  000646                          BR      ALCK1
 8260
 8261  041316  005337  007366          ALCK2A: DEC     DVRCT           ;DEC REC COUNT
 8262  041322  005737  007366                  TST     DVRCT           ;IS IT ALL DONE
 8263  041326  001005                          BNE     ALCK3           ;NO. GO CHECK TX
 8264  041330  042737  000004  007502          BIC     #QRX,FLAG       ;CLEAR THE RX FLAG
 8265  041336  005037  007414                  CLR     CPTRR           ;YES. CLEAR POINTER
 8266  041342  032737  000010  007502  ALCK3:  BIT     #QTX,FLAG       ;IS IT TX
 8267  041350  001447                          BEQ     ALCK4           ;IF NOT TX THEN GO BACK
 8268  041352  013737  007346  007432          MOV     DVTXA,TEMP2
 8269  041360  013737  007350  007434          MOV     DVTCC,TEMP3     ;LOG TX COMPLETED
 8270  041366  004737  026370                  JSR     PC,LOGTXC
 8271  041372  005337  007352                  DEC     DVTCT           ;DEC TX COUNT
```

```
 8272  041376  022737  000002  007470          CMP     #PAS,MODTYP
 8273  041404  001013                          BNE     ALCK3A          ;IF NOT PASSIVE MODE GO TO 3A
 8274  041406  042737  000210  007502          BIC     #QTX+ETX,FLAG   ;CLEAR THE TX FLAGS
 8275  041414  052737  000104  007502          BIS     #QRX+ERX,FLAG   ;AND SET THE RX FLAGS
 8276  041422  005737  007352                  TST     DVTCT
 8277  041426  001005                          BNE     ALCK3C          ;IF MORE RX'S DO IT
 8278  041430  000137  041510                  JMP     CMPSR           ; ELSE COMPARE
 8279  041434  005737  007352          ALCK3A: TST     DVTCT           ;IS IT ALL DONE
 8280  041440  001402                          BEQ     ALCK3B          ;IF NOT GO BACK TO 5
 8281  041442  000137  040762          ALCK3C: JMP     ALCK5
 8282  041446  005037  007416          ALCK3B: CLR     CPTR            ;IF SO CLEAR POINTER
 8283  041452  042737  000010  007502          BIC     #QTX,FLAG       ;CLEAR TX FLAG
 8284  041460  032737  000002  007476          BIT     #DATCKB,PARAM   ;IS IT DAT CK
 8285  041466  001403                          BEQ     ALCK4A          ;IF NOT THEN END WO CKING RX.
 8286  041470  005737  007414          ALCK4:  TST     CPTRR
 8287
 8288  041474  001362                          BNE     ALCK3C          ;IF SOME RX'S LEFT GO BACK
 8289  041476  005737  007416          ALCK4A: TST     CPTR
 8290  041502  001402                          BEQ     ALCK4B          ;BRANCH IF ANY TX'S LEFT
 8291  041504  000137  041076                  JMP     ALCK2
 8292  041510                          ALCK4B:
 8293
 8294
 8295
```

```
                                .SBTTL          DATA COMPARISON CODE


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

041510  032737  000002  007476  CMPSR:  BIT     #DATCKB,PARAM   ;IS DATA CHECKING TO BE DONE
041516  001522                          BEQ     CMPSEX          ;IF NOT THEN EXIT
041520  013737  007332  007416          MOV     RXPTR,CPTR      ;PUT START OF RX POINTERS TO CPTR
041526  013737  007336  007414          MOV     CMPPTR,CPTRR    ; AND START OF COMPARE POINTS TO CPTRR
041534  013737  007370  007366          MOV     RXMTOT,DVRCT

041542                          CMPS3:
041542  013702  007416                  MOV     CPTR,R2         ;MOVE CURRET RX PT.TO R2
041546  011237  007432                  MOV     (R2),TEMP2      ;MOVE RX ADD TO EVENT LOG
041552  012201                          MOV     (R2)+,R1        ;SET R1 TO START ADD OF RX
041554  012237  007434                  MOV     (R2)+,TEMP3     ;SET CHAR  COUNT TO EVENT LOG
041560  010237  007416                  MOV     R2,CPTR         ;RESTORE RX POINT

041564  013702  007414                  MOV     CPTRR,R2        ;PUT R2 AT COMPARE TABLE
041570  012203                          MOV     (R2)+,R3        ;SET R3 TO COMPARE ADD
041572  012204                          MOV     (R2)+,R4        ;SET R4 TO COMP CC
041574  010237  007414                  MOV     R2,CPTRR        ;RESTORE POINTER
041600  010437  007436                  MOV     R4,TEMP4
041604  004737  026520                  JSR     PC,LOGCMP       ;LOG COMPARE START.

041610  020437  007434                  CMP     R4,TEMP3        ;IS COMPARE COUNT = TO RX COUNT
041614  001410                          BEQ     CMPS7           ;IF SO GO TO 7
041616  005237  007400                  INC     ERRCNT
041622                                  ERRSOFT 1,EDDLE,ERR10   ;PRINT ERROR
```

```
  (4)  041622  104457                                                          TRAP    C$ERSOFT
  (5)  041624  000001                                                          .WORD   1
  (5)  041626  016046                                                          .WORD   EDDLE
  (5)  041630  025720                                                          .WORD   ERR10
8353   041632  004737  026536              JSR     PC,LOGCML       ;LOG LENGTH ERROR
8354
8355   041636  005037  007436      CMPS7:  CLR     TEMP4           ;CLEAR BAD BYTE COUNTER
8356   041642  012737  000001  007424      MOV     #1,OFSET        ;SET OFSET BYTE COUNT TO 1
8357   041650  122123              CMPS1:  CMPB    (R1)+,(R3)+     ;COMPARE RX WITH EXPETED
8358   041652  001422                      BEQ     CMPS6           ;IF EQUAL THEN GO TO 6
8359
8360   041654  005237  007436      CMPS2:  INC     TEMP4           ;INC BAD COUNT
8361   041660  023727  007436  000005      CMP     TEMP4,#5        ;IS IT MORE THEN 5
8362   041666  101014                      BHI     CMPS6           ;IF SO GO  FOR MORE
8363   041670  114337  007446              MOVB    -(R3),GOOD      ;STORE GOOD BYTE FOR ERROR
8364   041674  114137  007447              MOVB    -(R1),BAD       ;STORE BAD BYTE FOR ERROR
8365   041700  005237  007400              INC     ERRCNT
8366   041704                              ERRSOFT 2,EDDDE,ERR1    ;REPORT COMPARISON FAILURE TO OPR.
  (4)  041704  104457                                                          TRAP    C$ERSOFT
  (5)  041706  000002                                                          .WORD   2
  (5)  041710  016103                                                          .WORD   EDDDE
  (5)  041712  025630                                                          .WORD   ERR1
8367   041714  005201                      INC     R1
8368   041716  005203                      INC     R3
8369   041720  005237  007424      CMPS6:  INC     OFSET           ;INC OFFSET
8370   041724  005304                      DEC     R4              ;ELSE DEC CHAR COUNT AND SEE IF 0
8371   041726  001350                      BNE     CMPS1           ;IF NOT GO BACK
8372   041730  005737  007436              TST     TEMP4           ;SEE IF ANY CMP ERRS FOR THIS MSG
8373   041734  001410                      BEQ     CMPS5A          ;BR IF NONE
8374   041736  005237  007400              INC     ERRCNT
8375   041742                              ERRSOFT 3,EDDDE,ERR2    ;REPORT # OF MISMATCHES FOR MESSAGE
  (4)  041742  104457                                                          TRAP    C$ERSOFT
  (5)  041744  000003                                                          .WORD   3
  (5)  041746  016103                                                          .WORD   EDDDE
  (5)  041750  025672                                                          .WORD   ERR2
8376   041752  004737  026554      CMPS5:  JSR     PC,LOGCMD       ;LOG DATA ERROR IN COMPARE
8377   041756                      CMPS5A:
8378   041756  005337  007366              DEC     DVRCT
8379   041762  001267                      BNE     CMPS3           ;IF NOT ALL DONE GO BACK
8380
```

```
                                        .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
                                        ;------

041764  005237  007376          CMPSEX: INC     PSCNT           ;BUMP PASS COUNT

041770  013737  007374  007436          MOV     NOBUF,TEMP4
041776  013737  007376  007432          MOV     PSCNT,TEMP2
042004  013737  007400  007434          MOV     ERRCNT,TEMP3
042012  004737  026572                  JSR     PC,LOGEOP       ;LOG END OF PASS
042016  022737  177777  007500  5$:     CMP     #-1,RPASS       ;SEE IF RPASS=-1
042024  001403                          BEQ     1$              ;IF IT IS DON'T DECRMNT, LOOP FOREVER
042026  005337  007500                  DEC     RPASS           ;DEC PASS COUNT
042032  001402                          BEQ     2$              ;IF DONE EXIT TEST
042034  000137  040524          1$:     JMP     GTRX2           ;ELSE GO BACK AND DISPATCH
042040  004737  045714          2$:     JSR     PC,DVBTUP       ;GO UPDATE BASE TABLE
042044  000137  035224                  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
                                ;--

042050                          DLL:    GMANID  DLLQ1,TEMP3,0,377,0,377,NO      ;GET PASSWORD
(3)  042050  104443                                                             TRAP    C$GMAN
(3)  042052  000406                                                             BR      10001$
(4)  042054  007434                                                             .WORD   TEMP3
(5)  042056  000022                                                             .WORD   T$CODE
(5)  042060  013122                                                             .WORD   DLLQ1
(5)  042062  000377                                                             .WORD   377
(5)  042064  000000                                                             .WORD   T$LOLIM
(5)  042066  000377                                                             .WORD   T$HILIM
(3)  042070                                                                     10001$:
042070  113737  007434  002650          MOVB    TEMP3,PASS1                     ;PUT PASSWORD IN MESSAGE
042076  113737  007434  002651          MOVB    TEMP3,PASS2                     ;PASSWORD IS DUPLICATE
042104  113737  007434  002652          MOVB    TEMP3,PASS3                     ;;HERE
042112  113737  007434  002653          MOVB    TEMP3,PASS4                     ;;;AND HERE.
042120  052737  000100  007502          BIS     #ERX,FLAG       ;SET EXPECTED TO RX
042126  042737  000002  007476          BIC     #DATCKB,PARAM   ;CLEAR NOCHECK
042134  012737  002647  007420          MOV     #DLLM1,CURADD   ;SET THE DOWN LINE LOAD MSG TO #1
042142  013737  002172  007412          MOV     DLLM1C,CURCC    ;SET THE CC
042150  004737  042242                  JSR     PC,DLTXRX       ;GO TO THE DOWN LINE TX RX ROUTINE

                                ;RETURN WHEN TX AND RX ARE COMPLETED

042154  012737  002654  007420          MOV     #DLLM2,CURADD   ;SET THE DOWN LINE LOAD MSG TO #2
042162  013737  002174  007412          MOV     DLLM2C,CURCC    ;SET CC
042170  042737  000400  007502          BIC     #DLLGA,FLAG     ;CLEAR THE GO AHEAD FLAG
042176  004737  042242                  JSR     PC,DLTXRX       ;GO TO THE DOWN LINE TX RX ROUTINE

                                ; RETURN WHEN TX AND RX ARE COMPLETED
042202                          DLLPRI:
```

```
 8461  042202                                    PRINTF  #DLLCM
  (7)  042202  012746  015030                                                    MOV     #DLLCM,-(SP)
  (6)  042206  012746  000001                                                    MOV     #1,-(SP)
  (3)  042212  010600                                                            MOV     SP,R0
  (4)  042214  104417                                                            TRAP    C$PNTF
  (4)  042216  062706  000004                                                    ADD     #4,SP
 8462  042222  000137  035224                    JMP     GTRA5
 8463
 8464  042226                              DLLEA:
 8465  042226                                    ERRHRD  20,DLLAB,ERR14
  (4)  042226  104456                                                            TRAP    C$ERHRD
  (5)  042230  000024                                                            .WORD   20
  (5)  042232  025214                                                            .WORD   DLLAB
  (5)  042234  026142                                                            .WORD   ERR14
 8466
 8467  042236  000137  035224                    JMP     GTRA5           ;PRINT ABORT AND EXIT
 8468
 8469
 8470
 8471  042242                              DLTXRX:
 8472  042242  052737  000004  007502              BIS     #QRX,FLAG       ;SET THE QUE RX FLAG
 8473  042250  012737  004562  007362              MOV     #RXBUF,DVRXA    ;SET THE DEVICE RX BUFFER TO RXBUF
 8474  042256  012737  004562  007432              MOV     #RXBUF,TEMP2    ;SET UP FOR LOG
 8475  042264  012737  000400  007364              MOV     #256.,DVRCC     ;SET UP FOR CC OF 256
 8476  042272  012737  000400  007434              MOV     #256.,TEMP3     ;SET UP FOR LOG
 8477  042300  004737  044040                      JSR     PC,DVRXQ        ; GO QUE RX
 8478  042304  004737  026406                      JSR     PC,LOGRXQ       ;AND LOG IT...
 8479
 8480  042310  013737  007420  007346              MOV     CURADD,DVTXA    ;SET UP FOR TX
 8481  042316  013737  007420  007432              MOV     CURADD,TEMP2    ;AND LOG
 8482  042324  013737  007412  007350              MOV     CURCC,DVTCC     ;SE UP FOR TX COUNT
 8483  042332  013737  007412  007434              MOV     CURCC,TEMP3     ;AND LOG IT
 8484  042340  004737  026352                      JSR     PC,LOGTXQ       ;LOG THE TX QUEUED
 8485  042344  052737  000210  007502              BIS     #QTX+#ETX,FLAG  ;SET UP TO QUE AND EXPECTED
 8486  042352  004737  044120              DLLE2:  JSR     PC,DVTXRX       ;GO TO DEVICE ROUTINE
 8487  042356  032737  000400  007502              BIT     #DLLGA,FLAG     ;TEST FOR GO AHEAD BIT
 8488  042364  001047                              BNE     DLLE1           ;IF SET GO TO ONE
 8489  042366  032737  000010  007502              BIT     #QTX,FLAG       ;ELSE CHECK FOR TX DONE
 8490  042374  001020                              BNE     DLLE6           ;IF DONE THEN BRANCH
 8491                                                                      ;ELSE ERROR
 8492  042376  012737  025521  007442              MOV     #TXNC,CONOTM
 8493  042404  013737  004562  007434  DLLE7:     MOV     RXBUF,TEMP3
 8494  042412  013737  003562  007436              MOV     TXBUF,TEMP4
 8495  042420  012737  025214  007432              MOV     #DLLAB,TEMP2
 8496  042426  004737  026434                      JSR     PC,LGDVE        ;LOG ERROR
 8497  042432  000137  042226                      JMP     DLLEA           ;ABORT TEST
 8498
 8499  042436  013737  007346  007432  DLLE6:     MOV     DVTXA,TEMP2
 8500  042444  013737  007350  007434              MOV     DVTCC,TEMP3
 8501  042452  004737  026370                      JSR     PC,LOGTXC       ;LOG TX DONE
 8502  042456  042737  000210  007502              BIC     #QTX+#ETX,FLAG  ;CLEAR QUE AND EXPECTED
 8503  042464  052737  000400  007502              BIS     #DLLGA,FLAG     ;SET THE GO AHEAD BIT
 8504  042472  023737  002174  007350              CMP     DLLM2C,DVTCC
 8505  042500  001472                              BEQ     DLLE5           ;EXIT IF SECOND MSG.
 8506  042502  000723                              BR      DLLE2           ;AND GO BACK TO 2
 8507  042504  032737  000004  007502  DLLE1:     BIT     #QRX,FLAG       ;IS THE A RX COMPLETED
```

```
 8508  042512  001004                          BNE     DLLE8           ;IF SO GO TO 8
 8509  042514  012737  025541  007442          MOV     #RXNC,CONOTM    ;ELSE SET UP ERROR AND ABORT.
 8510  042522  000730                          BR      DLLE7
 8511  042524  013737  007362  007432  DLLE8:  MOV     DVRXA,TEMP2
 8512  042532  013737  007364  007434          MOV     DVRCC,TEMP3
 8513  042540  004737  026424                  JSR     PC,LOGRXC       ;LOG RECEIVE COMPLETE
 8514  042544  122737  000010  004562          CMPB    #10,RXBUF       ;CHECK FOR FIRST WORD OF RX
 8515                                          ;SEC BOOT MSG.
 8516  042552  001404                          BEQ     DLLE3
 8517  042554  012737  025561  007442  DLLE4:  MOV     #RXM1,CONOTM    ;SET UP MESG AND ABORT
 8518  042562  000710                          BR      DLLE7           ;ABORT TEST
 8519
 8520  042564  122737  000001  004564  DLLE3:  CMPB    #1,RXBUF+2      ;IS SECOND WORD 1?
 8521  042572  001404                          BEQ     DLLE5A          ;YES,BRANCH
 8522  042574  012737  025604  007442          MOV     #RXM2,CONOTM
 8523  042602  000700                          BR      DLLE7           ;SET UP MESSAGE AND ABORT
 8524
 8525                                  ;;PRINT ID OF DEVICE REQUESTING LOAD    REV B BY EC
 8526  042604  012737  020273  007426  DLLE5A: MOV     #UNKM,TEMP      ;SET UP FOR UNKNOWN DEVICE
 8527  042612  113703  004563                  MOVB    RXBUF+1,R3      ;GET DEVTYPE FROM MESSAGE
 8528  042616  120327  000042                  CMPB    R3,#34.         ;OUT OF LEGAL RANGE ?
 8529  042622  101006                          BHI     DLLE5B          ;YES,BRANCH
 8530  042624  132703  000001                  BITB    #1,R3           ;ODD ?
 8531  042630  001003                          BNE     DLLE5B          ;YES,BRANCH
 8532  042632  016337  010542  007426          MOV     DLLIND(R3),TEMP ;GET ASCIZ MESSAGE FROM TABLE
 8533
 8534  042640                          DLLE5B: PRINTF  #SECRM,TEMP,R3  ;PRINT ID MESSAGE
   (9) 042640  010346                                                          MOV     R3,-(SP)
   (8) 042642  013746  007426                                                  MOV     TEMP,-(SP)
   (7) 042646  012746  020135                                                  MOV     #SECRM,-(SP)
   (6) 042652  012746  000003                                                  MOV     #3,-(SP)
   (3) 042656  010600                                                          MOV     SP,R0
   (4) 042660  104417                                                          TRAP    C$PNTF
   (4) 042662  062706  000010                                                  ADD     #10,SP
 8535  042666  000207                  DLLE5:  RTS     PC              ;RETURN TO CALLER
 8536
 8537
 8538
```

```
 8540                                   .SBTTL          TALK MODE SECTION
 8541
 8542                                   ;++
 8543                                   ; FUNCTIONAL DESCRIPTION:
 8544                                   ;       TALK MODE SECTION
 8545                                   ;       IN THIS MODE, THE ''TALK'' END OF THE LINK TRANSMITS OPERATOR
 8546                                   ;       SPECIFIED MESSAGES UNTIL A ''EXIT'' MESSAGE IS TYPE.  AT THAT POINT,
 8547                                   ;       THIS END OF THE LINK GOES INTO ''LISTEN'' MODE.
 8548                                   ;
 8549                                   ; SUBORDINATE ROUTINES USED:
 8550                                   ;
 8551                                   ;               'LOGTXQ' - LOG TX BUFFER QUED TO EVENT LOG
 8552                                   ;               'DVTXRX'' - QUE TX BUFFER TO DEVICE AND WAIT FOR COMPLETE
 8553                                   ;               'LOGTXC'' - LOG TX COMPLETE TO EVENT LOG
 8554                                   ;
 8555                                   ; CALLING SEQUENCE:
 8556                                   ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
 8557                                   ;--
 8558
 8559  042670                           TALCK:
 8560  042670  042737  000002  007476          BIC     #DATCKB,PARAM   ;SET NOCHECK
 8561  042676  012702  002524                  MOV     #OPBUF,R2
 8562  042702  012722  177777          1$:     MOV     #-1,(R2)+       ;CLEAR OUT OPBUFFER FIRST
 8563  042706  022702  002646                  CMP     #OPEND,R2
 8564  042712  001373                          BNE     1$
 8565  042714                                  GMANID  OPRMM,OPBUF,A,0,1,72.,NO        ;GET TALK MESSAGE
   (3) 042714  104443                                                                      TRAP    C$GMAN
   (3) 042716  000406                                                                      BR      10002$
   (4) 042720  002524                                                                      .WORD   OPBUF
   (5) 042722  000142                                                                      .WORD   T$CODE
   (5) 042724  014764                                                                      .WORD   OPRMM
   (5) 042726  000000                                                                      .WORD   0
   (5) 042730  000001                                                                      .WORD   T$LOLIM
   (5) 042732  000110                                                                      .WORD   T$HILIM
   (3) 042734                                                                      10002$:
 8566  042734  005002                          CLR     R2              ;NOW GET CHAR COUNT
 8567  042736  122762  000377  002524  2$:     CMPB    #377,OPBUF(R2)
 8568  042744  001402                          BEQ     3$
 8569  042746  005202                          INC     R2
 8570  042750  000772                          BR      2$
 8571  042752  010237  002166          3$:     MOV     R2,OPCNT
 8572
 8573  042756  012737  002524  007346          MOV     #OPBUF,DVTXA    ;SET UP TX ADDR.
 8574  042764  012737  002524  007432          MOV     #OPBUF,TEMP2
 8575  042772  013737  002166  007434          MOV     OPCNT,TEMP3
 8576  043000  013737  002166  007350          MOV     OPCNT,DVTCC     ;SET UP TX CC
 8577  043006  004737  026352                  JSR     PC,LOGTXQ
 8578  043012  052737  000210  007502          BIS     #QTX+#ETX,FLAG  ;SET UP FLAGS
 8579  043020  005037  007414                  CLR     CPTRR           ;CLEAR RX POINTER
 8580
 8581  043024  004737  044120                  JSR     PC,DVTXRX
 8582
 8583  043030  013737  007346  007432          MOV     DVTXA,TEMP2
 8584  043036  013737  007350  007434          MOV     DVTCC,TEMP3
 8585  043044  004737  026370                  JSR     PC,LOGTXC
 8586  043050  022737  054105  002524          CMP     #'EX,OPBUF      ;CHECK FOR EXIT
```

```
 8587  043056  001304                    BNE     TALCK
 8588  043060  022737  052111  002526    CMP     #''IT,OPBUF+2
 8589  043066  001300                    BNE     TALCK
 8590  043070  042737  000210  007502    BIC     #QTX+#ETX,FLAG  ;CLEAR THE TX BITS
 8591  043076  012737  000006  007470    MOV     #LIS,MODTYP     ;CHANGE TO LISTEN MODE
 8592  043104  000137  040524            JMP     GTRX2           ;AND GO BACK TO DISPATCH
```

```
                                .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
                                ;--

043110  042737  000002  007476  LISCK:  BIC     #DATCKB,PARAM   ;CLEAR CHECK BIT
043116                                  PRINTF  #LISP           ;PRINT PROMPT FOR OPR.
(7)  043116  012746  014753                                                     MOV     #LISP,-(SP)
(6)  043122  012746  000001                                                     MOV     #1,-(SP)
(3)  043126  010600                                                             MOV     SP,R0
(4)  043130  104417                                                             TRAP    C$PNTF
(4)  043132  062706  000004                                                     ADD     #4,SP
043136  012737  002524  007362  LISCKA: MOV     #OPBUF,DVRXA    ;SET DEVICE UP TO REC AT OPBUF
043144  012737  002524  007432          MOV     #OPBUF,TEMP2
043152  012737  000122  007364          MOV     #82.,DVRCC      ;SET UP CHAR COUNT TO 82.
043160  012737  000122  007434          MOV     #82.,TEMP3
043166  052737  000104  007502          BIS     #QRX+#ERX,FLAG  ;SET UP FLAG
043174  005037  007416                  CLR     CPTR            ;CLEAR THE TX.

043200  004737  044040                  JSR     PC,DVRXQ        ;QUE RX
043204  004737  026406                  JSR     PC,LOGRXQ

043210  004737  044120                  JSR     PC,DVTXRX       ;GO TO DEVICE RX. SUBROUTINE

043214  013737  007362  007432          MOV     DVRXA,TEMP2
043222  013737  007364  007434          MOV     DVRCC,TEMP3     ;SET UP ADDR.AND CC.
043230  004737  026424                  JSR     PC,LOGRXC       ;LOG COMPLETED
043234  063737  007362  007364          ADD     DVRXA,DVRCC
043242  105077  144116                  CLRB    @DVRCC
043246                                  PRINTF  #OPBFPT
(7)  043246  012746  002520                                                     MOV     #OPBFPT,-(SP)
(6)  043252  012746  000001                                                     MOV     #1,-(SP)
(3)  043256  010600                                                             MOV     SP,R0
(4)  043260  104417                                                             TRAP    C$PNTF
(4)  043262  062706  000004                                                     ADD     #4,SP
043266  022737  054105  002524          CMP     #'EX,OPBUF      ;COMPARE FOR EX OF 'EXIT'
043274  001320                          BNE     LISCKA          ;IF NOT EXIT THEN GO BACK
043276  022737  052111  002526          CMP     #'IT,OPBUF+2    ;IF FIRST HALF OK CHECK NEXT PART
043304  001314                          BNE     LISCKA          ;IF NOT EXIT THE GO BACK
043306  012737  000005  007470          MOV     #TAL,MODTYP     ;CHANGE MODE TO TALK
043314  000137  040524                  JMP     GTRX2           ;RETURN TO DISPATCHER
```

```
8640
8641
```

```
 8643                                   .SBTTL          DEVICE FUNCTION SUBROUTINES
 8644
 8659
 8660
 8661
 8662                                   .SBTTL                  DEVICE INIT SUBROUTINE
 8663
 8681
 8682                                   ;++
 8683                                   ; FUNCTIONAL DESCRIPTION:
 8684                                   ;       DVINIT- DEVICE INIT ROUTINE
 8685                                   ;       THIS ROUTINE IS DEVICE DEPENDENT CODE THAT INITS
 8686                                   ;       THE DEVICE BEING TESTED. (I.E. FULL/HALF DUPLEX BAUD RATE, MAINT MODE.)
 8687                                   ;
 8688                                   ; INPUTS:       'FHDPLX'' INDICATES IF MODE IS FULL OR HALF DUPLEX. (1=FULL)
 8689                                   ;               ADDRESS POINTERS (SEL0,...) ALREADY POINT TO DEVICE'S REG.S
 8690                                   ;
 8691                                   ; SUBORDINATE ROUTINES USED:
 8692                                   ;
 8693                                   ;                'LGDVE'' - LOG DEVICE ERROR TO EVENT LOG
 8694                                   ;               ''TOORIO'' - TIME OUT OR INPUT INTERRUPT OR OUTPUT INTERRUPT
 8695                                   ;                ''CLRAW'' - CLEAR RQI AND WAIT FOR RDI TO GO AWAY
 8696                                   ;
 8697                                   ;
 8698                                   ; CALLING SEQUENCE:
 8699                                   ;               JSR     PC,DVINIT
 8700                                   ;--
 8701
 8702  043320                           DVINIT:
 8703                                           ;MASTER CLEAR DEVICE
 8704
 8705  043320  012737  000100  007542           MOV     #100,TIMER1     ;SET UP TIMER 1 FOR 100(OCTAL) TICKS
 8706  043326  005077  147042                   CLR     @SEL6
 8707  043332  005077  147032                   CLR     @SEL4
 8708  043336  012777  040000  147014           MOV     #MCLR,@SEL0     ;DO A MASTER CLEAR
 8709
 8710  043344  022737  000004  012406           CMP     #DMRC6,OPTYP    ;IS THIS A 8206
 8711  043352  001003                           BNE     DVIN6           ;IF NOT GO TO 6
 8712  043354  112777  000200  147000           MOVB    #200,@BSEL1     ;SET RUN FOR 8206
 8713  043362  022737  000006  012406  DVIN6:   CMP     #DMR6,OPTYP     ;IS THIS AN 8206 DMR
 8714  043370  001003                           BNE     DVIN2           ;IF NOT GO TO 2
 8715  043372  112777  000200  146762           MOVB    #200,@BSEL1     ;SET RUN BIT FOR 8206
 8716
 8717  043400  005777  146754          DVIN2:   TST     @SEL0           ;IS RUN BIT SET
 8718  043404  100426                           BMI     DVIN1           ;IF YES GO TO 1 ELSE...
 8719  043406                                   BREAK
   (3) 043406  104422                                                                   TRAP    C$BRK
 8720  043410  005737  007542                   TST     TIMER1          ;SEE IF TIME HAS EXPIRED
 8721  043414  001371                           BNE     DVIN2           ;IF NOT GO BACK AND CHECK
 8722                                                                   ;AGAIN ELSE...PRINT ERROR
 8723  043416  012737  024424  007432           MOV     #DVEM3,TEMP2
 8724  043424  017737  146730  007434           MOV     @SEL0,TEMP3
 8725  043432  017737  146726  007436           MOV     @SEL2,TEMP4     ;LOAD UP ERRM. AND REG OUTPUTS
 8726  043440  004737  026434                   JSR     PC,LGDVE        ;LOG TIME OUT WAITING FOR RUN
 8727  043444  005237  007400                   INC     ERRCNT
 8728  043450                                   ERRSOFT 11,DVEM3,ERR13
```

```
 (4)  043450  104457                                                          TRAP    C$ERSOFT
 (5)  043452  000013                                                          .WORD   11
 (5)  043454  024424                                                          .WORD   DVEM3
 (5)  043456  026110                                                          .WORD   ERR13
 8729
 8730  043460  000717                        BR DVINIT              ;GO BACK AND TRY MSTR CLR AGAIN IF ERROR
 8731
 8732  043462                        DVIN1:
 8733
 8734                                ; DO BASE IN COMMAND
 8735
 8736  043462  042737  000003  007502        BIC     #3,FLAG        ;CLEAR INPUT AND OUTPUT INT FLAGS
 8737  043470  112777  000143  146662        MOVB    #143,@BSEL0    ;SET UP BASE IN INT EN
 8738  043476  004737  044746               JSR     PC,TOORIO      ;GO WAIT FOR INTERRUPT OR TIME OUT
 8739  043502  012777  017370  146660        MOV     #BASE,@SEL4
 8740
 8741  043510  012777  000000  146656        MOV     #0,@SEL6       ;SET UP SEL 6
 8742  043516  023727  012406  000006        CMP     OPTYP,#6       ;IS THIS DMR MODE
 8743  043524  002403                        BLT     DVIN7          ;IF NOT GO TO 7
 8744  043526  012777  000522  146640        MOV     #522,@SEL6     ;SET DMR MODE
 8745  043534  052777  000100  146622  DVIN7: BIS     #IEO,@SEL2     ;SET IEO
 8746  043542  042777  004000  146610        BIC     #LULOOP,@SEL0  ;CLEAR LU LOOP
 8747  043550  022737  000001  007472        CMP     #TTL,MLTYP     ;IS TTL SELECTED
 8748  043556  001003                        BNE     DVIN3          ; IF NOT GO TO 3
 8749  043560  052777  004000  146572        BIS     #LULOOP,@SEL0  ;ELSE SET LU LOOP
 8750  043566  004737  044640        DVIN3: JSR     PC,CLRAW
 8751
 8752                                ; DO WRITE MODEM IF DMR MODE
 8753
 8754  043572  023727  012406  000006        CMP     OPTYP,#6       ;IS THIS DMR MODE
 8755  043600  002437                        BLT     DVIN8          ;IF NOT GO TO 8
 8756  043602  112777  000145  146550        MOVB    #145,@BSEL0    ;SET UP WRITE MODEM
 8757  043610  004737  044746               JSR     PC,TOORIO      ;GO TO  WAIT FOR INT
 8758  043614  042777  000014  146552        BIC     #BIT2+#BIT3,@SEL6              ;CLEAR BSEL6 AND 7
 8759  043622  022737  000004  007472        CMP     #MODREM,MLTYP  ;IS THIS REMOTE LOOP
 8760  043630  001003                        BNE     DVIN9          ;IF NOT GO TO 9
 8761  043632  052777  000004  146534        BIS     #BIT2,@BSEL6   ;SET THE BIT
 8762  043640  022737  000003  007472  DVIN9: CMP     #MODLOC,MLTYP  ;IS IT MODEM LOCAL
 8763  043646  001003                        BNE     DVIN10         ;IF NOT EXIT
 8764  043650  052777  000010  146516        BIS     #BIT3,@BSEL6   ;SET MODEM LOCAL
 8765  043656  004737  044640        DVIN10: JSR     PC,CLRAW       ;CLEAR RDI AND WAIT
 8766
 8767                                ; ENABLE EXTENDED ERROR IF DMR MODE
 8768
 8769
 8770  043662  112777  000146  146470        MOVB    #146,@BSEL0    ;SET UP FOR ENABLE
 8771  043670  004737  044746               JSR     PC,TOORIO
 8772  043674  004737  044640               JSR     PC,CLRAW       ;CLEAR RDI AND WAIT
 8773
 8774                                ; DO CONTROL IN COMMAND
 8775
 8776  043700  112777  000141  146452  DVIN8: MOVB    #141,@BSEL0    ;SET UP CONTROL IN
 8777  043706  004737  044746               JSR     PC,TOORIO      ;WAIT FOR INT OR TIME OUT
 8778  043712  005077  146456               CLR     @SEL6          ;CLEAR HALF/DUP
 8779  043716  022737  000004  007470        CMP     #DOW,MODTYP    ;IS THIS DOWN LINE LOAD?
 8780  043724  001004                        BNE     DVIN5          ; BR IF NOT
```

```
8781  043726  052777  002400  146440          BIS     #MAINTB+HALFDB,@SEL6    ;IF SO SET MAINT MODE  BIT
8782  043734  000406                          BR      DVIN4           ; AND FORCE HALF DUPLEX
8783
8784  043736  005737  007474          DVIN5:  TST     FHDPLX          ;IS THIS A HALF/DUP
8785  043742  001003                          BNE     DVIN4           ;IF NOT GO TO 4
8786  043744  052777  002000  146422          BIS     #HALFDB,@SEL6   ;ELSE SET HALF/DUP
8787
8788  043752  017737  146416  007444  DVIN4:  MOV     @SEL6,CONTIN    ;SET UP CONTROL IN FOR MODS
8789  043760  004737  044640                  JSR     PC,CLRAW        ;GO CLEAR RQI AND WAIT
8790                                                                  ;FOR RDI TO GO AWAY.
8791  043764  023727  012406  000006          CMP     OPTYP,#6        ;IS THIS DMR ?
8792  043772  002403                          BLT     DVINEX          ;NO,EXIT
8793  043774  052737  001000  007502          BIS     #DMRRUN,FLAG    ;SET DMRRUN OUTPUT EXPECTED BIT
8794  044002  000207                  DVINEX: RTS     PC              ;RETURN TO CALLER
8795
8796
8797
8798
8799
```

```
                                .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
                                ;--

044004  112777  000141  146346  DVMODS: MOVB    #141,@BSEL0     ;SET UP CONTORL IN
044012  004737  044746                  JSR     PC,TOORIO       ;GO TIME OUT CHECK
044016  017737  146346  010456          MOV     @SEL4,MODS      ;SET UP MODEM STATUS
044024  013777  007444  14  42          MOV     CONTIN,@SEL6    ;SET UP OLD CONTORL IN
044032  004737  044640                  JSR     PC,CLRAW
044036  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
                                ;--

044040                          DVRXQ:
044040  032737  000004  007502          BIT     #QRX,FLAG
044046  001423                        BEQ     DVREX           ;IF NOT RX THEN EXIT
                                                                ;ELSE QUE RX
044050  042737  000004  007502        BIC     #QRX,FLAG       ;CLEAR FLAG FOR RX
044056  112777  000144  146274        MOVB    #144,@BSEL0
044064  004737  044746                JSR     PC,TOORIO       ;GO CHECK FOR IN OR OUT
044070  017737  146274  010456        MOV     @SEL4,MODS      ;SET UP NEW MOD STATUS
044076  013777  007362  146264        MOV     DVRXA,@SEL4
044104  013777  007364  146262        MOV     DVRCC,@SEL6     ;LOAD CC AND ADDR
044112  004737  044640                JSR     PC,CLRAW        ;CLEAR AND WAIT
044116  000207                DVREX:  RTS     PC              ;RETURN TO CALLER
```

```
 8898                                   .SBTTL                  DEVICE TRANSMIT AND RECEIVE SUBROUTINE
 8899
 8928
 8929                                   ;++
 8930                                   ; FUNCTIONAL DESCRIPTION:
 8931                                   ;       DVTXRX-DEVICE TRANSMIT AND RECEIVE ROUTINE
 8932                                   ;       THIS CODE QUES THE TRANSMIT BUFFER TO THE DEVICE
 8933                                   ;       IF NEEDED. THE CODE THEN WAITS FOR A TX COMPLE,
 8934                                   ;       RX COMPLETE OR BOTH. THE CODE REPORTS A TIME OUT
 8935                                   ;       ERROR IF NO BACC OUTPUT INTERRUPT IS RECIEVED BEFORE
 8936                                   ;       60 SECONDS. AFTER REPORTING ERROR TIMER IS RE STARTED
 8937                                   ;       AND DEVICE WILL CONTINUE TO WAIT FOR INTERRUPT. CODE
 8938                                   ;       ALSO REPORTS ERROR IF INPUT INTERRUPT OCCURS WHEN
 8939                                   ;       EXPECTING OUTPUT INTERRUPT;WHEN RX BACC OCCURS WHEN
 8940                                   ;       EXPECTING TX,AND WHEN TX INT. OCCURS WHEN EXPECTING
 8941                                   ;       RECIEVE.
 8942                                   ;
 8943                                   ;INPUTS:
 8944                                   ;       'DVTXA' = ADDRESS OF TRANSMIT MSG.
 8945                                   ;       'DVTCC' = BYTE COUNT OF TRANSMIT MSG.
 8946                                   ;    'QTX' BIT  = SET IF TRANSMIT REQUESTED
 8947                                   ;    'ETX' BIT  = SET IF TRNASMIT EXPECTED
 8948                                   ;    'ERX' BIT  = SET IF RECIEVE EXPECTED
 8949                                   ;
 8950                                   ;OUTPUTS:
 8951                                   ;       'DVTXA' = ADDRESS OF TX MSG. COMPLETED
 8952                                   ;       'DVTCC' = BYTE COUNT OF TX MSG. COMPLETED
 8953                                   ;       'QTX'   = SET IF TX COMPLETED
 8954                                   ;       'DVRXA' = ADDRESS OF RX MSG. COMPLETED
 8955                                   ;       'DVRCC' = BYTE COUNT OF RX MSG. COMPLETED
 8956                                   ;       'QRX'   = SET IF RX COMPLETED
 8957                                   ;
 8958                                   ;
 8959                                   ; SUBORDINATE ROUTINES USED:
 8960                                   ;
 8961                                   ;               'TOORIO' - TIME OUT OR OUTPUT INTERRUPT OR INTPUT INTERRUPT
 8962                                   ;               'CLRAW'  - CLEAR RQI AND WAIT FOR RDI TO CLEAR
 8963                                   ;               'LGDVE'  - LOG DEVICE ERROR TO EVENT LOG
 8964                                   ;               'OUTHDL' - OUTPUT INTERRUPT HANDLER CODE
 8965                                   ;
 8966                                   ; CALLING SEQUENCE:
 8967                                   ;       JSR     PC,DVTXRX
 8968                                   ;--
 8969
 8970  044120  032737  000010  007502  DVTXRX: BIT     #QTX,FLAG       ;ANY TX TO QUE
 8971  044126  001423                          BEQ     DVTR3           ;IF NOT GO WAIT FOR OUPUT
 8972  044130  042737  000010  007502          BIC     #QTX,FLAG       ;CLEAR FLAG
 8973  044136  112777  000140  146214          MOVB    #140,@BSEL0
 8974  044144  004737  044746                  JSR     PC,TOORIO       ;GO CHECK FOR IN OR OUT
 8975  044150  017737  146214  010456          MOV     @SEL4,MODS      ;PUT IN NEW MOD STAT
 8976  044156  013777  007346  146204          MOV     DVTXA,@SEL4
 8977  044164  013777  007350  146202          MOV     DVTCC,@SEL6
 8978  044172  004737  044640                  JSR     PC,CLRAW        ;CLEAR RQUI ANDWAIT
 8979  044176                          DVTR3:
 8980  044176  012737  000074  007546          MOV     #60.,TIMERS     ;SET TIMER FOR 60 SECS
 8981  044204  032737  000060  007502  TOINOT: BIT     #CRX+#CTX,FLAG  ;IS IT TX OR RX COMP ALREADY?
```

```
 8982  044212  001071                          BNE     DVTR4           ;IS SO EXIT
 8983
 8984  044214  005737  007546                  TST     TIMERS          ;IS TIMER EXPIRED
 8985  044220  001022                          BNE     TOIN1
 8986  044222  012737  024510  007432          MOV     #DVEM4,TEMP2
 8987  044230  017737  146124  007434          MOV     @SEL0,TEMP3
 8988  044236  017737  146122  007436          MOV     @SEL2,TEMP4
 8989  044244  004737  026434                  JSR     PC,LGDVE
 8990  044250  005237  007400                  INC     ERRCNT
 8991  044254                                  ERRSOFT 12,DVEM4,ERR13
   (4) 044254  104457                                                      TRAP    C$ERSOFT
   (5) 044256  000014                                                      .WORD   12
   (5) 044260  024510                                                      .WORD   DVEM4
   (5) 044262  026110                                                      .WORD   ERR13
 8992  044264  000744                          BR      DVTR3           ;RETURN TO CHECK TIMER
 8993
 8994
 8995  044266                          TOIN1:  BREAK
   (3) 044266  104422                                                      TRAP    C$BRK
 8996  044270  032737  000001  007502          BIT     #ININT,FLAG     ;IS IT INPUT INTERRUPT
 8997  044276  001425                          BEQ     TOIN2           ;IF SO LOG ERROR
 8998
 8999  044300  012737  024602  007432          MOV     #DVEM5,TEMP2
 9000  044306  017737  146046  007434          MOV     @SEL0,TEMP3
 9001  044314  017737  146044  007436          MOV     @SEL2,TEMP4
 9002  044322  004737  026434                  JSR     PC,LGDVE
 9003  044326  042737  000001  007502          BIC     #ININT,FLAG     ;CLEAR BIT
 9004  044334  005237  007400                  INC     ERRCNT
 9005  044340                                  ERRSOFT 13,DVEM5,ERR13
   (4) 044340  104457                                                      TRAP    C$ERSOFT
   (5) 044342  000015                                                      .WORD   13
   (5) 044344  024602                                                      .WORD   DVEM5
   (5) 044346  026110                                                      .WORD   ERR13
 9006  044350  000715                          BR      TOINOT
 9007
 9008  044352  032737  000002  007502  TOIN2:  BIT     #OTINT,FLAG
 9009  044360  001711                          BEQ     TOINOT          ;IF NOT OUTPUT GO BACK AND
 9010                                                                    ;CHECK TIMER AGAIN
 9011  044362  004737  045070                  JSR     PC,OUTHDL       ;ELSE HANDLE OUTPUT AND RETURN
 9012  044366  032737  000060  007502          BIT     #CTX+#CRX,FLAG  ;IS IT TX OR RX
 9013  044374  001703                          BEQ     TOINOT          ;IF NOT GO BACK AND TRY AGAIN
 9014  044376  032737  000020  007502  DVTR4:  BIT     #CTX,FLAG       ;IS IT TX
 9015  044404  001440                          BEQ     DVTR5           ;IF NOT TRY RX
 9016  044406  032737  000200  007502          BIT     #ETX,FLAG       ;IF SO SHOULD IT BE
 9017  044414  001020                          BNE     DVTR4A          ;IF IT SHOULD GO TO 4A
 9018  044416  012737  025125  007432          MOV     #DVEM9,TEMP2    ;ELSE LOG ERROR
 9019  044424  013737  045702  007434          MOV     TSEL4,TEMP3
 9020  044432  013737  045704  007436          MOV     TSEL6,TEMP4
 9021  044440  004737  026434                  JSR     PC,LGDVE        ;
 9022  044444                                  ERRSOFT 14,DVEM9,ERR13  ;REPORT ERROR
   (4) 044444  104457                                                      TRAP    C$ERSOFT
   (5) 044446  000016                                                      .WORD   14
   (5) 044450  025125                                                      .WORD   DVEM9
   (5) 044452  026110                                                      .WORD   ERR13
 9023
 9024  044454  000411                          BR      DVTR4B          ;THEN CLEAR COMPL.FLAG
```

```
 9025  044456  013737  045702  007346  DVTR4A: MOV     TSEL4,DVTXA
 9026  044464  013737  045704  007350          MOV     TSEL6,DVTCC
 9027  044472  052737  000010  007502          BIS     #QTX,FLAG       ;AND SET TX COMPL FLAG
 9028  044500  042737  000020  007502  DVTR4B: BIC     #CTX,FLAG       ;ELSE CLEAR FLAG
 9029  044506  032737  000040  007502  DVTR5:  BIT     #CRX,FLAG       ;IS IT RX TOO?
 9030  044514  001440                          BEQ     DVTREX          ;IF NOT THEN EXIT.
 9031  044516  032737  000100  007502          BIT     #ERX,FLAG       ;TEST ;S THIS SUPPOSED TO BE RX
 9032  044524  001020                          BNE     DVTR5A          ;IF YES PROCESS AS SUCH
 9033  044526  012737  025036  007432          MOV     #DVEM8,TEMP2
 9034  044534  013737  045706  007434          MOV     RSEL4,TEMP3
 9035  044542  013737  045710  007436          MOV     RSEL6,TEMP4     ;ELSE
 9036  044550  004737  026434                  JSR     PC,LGDVE        ;LOG ERROR
 9037  044554                                  ERRSOFT 15,DVEM8,ERR13
  (4)  044554  104457                                                  TRAP    C$ERSOFT
  (5)  044556  000017                                                  .WORD   15
  (5)  044560  025036                                                  .WORD   DVEM8
  (5)  044562  026110                                                  .WORD   ERR13
 9038
 9039  044564  000411                          BR      DVTRX1          ;AND EXIT
 9040  044566  013737  045706  007362  DVTR5A: MOV     RSEL4,DVRXA
 9041  044574  013737  045710  007364          MOV     RSEL6,DVRCC
 9042  044602  052737  000004  007502          BIS     #QRX,FLAG
 9043  044610  042737  000040  007502  DVTRX1: BIC     #CRX,FLAG       ;CLEAR FLAG FOR RX DONE
 9044  044616  000207                  DVTREX: RTS     PC              ;AND EXIT
 9045
```

```
                                ; DEVICE DEPENDENT SUBROUTINES
                                
                                .SBTTL                  DEVICE INTERRUPT SERVICE ROUTINES
                                
044620                                  BGNSRV  DVINS
044620                                                                          DVINS::
044620  052737  000001  007502          BIS     #ININT,FLAG
044626                                  ENDSRV
044626                                                                          L10021:
044626  000002                                                                          RTI
                                
044630                                  BGNSRV  DVOUTS
044630                                                                          DVOUTS::
044630  052737  000002  007502          BIS     #OTINT,FLAG
044636                                  ENDSRV
044636                                                                          L10022:
044636  000002                                                                          RTI
                                
                                
                                ;++
                                ; 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
                                ;--
                                
044640  011637  007462          CLRAW:  MOV     (SP),PCADD      ;SAVE PC OF CALLING ROUTINE
044644  042777  000040  145506          BIC     #RQI,@SEL0
044652  012737  000050  007542  CLRA3:  MOV     #50,TIMER1      ;SET UP TIMER FOR 50(OCTAL) TICKS
044660  005737  007542          CLRA1:  TST     TIMER1
044664  001406                          BEQ     CLRA2           ;IF TIMER EXPIRED ERROR
044666                                  BREAK
044666  104422                                                                  TRAP    C$BRK
044670  032777  000200  145462          BIT     #RDI,@SEL0      ;IS RDI CLEAR
044676  001370                          BNE     CLRA1           ;IF NOT GO CHECK TIMER
                                                                ; ELSE
044700  000207                          RTS     PC              ;RETURN TO CALLER
044702  012737  024252  007432  CLRA2:  MOV     #DVEM0,TEMP2
044710  017737  145444  007434          MOV     @SEL0,TEMP3
044716  017737  145442  007436          MOV     @SEL2,TEMP4
044724  004737  026434                  JSR     PC,LGDVE        ;LOG DEVEICE EVENT 0
044730  005237  007400                  INC     ERRCNT
```

```
 9124  044734                 ERRSOFT 16,DVEM0,ERR9   ;WHILE WAITING FOR RDI
  (4)  044734  104457                                                    TRAP    C$ERSOFT
  (5)  044736  000020                                                    .WORD   16
  (5)  044740  024252                                                    .WORD   DVEM0
  (5)  044742  026032                                                    .WORD   ERR9
 9125  044744  000742                 BR      CLRA3           ;RESET TIMER AND CONTINUE
```

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

                                ;++
                                ; FUN TIONAL 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
                                ;--

044746  011637  007462        TOORIO: MOV     (SP),PCADD      ;SAVE ADDR. OF CALLING ROUTINE
044752  012737  000100  007542        MOV     #100,TIMER1     ;SET UP TIMER
044760  005737  007542        TOOR3:  TST     TIMER1          ;IS TIME EXPIRED
044764  001022                        BNE     TOOR1           ;IF NOT CONTINUE
                                                                ;IF YES ERROR
044766  012737  024340  007432        MOV     #DVEM1,TEMP2
044774  017737  145364  00743(        MOV     @SEL2,TEMP4
045002  017737  145352  007434        MOV     @SEL0,TEMP3
045010  004737  026434                JSR     PC,LGDVE
045014  005237  007400                INC     ERRCNT
045020                                ERRSOFT 17,DVEM1,ERR9
 (4)    045020  104457                                                  TRAP    C$ERSOFT
 (5)    045022  000021                                                  .WORD   17
 (5)    045024  024340                                                  .WORD   DVEM1
 (5)    045026  026032                                                  .WORD   ERR9
045030  000746                        BR      TOORIO

045032                        TOOR1:  BREAK
 (3)    045032  104422                                                  TRAP    C$BRK
045034  032737  000002  007502        BIT     #OTINT,FLAG     ;IS THERE AN OUTPUT
                                                                ;PENDING
045042  001402                        BEQ     TOOR2           ;IF NOT GO TO 2
                                                                ;ELSE GO HANDL IT
045044  004737  045070                JSR     PC,OUTHDL       ;
045050  032737  000001  007502 TOOR2:  BIT     #ININT,FLAG     ;IS THERE AN INPUT PENDING
045056  001740                        BEQ     TOOR3           ;IF NOT GO BACK TO TIMER CK.
045060  042737  000001  007502        BIC     #ININT,FLAG     ;ELSE CLEAR THE INPUT PEND FLAG
045066  000207                        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
                        ;
                        ;--
045070  011637  007462          OUTHDL: MOV     (SP),PCADD      ;SAVE ADDR. OF CALLING ROUTINE
045074  042737  000002  007502          BIC     #OTINT,FLAG
045102  032777  000200  145254          BIT     #RDO,@SEL2      ;CLEAR PEND FLAG AND CHK FOR RDO
045110  001023                          BNE     OUTH1           ;IF RDO OK ...ELSE LOG ERROR
045112  012737  024674  007432          MOV     #DVEM6,TEMP2
045120  017737  145240  007434          MOV     @SEL2,TEMP3
045126  017737  145242  007436          MOV     @SEL6,TEMP4
045134  004737  026434                  JSR     PC,LGDVE        ;GO LOG ERROR
045140  005237  007400                  INC     ERRCNT
045144                                  ERRSOFT 18,DVEM6,ERR9
  (4)   045144  104457                                                  TRAP    C$ERSOFT
  (5)   045146  000022                                                  .WORD   18
  (5)   045150  024674                                                  .WORD   DVEM6
  (5)   045152  026032                                                  .WORD   ERR9

                                        ;EXIT TEST IF ERROR

045154                                  ESCAPE  TST
  (3)   045154  104410                                                  TRAP    C$ESCAPE
  (3)   045156  000676                                                  .WORD   L10020-.

045160  032777  000001  145176  OUTH1:  BIT     #BACC,@SEL2     ;IS THE OUTPUT BACC
045166  00 002                          BNE     1$              ; BR IF NO
045170  000137  045614                  JMP     OUTH2           ;IF SO GO TO 2
```

```
                                                        ;ELSE LOG ERROR AND PRINT IT
045174  017737  145174  007436  1$:     MOV     @SEL6,TEMP4
                                        ; IF NO BUFFER OUTPUT JUST COUNT THEM

045202  032737  000004  007436          BIT     #BIT2,TEMP4
045210  001404                          BEQ     OUTH6           ;IF NO BUFF INC COUNT AND EXIT
                                                                ;ELSE GO TO 6
045212  005237  007374                  INC     NOBUF
045216  000137  045672                  JMP     OUTHEX

045222  023727  012406  000006  OUTH6:  CMP     OPTYP,#6        ;DMR ?
045230  002426                          BLT     51$             ;IF NOT DMR MODE SKIP TO 51
045232  032737  002000  007502          BIT     #BTUP,FLAG      ;HERE BECAUSE OF BASE TABLE UPDATE REQ ? REV B BY EC
045240  001402                          BEQ     50$             ;NO,BRANCH  REV B BY EC
045242  000137  045672                  JMP     OUTHEX          ;EXIT
045246  032737  000040  007436  50$:    BIT     #BIT5,TEMP4     ;IS IT RUN STATE
045254  001414                          BEQ     51$             ;IF NOT BRANCH
045256  032737  001000  007502          BIT     #DMRRUN,FLAG    ;IS RUN EXPECTED
045264  001405                          BEQ     52$             ;IF NOT BRANCH
045266  042737  001000  007502          BIC     #DMRRUN,FLAG    ;IF SO THEN CLEAR EXPECTED
045274  000137  045672                  JMP     OUTHEX          ;AND EXIT
045300  012737  025437  007442  52$:    MOV     #RUNSBM,CONOTM
045306  012737  024751  007432  51$:    MOV     #DVEM7,TEMP2
045314  017737  145044  007434          MOV     @SEL2,TEMP3

045322  004737  026434                  JSR     PC,LGDVE
045326  012737  014635  007442          MOV     #LP0,CONOTM     ;LOAD 'NULL STRING'' TO INIT CONOTM
045334  032737  000001  007436          BIT     #BIT0,TEMP4     ;IS THIS DATA CHECK
045342  001403                          BEQ     1$
045344  012737  025424  007442          MOV     #DATCKM,CONOTM
045352  032737  000002  007436  1$:     BIT     #BIT1,TEMP4     ;IS THIS TIMEOUT
045360  001403                          BEQ     2$
045362  012737  025413  007442          MOV     #TIMOM,CONOTM
045370  032737  000010  007436  2$:     BIT     #BIT3,TEMP4     ;IS THIS DDCMP MAINT RECVD
045376  001403                          BEQ     4$
045400  012737  025373  007442          MOV     #DDCMRM,CONOTM
045406  032737  000020  007436  4$:     BIT     #BIT4,TEMP4     ;IS THIS LOST DATA
045414  001403                          BEQ     5$
045416  012737  025361  007442          MOV     #LOSDAM,CONOTM
045424  032737  000100  007436  5$:     BIT     #BIT6,TEMP4     ;IS THIS DISCONNECT
045432  001403                          BEQ     6$
045434  012737  025346  007442          MOV     #DISCOM,CONOTM
045442  032737  000200  007436  6$:     BIT     #BIT7,TEMP4     ;IS THIS DDCMP START RECVD
045450  001403                          BEQ     7$
045452  012737  025326  007442          MOV     #DDCSRM,CONOTM
045460  032737  000400  007436  7$:     BIT     #BIT8,TEMP4     ;IS THIS NON-EXSISTENT MEMORY
045466  001403                          BEQ     8$
045470  012737  025310  007442          MOV     #NXMM,CONOTM
045476  032737  001000  007436  8$:     BIT     #BIT9,TEMP4     ;IS THIS PROCEDURE ERROR
045504  001403                          BEQ     9$
045506  012737  025270  007442          MOV     #PROEM,CONOTM
045514  023727  012406  000006  9$:     CMP     OPTYP,#6        ;IS THIS DMR MODE
045522  002416                          BLT     11$             ;IF NOT BRANCH
045524  032737  002000  007436          BIT     #BIT10,TEMP4    ;IS THIS A RX IDLE
045532  001403                          BEQ     10$             ;IF NOT BRANCH
045534  012737  025461  007442          MOV     #RXIDM,CONOTM   ;IF SO SET UP MESSAGE
```

```
 9284  045542  032737  004000  007436  10$:    BIT     #BIT11,TEMP4    ;IS THIS CTS FAILED
 9285  045550  001403                          BEQ     11$             ;IF NOT BRANCH
 9286  045552  012737  025505  007442          MOV     #CTSFM,CONOTM   ;IF SO SET UP MESSAGE
 9287  045560  032737  010000  007436  11$:    BIT     #BIT12,TEMP4    ;IS THIS CD GLITCHED
 9288  045566  001403                          BEQ     12$             ;BR IF NO
 9289  045570  012737  025471  007442          MOV     #CDGLM,CONOTM   ;IF SO SET UP MESSAGE
 9290
 9291  045576  005237  007400          12$:    INC     ERRCNT
 9292  045602                                  ERRSOFT 19,DVEM7,ERR8
   (4) 045602  104457                                                                  TRAP    C$ERSOFT
   (5) 045604  000023                                                                  .WORD   19
   (5) 045606  024751                                                                  .WORD   DVEM7
   (5) 045610  025750                                                                  .WORD   ERR8
 9293  045612  000427                          BR      OUTHEX          ;CLEAR RDO AND RETURN TO CALLER
 9294
 9295  045614                          OUTH2:
 9296  045614  032777  000004  144542          BIT     #RXBIT,@SEL2    ;IS THIS RX BACC OUT
 9297  045622  001012                          BNE     OUTH3           ;IF NOT THEN IT MUST BE TX.
 9298  045624  052737  000020  007502          BIS     #CTX,FLAG
 9299  045632  017737  144532  045702          MOV     @SEL4,TSEL4
 9300  045640  017737  144530  045704          MOV     @SEL6,TSEL6
 9301  045646  000411                          BR      OUTHEX
 9302
 9303  045650  052737  000040  007502  OUTH3:  BIS     #CRX,FLAG       ;SET RX COMPL
 9304  045656  017737  144506  045706  OUTH4:  MOV     @SEL4,RSEL4     ;THEN MOVE TO TEMP
 9305  045664  017737  144504  045710          MOV     @SEL6,RSEL6     ;AND SEL6 TO TEMP
 9306  045672  042777  000200  144464  OUTHEX: BIC     #RDO,@SEL2      ;CLEAR RDO
 9307  045700  000207                          RTS     PC              ;RETURN TO CALLER
 9308  045702  000000                  TSEL4:  .WORD   0
 9309  045704  000000                  TSEL6:  .WORD   0
 9310  045706  000000                  RSEL4:  .WORD   0
 9311  045710  000000                  RSEL6:  .WORD   0
 9312
 9313  045712  000207                          RTS     PC
 9314
 9315
```

```
9328
9329                                    .EVEN
9330
9331
9332
9338                            .SBTTL  REQUEST BASE TABLE UPDATE
9339                            ;       THIS ROUTINE ADDED FOR REV B B; EC
9340                            ;++
9341                            ;FUNCTIONAL DESCRIPTION:
9342                            ;       DVBTUP - THIS ROUTINE IS CALLED AT END OF PASS TO UPDATE THE DMR
9343                            ;                BASE TABLE IN LOCAL MEMORY BY ISSUING AN UPDATE BASE TABLE
9344                            ;                REQUEST TO THE DMR.
9345                            ;
9346                            ;USE OF FLAGS:
9347                            ;       FLAG - BITS IN THIS WORD ARE SET BY THE DEVICE INTERRUPT ROUTINE.
9348                            ;              WHEN SET TO A 1, THE FOLLOWING BITS MEAN
9349                            ;                ININT = BIT 1 = DEVICE INPUT READY
9350                            ;                OTINT = BIT 2 = DEVICE OUT READY
9351                            ;
9352                            ;INPUTS: NONE REQUIRED.
9353                            ;
9354                            ;SUBORDINATE ROUTINES USED:
9355                            ;
9356                            ;       CLRAW - CLEAR RQI AND WAIT FOR DEVICE TO CLEAR RDI.
9357                            ;       OUTHDL - THE DEVICE OUTPUT SERVICE ROUTINE.
9358                            ;       LGDVE - LOG A DEVICE ERROR TO EVENT LOG.
9359                            ;
9360                            ;CALLING SEQUENCE:
9361                            ;       JSR     PC,DVBTUP
9362                            ;--
9363                                    ;DEVICE BASE TABLE UPDATE ROUTINE
9364  045714                    DVBTUP:
9365  045714  023727  012406  000006        CMP     OPTYP,#6        ;DMR IN DMR MODE ?
9366  045722  002453                        BLT     40$             ;NO,BRANCH
9367  045724  112777  000151  144426        MOVB    #151,@BSEL0     ;REQUEST BASE TABLE UPDATE
9368  045732  052777  000100  144424        BIS     #IEO,@SEL2      ;ENABLE INTERRUPT ON OUTPUT READY
9369  045740  004737  044746                JSR     PC,TOORIO       ;WAIT FOR RDI
9370  045744  004737  044640                JSR     PC,CLRAW        ;CLEAR RQI AND WAIT FOR RDI TO CLEAR
9371  045750  012737  000200  007542        MOV     #200,TIMER1     ;WAIT FOR INTERRUPT
9372  045756                    10$:        BREAK
  (3) 045756  104422                                                        TRAP    C$BRK
9373  045760  032737  000002  007502        BIT     #OTINT,FLAG     ;OUTPUT READY ?
9374  045766  001021                        BNE     30$             ;YES,BRANCH
9375  045770  005737  007542                TST     TIMER1          ;TIME OUT ?
9376  045774  001370                        BNE     10$             ;NO,BRANCH
9377  045776  012737  024424  007432        MOV     #DVEM3,TEMP2    ;DEVICE ERROR
9378  046004  017737  144350  007434        MOV     @SEL0,TEMP3     ;SAVE REGISTER
9379  046012  017737  144346  007436        MOV     @SEL2,TEMP4     ;SAVE REGISTER
9380  046020  004737  026434                JSR     PC,LGDVE        ;GO LOG DEVICE ERROR
9381  046024  005237  007400                INC     ERRCNT          ;BUMP ERROR COUNT
9382  046030  000410                        BR      40$             ;RETURN
9383  046032  052737  002000  007502  30$:   BIS     #BTUP,FLAG      ;SET BASE TABLE UPDATE BIT
9384  046040  004737  045070                JSR     PC,OUTHDL       ;OUTPUT SERVICE ROUTINE
9385  046044  042737  002000  007502        BIC     #BTUP,FLAG      ;CLEAR BASE TABLE UPDATE BIT
9386  046052  000207                40$:    RTS     PC              ;RETURN
9387
```

```
 9388  046054                                  ENDTST
  (3)  046054                                                          L10020:
  (3)  046054  104401                                                          TRAP    C$ETST
 9389
 9390
 9391
```

```
 9393                                   .SBTTL  HARDWARE PARAMETER CODING SECTION
 9394
 9395
 9396                                   ;++
 9397                                   ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
 9398                                   ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
 9399                                   ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
 9400                                   ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
 9401                                   ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
 9402                                   ; WITH THE OPERATOR.
 9403                                   ;--
 9404
 9405  046056                                   BGNHRD
  (3)  046056  000025                                                          .WORD L10023-L$HARD/2
  (3)  046060                                                                L$HARD::
 9406
 9416
 9417                                   .SBTTL          DEVICE INDEPENDENT SECTION
 9418
 9419  046060                                   GPRML   DPLX,0,1,YES
  (4)  046060  000130                                                          .WORD   T$CODE
  (4)  046062  046132                                                          .WORD   DPLX
  (4)  046064  000001                                                          .WORD   1
 9420
 9421
 9432
 9433
 9434                                   .SBTTL          DEVICE DEPENDENT SECTION
 9435
 9436  046066                                   GPRMA   CSRADR,2,0,160000,177776,YES
  (4)  046066  001031                                                          .WORD   T$CODE
  (4)  046070  046163                                                          .WORD   CSRADR
  (4)  046072  160000                                                          .WORD   T$LOLIM
  (4)  046074  177776                                                          .WORD   T$HILIM
 9437  046076                                   GPRMA   VECTOR,4,0,300,776,YES
  (4)  046076  002031                                                          .WORD   T$CODE
  (4)  046100  046211                                                          .WORD   VECTOR
  (4)  046102  000300                                                          .WORD   T$LOLIM
  (4)  046104  000776                                                          .WORD   T$HILIM
 9438  046106                                   GPRMD   PRIOR,6,0,340,4,7,YES
  (4)  046106  003032                                                          .WORD   T$CODE
  (4)  046110  046244                                                          .WORD   PRIOR
  (4)  046112  000340                                                          .WORD   340
  (4)  046114  000004                                                          .WORD   T$LOLIM
  (4)  046116  000007                                                          .WORD   T$HILIM
 9439                                   ;       GPRMD   DEVPRM,10,D,17,0,15.,YES
 9440
 9441  046120                                   GPRMD   OPTN,12,0,7,0,7,YES
  (4)  046120  005032                                                          .WORD   T$CODE
  (4)  046122  046272                                                          .WORD   OPTN
  (4)  046124  000007                                                          .WORD   7
  (4)  046126  000000                                                          .WORD   T$LOLIM
  (4)  046130  000007                                                          .WORD   T$HILIM
 9442                                   ;       GPRMD   BAUD,14,0,7,0,7,YES
 9443                                   ;       GPRMD   LININ,16,0,7,0,7,YES
 9444
```

```
 9445  046132                                  ENDHRD
  (2)                                                                    .EVEN
  (3)  046132                                                        L10023:
 9446
 9447                                  .NLIST  BEX
 9448
 9449                                  ;DEVICE INDEPENDENT QUESTIONS
 9450
 9451  046132  052506  046114  042040  DPLX:   .ASCIZ  /FULL DUPLEX OPERATION : /
 9452
 9460
 9461                                  ;DEVICE DEPENDENT QUESTION
 9462
 9463  046163     104  053105  041511  CSRADR: .ASCIZ  /DEVICE CSR ADDRESS : /
 9464  046211     111  052116  051105  VECTOR: .ASCIZ  /INTERRUPT VECTOR ADDRESS: /
 9465  046244  047111  042524  051122  PRIOR:  .ASCIZ  /INTERRUPT PRIORITY : /
 9466  046272  042504  044526  042503  OPTN:   .ASCIZ  /DEVICE OPTION TYPE : (0-DMC,5=DMR-DMC MODE ,7-DMR)/
 9467
 9468
 9469                                  .LIST   BEX
 9470          046356                          .EVEN
 9471
 9478
```

```
 9481                                   ;.SBTTL SOFTWARE PARAMETER CODING SECTION
 9482
 9483                                   ;++
 9484                                   ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
 9485                                   ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
 9486                                   ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
 9487                                   ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
 9488                                   ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
 9489                                   ; WITH THE OPERATOR.
 9490                                   ;--
 9491
 9492                                   ;       BGNSFT
 9493
 9502
 9503
 9504                                   ;       ENDSFT
 9505
 9506
 9513
 9514                                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 9515                                   ; TEMPORARY PATCH AREA - FOR DEBUG PURPOSES
 9516                                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 9517
 9518  046356                           $PATCH:
 9519  046356  000030                           .BLKW   30
 9520
 9527
 9528  046436                                   LASTAD
   (2)                                                                  .EVEN
   (4)  046436  000000                                                  .WORD   0
   (4)  046440  000000                                                  .WORD   0
   (3)  046442                        L$LAST::
 9529  046442                                   ENDMOD
 9530
 9531         000001                    .END
```

CZCLKB.P11 16-JUN-81 14:38 CROSS REFERENCE TABLE -- USER SYMBOLS

ACT =	000003	4495#	7543	7925	8001	8241		
ACTATV	037710	7711	7925#					
ACTBCR	037514	7729	7890#					
ACTCHK	040124	7691	7967#					
ACTCLB	037036	7790	7804#					
ACTCLP	040236	7725	7995#					
ACTCLR	036472	7689	7743#					
ACTCOP	037334	7699	7858#					
ACTCRC	040140	7720	7973#					
ACTCSE	036626	7694	7766#					
ACTCST	036754	7695	7792#					
ACTDLL	037756	7715	7939#					
ACTDME	037262	7731	7838	7841#				
ACTDMQ	037254	7732	7840#					
ACTDMS	037232	7730	7835#					
ACTDMX	037270	7842#						
ACTECH	040034	7719	7956#					
ACTEQO	037456	7703	7879#					
ACTEXT	036556	7735	7755#					
ACTHLP	036512	7693	7749#					
ACTLIS	037746	7714	7936#					
ACTLLP	040246	7726	7997#					
ACTLPX	040264	7992	7994	7996	7998	8001#		
ACTLXX	040326	7965	7986	7989	8002	8006#		
ACTMEX	037702	7872	7888	7905	7910	7916	7919	7921#
ACTME1	037636	7894	7896	7898	7900	7902	7909#	
ACTMOP	040216	7723	7991#					
ACTMOS	040146	7734	7976#					
ACTMS0	037536	7704	7893#					
ACTMS1	037544	7705	7895#					
ACTMS2	037554	7706	7897#					
ACTMS3	037564	7707	7899#					
ACTMS4	037574	7708	7901#					
ACTMS5	037604	7709	7903#					
ACTMS6	037622	7710	7906#					
ACTM2X	040004	7926	7934	7937	7940	7943	7947#	
ACTNO	040024	7718	7953#					
ACTNUF	036462	7728	7740#					
ACTNUL	036470	7688	7741#					
ACTNUM	037344	7700	7861#					
ACTOPM	037436	7701	7874#					
ACTPAS	037720	7712	7928#					
ACTPRO	040154	7721	7979#					
ACTPRT	036566	7733	7757#					
ACTQFG	040160	7968	7971	7974	7977	7981#		
ACTREC	037740	7713	7933#					
ACTREX	027576	6284	6304#					
ACTRHL	027532	6283	6296#					
ACTRLG	027606	6285	6308#					
ACTRLP	040256	7727	7999#					
ACTRNF	027522	6289	6292#					
ACTRNL	027530	6282	6293#					
ACTRPS	040206	7722	7988#					
ACTRUN	036602	7692	7761#					
ACTSEX	037646	7736	7913#					
ACTSHO	036502	7690	7746#					

```
ACTSHW  037076            7777    7796    7815#   7827
ACTSTE  037276            7696    7848#
ACTSTS  040132            7702    7970#
ACTSTT  037306            7697    7851#
ACTSTX  037314            7849    7852#
ACTSWE  027622            6286    6313#
ACTSWF  027764            6287    6334#
ACTSWO  030076            6288    6351#
ACTSZE  037324            7698    7855#
ACTTAL  037776            7717    7945#
ACTTLP  040226            7724    7993#
ACTTRA  037766            7716    7942#
ADDCC   031712            6604#   7627    7661
ADDC1   032006            6606    6616#
ADR   = 000020 G          4478#
ALCK    040720            5015    8150#
ALCK1   041032            8206    8215#   8259
ALCK2   041076            8216    8225#   8291
ALCK2A  041316            8256    8261#
ALCK3   041342            8228    8263    8266#
ALCK3A  041434            8273    8279#
ALCK3B  041446            8280    8282#
ALCK3C  041442            8277    8281#   8288
ALCK4   041470            8267    8286#
ALCK4A  041476            8285    8289#
ALCK4B  041510            8290    8292#
ALCK5   040762            8205#   8281
ALLTR   040762            8079    8103    8128    8155    8204#
ASSEMB= 000010            4228
ATVMOD= 000027            4596#   5178
BACC  = 000001            4668#   9225
BAD     007447            4980#   5850    8364*
BADCHR= 000051            4614#   5292
BASE    017370            5625#   6318    6320    6326    6328    6338    6340    6344    6346    6369    6371    6377    6379
                          8739
BASM1   016242            5580#   6564
BASM1A  016207            5576#
BASM2   016233            5579#   6570
BASM3   016224            5578#   6568
BDATA   007456            4988#   6320*   6321*   6328*   6329*   6340*   6346*   6371*   6372*   6379*   6380*   6407
BEND    007454            4983#   6318*   6319*   6326*   6327*   6338*   6339*   6344*   6345*   6354*   6356*   6357    6369*
                          6370*   6377*   6378*   6415
BIT0  = 000001 G          4478#   4515    4649    4668    9255
BIT00 = 000001 G          4478#
BIT01 = 000002 G          4478#
BIT02 = 000004 G          4478#
BIT03 = 000010 G          4478#
BIT04 = 000020 G          4478#
BIT05 = 000040 G          4478#
BIT06 = 000100 G          4478#
BIT07 = 000200 G          4478#
BIT08 = 000400 G          4478#
BIT09 = 001000 G          4478#
BIT1  = 000002 G          4478#   4516    9258
BIT10 = 002000 G          4478#   4672    9281
BIT11 = 004000 G          4478#   4670    9284
```

CZCLKB.P11 16-JUN-81 14:38 CROSS REFERENCE TABLE -- USER SYMBOLS

```
BIT12 = 010000 G        4478#   9287
BIT13 = 020000 G        4478#
BIT14 = 040000 G        4478#   4652    4669
BIT15 = 100000 G        4478#
BIT2  = 000004 G        4478#   4517    4647    4673    8758    8761    9232
BIT3  = 000010 G        4478#   4518    4648    8758    8764    9261
BIT4  = 000020 G        4478#   4519    9264
BIT5  = 000040 G        4478#   4520    4650    4665    9243
BIT6  = 000100 G        4478#   4674    9267
BIT7  = 000200 G        4478#   4651    4666    4667    9270
BIT8  = 000400 G        4478#   4671    9273
BIT9  = 001000 G        4478#   4653    9276
BLDBEX  032120          6661    6665#
BLDBUF  032010          6643#   7534    7541    7628    7662    7811    8052
BLDB1   032020          6648#
BLDB2   032062          6656#   6663
BLDB3   032100          6659#   6664
BOE   = 000400 G        4478#
BSEL0   012360          5355#   8737*   8756*   8770*   8776*   8835*   8889*   8973*   9367*
BSEL1   012362          5356#   7277*   7278*   8712*   8715*
BSEL2   012364          5358#
BSEL3   012366          5359#   7281*   7282*
BSEL4   012370          5361#
BSEL5   012372          5362#   7285*   7286*
BSEL6   012374          5364#   8761*   8764*
BSEL7   012376          5365#   7289*   7290*
BTHEAD  030476          6405    6422#
BTUP  = 002000          4556#   9240    9383    9385
BUFEX   015223          5540#   7611    7640
BUFLIM= 001000          4483#   4929    4930    4931    6605    6612    6733    7609    7638    7804    8048
BYTBIT  007406          4960#   6565    7836*   7840*
CABLE = 000002          4502#
CBLLOP= 000045          4610#   5321
CCURAD  007344          4940#   6745*   6753    6757*   7530*   7538    7660    7665*   7789*
CDGLM   025471          5790#   9289
CHECK = 000003          4576#   5226
CLEAR = 000001          4574#   5164    7602    7743
CLIACT  036302          7570    7680#
CLIALN= 000007          4567#   5290
CLIALP= 000006          4566#   5212    5222    5295
CLIBCR  012734          5458#   7890
CLIBDL  012627          5456#   8005
CLIBIF= 000C03          4563#
CLIBR = 000002          4562#   5161    5163    5165    5168    5170    5179    5181    5183    5185    5187    5190    5192
                        5207    5209    5219    5227    5239    5241    5257    5259    5262    5269    5271    5273    5275
                        5277    5279    5281    5283    5288    5291    5298    5300    5305    5320    5322    5324    5326
                        5331    6263
CLIBRX  012607          5455#   6983
CLIDEC= 000011          4569#   5304    5330
CLIERM  012512          5452#   6236    7575
CLIERR= 000000          4560#   5171    5292    6273
CLIEXI= 000001          4561#   5153    5155    5157    5159    5198    5200    5336    6255    6257    6259    6261    6266
                        6268    6272    6274
CLINBG  012565          5454#   7021
CLINPS  012671          5457#   7961
CLINUF  012542          5453#   6241    7579
```

CZCLKB.P11 16-JUN-81 14:38 CROSS REFERENCE TABLE -- USER SYMBOLS

```
CLINUM= 000005          4565#
CLIOCT= 000010          4568#   5214    5216    6271
CLIPW   013032          5460#   8018
CLIRAC  027464          6231    6277#
CLIRT   027266          6230    6253#
CLISEO  013001          5459#   7865
CLISPA= 000004          4564#   5151    5175    5196    5205    5213    5289    6253
CLISTR= 000012          4570#   5154    5156    5158    5160    5162    5164    5167    5169    5176    5178    5180    5182
                        5184    5186    5189    5191    5197    5199    5206    5208    5224    5226    5238    5240    5256
                        5258    5261    5268    5270    5272    5274    5276    5278    5280    5282    5297    5299    5319
                        5321    5323    5325    6256    6258    6260    6262    6265    6267    6269
CLITRE  010660          5148#   7569
CLI$PM  012476          5450#   7567
CLI$RP  012504          5451#   6228
CLKBR   007524          5023#   7235*
CLKCSR  007522          5022#   6050*   6071*   7216    7226*   7227*   7228*   7236*   7371*   7499*
CLKEN   007532          5026#   6071    7220*   7229*   7236    7499
CLKHZ   007530          5025#   6053    6068    6187    7237    7245
CLKINT  026230 G        6048#   7299
CLKSET  026204          6003#   7219    7225
CLKVEC  007526          5024#   7214*   7234*   7299
CLRAW   044640          8750    8765    8772    8789    8839    8894    8978    9109#   9370
CLRA1   044660          9112#   9116
CLRA2   044702          9113    9119#
CLRA3   044652          9111#   9125
CMDBUF  003060          4795#   6228    6229    7567    7568
CMPBUF  005562          4931#   6731*   6745    7530    7788
CMPPTR  007336          4937#   6739*   6740*   6753*   6754*   6755*   6759*   7524*   7525*   7537    7649*   7650*   7658*
                        7659    7663*   7770*   7771*   7772    7785*   7786*   7787    8027*   8028*   8029    8332
CMPSEX  041764          8330    8399#
CMPSR   041510          8278    8329#
CMPS1   041650          8357#   8371
CMPS2   041654          8360#
CMPS3   041542          8335#   8379
CMPS5   041752          8376#
CMPS5A  041756          8373    8377#
CMPS6   041720          8358    8362    8369#
CMPS7   041636          8350    8355#
CMPTOT  007340          4938#   6748*   6751    6760*   7542*   7644*   7651    7664*   7774    7778*   8016    8036
CMSG0 = 000020          4589#   5270
CMSG1 = 000021          4590#   5268
CMSG2 = 000022          4591#   5272
CMSG3 = 000023          4592#   5274
CMSG4 = 000024          4593#   5278
CMSG5 = 000025          4594#   5276
CMSG6 = 000026          4595#   5280
CONOTM  007442          4977#   5871    5885    8492*   8509*   8517*   8522*   9249*   9254*   9257*   9260*   9263*   9266*
                        9269*   9272*   9275*   9278*   9283*   9286*   9289*
CONTIN  007444          4978#   8788*   8838
CPTR    007416          4967#   6646    6650*   7533*   7537*   7625*   7629    7659*   7663    7787*   7800*   8050*   8078*
                        8099*   8125*   8151*   8217    8222*   8234    8282*   8289    8331*   8336    8340*   8621*
CPTRR   007414          4966#   8075*   8102*   8126*   8153*   8207    8212*   8240*   8245    8253*   8265*   8286    8332*
                        8342    8345*   8579*
CR      020000          5633#   6180
CRC   = 000040          4605#
CRCB  = 000020          4519#   7973
```

CZCLKB.P11 16-JUN-81 14:38 CROSS REFERENCE TABLE -- USER SYMBOLS

```
CRX    = 000040          4551#  8981   9012   9029   9043   9303
CSHEXP= 000006           4579#  5197
CSHTRN= 000007           4580#  5199
CSRADR  046163           9436   9463#
CTOTCC  007342           4939#  6747*  6755   6757   6758*  6763*  7517*  7637   7642   7666*  7779*
CTS    = 000004          4647#  5048
CTSFM   025505           5791#  9286
CTX    = 000020          4550#  8981   9012   9014   9028   9298
CURADD  007420           4968#  6648   6653   6655   6665*  7532*  7538*  7626*  7631   7660*  7665   7805*  8049*
                         8448*  8454*  8480   8481
CURCC   007412           4965#  6604   6611   6614*  6615   6649   6654   6665   7528*  7540*  7807*  7868*  7881*
                         7904*  7907*  7909*  8048*  8449*  8455*  8482   8483
C$AU   = 000052          4228#  7463
C$AUTO= 000061           4228#  7351
C$BRK  = 000022          4228#  8719   8995   9114   9166   9372
C$BSEG= 000004           4228#
C$BSUB= 000002           4228#
C$CEFG= 000045           4228#
C$CLCK= 000062           4228#  7217   7223
C$CLEA= 000012           4228#  7390
C$CLOS= 000035           4228#
C$CLP1= 000006           4228#
C$CVEC= 000036           . 28#
C$DCLN= 000044           4228#  7199
C$DODU= 000051           4228#
C$DRPT= 000024           4228#
C$DU   = 000053          4228#  7426
C$EDIT= 000003           4228#  4306
C$ERDF= 000055           4228#
C$ERHR= 000056           4228#  8465
C$ERRO= 000060           4228#
C$ERSF= 000054           4228#
C$ERSO= 000057           4228#  8352   8366   8375   8728   8991   9005   9022   9037   9124   9163   9219   9292
C$ESCA= 000010           4228#  9223
C$ESEG= 000005           4228#
C$ESUB= 000003           4228#
C$ETST= 000001           4228#  9388
C$EXIT= 000032           4228#  7241   7315   7374   7559   7601
C$GETB= 000026           4228#
C$GETW= 000027           4228#
C$GMAN= 000043           4228#  6228   7237   7567   8441   8565
C$GPHR= 000042           4228#  7258
C$GPLO= 000C30           4228#
C$GPRI= 000040           4228#
C$INIT= 000011           4228#  7331
C$INLP= 000020           4228#
C$MANI= 000050           4228#  7555
C$MEM  = 000031          4228#
C$MSG  = 000023          4228#  5851   5855   5859   5873   5878   5882   5886
C$OPEN= 000034           4228#
C$PNTB= 000014           4228#  5850   5854   5858   5871   5872   5876   5877   5881   5885
C$PNTF= 000017           4228#  6180   6183   6222   6236   6241   6297   6359   6405   6414   6564   6568   6570
                         6610   6983   7021   7240   7508   7548   7575   7579   7611   7620   7640   7654   7750
                         7824   7865   7890   7961   8005   8018   8461   8534   8615   8633
C$PNTS= 000016           4228#  6434   6452   6457   6463   6470   6471   6487   6495   6502   6515   6530   6803
                         6841
```

CZCLKB.P11 16-JUN-81 14:38 CROSS REFERENCE TABLE -- USER SYMBOLS

```
C$PNTX= 000015          4228#
C$QIO = 000377          4228#
C$RDBU= 000007          4228#   7232
C$REFG= 000047          4228#   7202    7204    7206    7209
C$RESE= 000033          4228#
C$REVI= 000003          4228#   7306
C$RFLA= 000021          4228#
C$RPT = 000025          4228#   7131
C$SEFG= 000046          4228#
C$SPRI= 000041          4228#   7314    7372
C$SVEC= 000037          4228#   7299    7311    7312
C$TPRI= 000013          4228#
DAM     020233          5088    5654#
DATCKB= 000002          4516#   6811    7967    8014    8098    8237    8284    8329    8447    8560    8614
DATCKM  025424          5787#   9257
DCD   = 000001          4649#   5050
DCK   = 000014          4538#   6145
DC.LG   007464          4991#   7196    7198*   7600*
DDCMRM  025373          5785#   9263
DDCSRM  025326          5782#   9272
DDE   = 000022          4541#   6153
DER   = 000010          4536#   6131
DEV1    010650          5127#   6468*   6471    6474*   6476*   6792    7549*   7828*
DEV2    010652          5128#   6469*   6471    6475*   6477*   6795    7550*   7829*
DEV3    010654          5129#   6478*   6802    7551*   7830*
DEV4    010656          5130#   6479*   6807    6811    6815    6819    7552*   7831*
DFPTBL  002130 G        4351#
DIAGMC= 000000          4228
DISCOM  025346          5783#   9269
DLE   = 000020          4540#   6149
DLL     042050          5016    8441#
DLLAB   025214          5777#   8465    8495
DLLCM   015030          5537#   8461
DLLEA   042226          8464#   8497
DLLE1   042504          8488    8507#
DLLE2   042352          8486#   8506
DLLE3   042564          8516    8520#
DLLE4   042554          8517#
DLLE5   042666          8505    8535#
DLLE5A  042604          8521    8526#
DLLE5B  042640          8529    8531    8534#
DLLE6   042436          8490    8499#
DLLE7   042404          8493#   8510    8518    8523
DLLE8   042524          8508    8511#
DLLGA = 000400          4554#   8456    8487    8503
DLLIND  010542          5084#   8532
DLLMOD= 000033          4600#   5186
DLLM1   002647          4697    4754#   8448
DLLM1C  002172          4697#   8449
DLLM1E  002654          4697    4759#
DLLM2   002654          4698    4761#   8454
DLLM2C  002174          4698#   8455    8504
DLLM2E  003060          4698    4788#
DLLPRI  042202          8460#
DLLQ1   013122          5461#   8441
DLM     020225          5086    5652#
```

CZCLKB.P11 16-JUN-81 14:38 CROSS REFERENCE TABLE -- USER SYMBOLS

DLTXRX	042242	8450	8457	8471#	
DLVM	020251	5092	5658#		
DMC =	000000	4524#			
DMCEND	003462	4890#	6317	6337	6368
DMCIND	003432	4878#	6315	6336	6366
DMCM	020242	5090	5656#		
DMC002	023604	4880	5736#		
DMC003	023625	4881	5737#		
DMC004	023662	4882	5738#		
DMC005	023723	4883	5739#		
DMC006	023756	4884	5740#		
DMC007	024013	4885	5741#		
DMC010	024050	4886	5742#		
DMC011	024103	4887	5743#		
DMC012	024125	4888	5744#		
DMC013	024147	4889	5745#		
DMC377	024206	4890	5746#		
DMFMT	030547	6414	6423#		
DMPE =	000053	4616#	5216		
DMPM	020255	5093	5659#		
DMPQ =	000054	4617#	5218		
DMPS =	000052	4615#	5214	7593	7837
DMRC6 =	000004	4525#	8710		
DMRC7 =	000005	4526#			
DMREND	003430	4875#	6325	6343	6376
DMRIND	003250	4819#	6323	6342	6374
DMRRUN=	001000	4555#	8793	9245	9247
DMR000	020342	4819	5673#		
DMR001	020402	4820	5674#		
DMR002	020432	4821	5675#		
DMR003	020467	4822	5676#		
DMR004	020532	4823	5677#		
DMR005	020566	4824	5678#		
DMR006	020620	4825	5679#		
DMR007	020663	4826	5680#		
DMR010	020717	4827	5681#		
DMR011	020751	4828	5682#		
DMR012	021003	4829	5683#		
DMR013	021035	4830	5684#		
DMR014	021065	4831	5685#		
DMR015	021114	4832	5686#		
DMR016	021146	4833	5687#		
DMR017	021176	4834	5688#		
DMR020	021226	4835	5689#		
DMR021	021255	4836	5690#		
DMR022	021307	4837	5691#		
DMR023	021333	4838	5692#		
DMR024	021365	4839	5693#		
DMR025	021410	4840	5694#		
DMR026	021463	4841	5695#		
DMR027	021513	4842	5696#		
DMR030	021544	4843	5697#		
DMR031	021627	4844	5698#		
DMR032	021664	4845	5699#		
DMR033	021721	4846	5700#		
DMR034	021756	4847	5701#		

CZCLKB.P11 16-JUN-81 14:38 CROSS REFERENCE TABLE -- USER SYMBOLS

DMR035	022042	4848	5702#		
DMR036	022107	4849	5703#		
DMR037	022154	4850	5704#		
DMR040	022212	4851	5705#		
DMR041	022245	4852	5706#		
DMR042	022303	4853	5707#		
DMR043	022351	4854	5708#		
DMR044	022405	4855	5709#		
DMR045	022441	4856	5710#		
DMR046	022472	4857	5711#		
DMR047	022540	4858	5712#		
DMR050	022602	4859	5713#		
DMR051	022630	4860	5714#		
DMR052	022664	4861	5715#		
DMR053	022711	4862	5716#		
DMR054	022744	4863	5717#		
DMR055	022766	4864	5718#		
DMR056	023041	4865	5719#		
DMR057	023114	4866	5720#		
DMR060	023136	4867	5721#		
DMR061	023170	4868	5722#		
DMR062	023222	4869	5723#		
DMR063	023272	4870	5724#		
DMR064	023342	4871	5725#		
DMR065	023376	4872	5726#		
DMR066	023432	4873	5727#		
DMR067	023475	4874	5728#		
DMR177	023540	4875	5729#		
DMR6 =	000006	4527#	8713		
DMR7 =	000007	4528#			
DMSGAD	002176	4702#	6657		
DMSGCT	002150	4687#	6656		
DMUNKN	020322	4878	4879	5672#	
DMVM	020316	5101	5667#		
DNM	020246	5091	5657#		
DOW =	000004	4496#	7939	8779	
DPLX	046132	9419	9451#		
DPM	020217	5084	5650#		
DQM	020230	5087	5653#		
DSR =	000010	4648#	5049		
DTEM	020261	5094	5660#		
DUM	020222	5085	5651#		
DUMEX	031710	6572	6578#		
DUMPSR	031554	6562#	7595		
DUM1	031646	6566	6570#		
DUM2	031670	6569	6571#		
DUM3	031604	6565#	6576		
DUM4	031560	6563#	6575		
DUPM	020236	5089	5655#		
DVBTUP	045714	7370	8410	9364#	
DVEM0	024252	5758#	9119	9124	
DVEM1	024340	5760#	9158	9163	
DVEM3	024424	5762#	8723	8728	9377
DVEM4	024510	5764#	8986	8991	
DVEM5	024602	5766#	8999	9005	
DVEM6	024674	5768#	9214	9219	

```
DVEM7   024751            5770#   9250    9292
DVEM8   025036            5773#   9033    9037
DVEM9   025125            5775#   9018    9022
DVI   = 000012            4537#   6136
DVINEX  044002            8792    8794#
DVINIT  043320            8046    8702#   8730
DVINS   044620 G          7311    9071#
DVIN1   043462            8718    8732#
DVIN10  043656            8763    8765#
DVIN2   043400            8714    8717#   8721
DVIN3   043566            8748    8750#
DVIN4   043752            8782    8785    8788#
DVIN5   043736            8780    8784#
DVIN6   043362            8711    8713#
DVIN7   043534            8743    8745#
DVIN8   043700            8755    8776#
DVIN9   043640            8760    8762#
DVM     020265            5095    5661#
DVMODS  044004            6161    8835#
DVOUTS  044630 G          7312    9085#
DVRCC   007364            4950#   8211*   8230    8475*   8512    8618*   8629    8631*   8632*   8893    9041*
DVRCT   007366            4951#   8076*   8152*   8239*   8243*   8261*   8262    8333*   8378*
DVREX   044116            8886    8895#
DVRXA   007362            4949#   8209*   8229    8473*   8511    8616*   8628    8631    8892    9040*
DVRXQ   044040            8213    8477    8623    8884#
DVTCC   007350            4943#   8221*   8269    8482*   8500    8504    8576*   8584    8977    9026*
DVTCT   007352            4944#   8100*   8124*   8150*   8271*   8276    8279
DVTREX  044616            9030    9044#
DVTRX1  044610            9039    9043#
DVTR3   044176            8971    8979#   8992
DVTR4   044376            8982    9014#
DVTR4A  044456            9017    9025#
DVTR4B  044500            9024    9028#
DVTR5   044506            9015    9029#
DVTR5A  044566            9032    9040#
DVTXA   007346            4942#   8219*   8268    8480*   8499    8573*   8583    8976    9025*
DVTXRX  044120            8225    8486    8581    8626    8970#
DZM     020270            5096    5662#
ECHO  = 000037            4604#   5240
ECHOB = 000004            4517#   6815    7947    7958    7963    8232
EDDCK   015770            5076    5565#
EDDDE   016103            5079    5568#   8366    8375
EDDER   015753            5074    5564#
EDDLE   016046            5078    5567#   8352
EDDVI   016020            5075    5566#
EDEOP   016136            5080    5569#
EDMOS   016152            5570#
EDRXC   015725            5073    5563#
EDRXQ   015700            5072    5562#
EDTXC   015651            5071    5561#
EDTXQ   015625            5070    5560#
EFM11   015550            5555#   5858
EFM2    015453            5553#   5854
EF.CON= 000036 G          4478#   7206
EF.NEW= 000035 G          4478#   7209
EF.PWR- 000034 G          4478#
```

```
EF.RES= 000037 G        4478#   7204
EF.STA= 000040 G        4478#   7202
EMSG0   002221          4688    4714#
EMSG1   002222          4689    4716#
EMSG2   002223          4690    4718#
EMSG3   002224          4691    4720#
EMSG4   002324          4692    4726#
EMSG5   002416          4693    4731#
EMSG6   002517          4694    4735#
EMSG8   002647          4696    4749#
ENADD   007404          4959#   6571    7841*
ENDEVT  031432          6435    6445    6449    6506#
ENDIT   034530          7208    7298#
ENDQO = 000017          4588#   5287
EOP   = 000024          4542#   6157
ERRCNT  007400          4957#   6160    6163    8042*   8351*   8365*   8374*   8403    8727*   8990*   9004*   9123*   9162*
                        9218*   9291*   9381*
ERR1    025630 G        5849#   8366
ERR10   025720 G        5857#   8352
ERR13   026110 G        5880#   8728    8991    9005    9022    9037
ERR14   026142 G        5884#   8465
ERR2    025672 G        5853#   8375
ERR8    025750 G        5870#   9292
ERR9    026032 G        5875#   9124    9163    9219
ERX   = 000100          4552#   8077    8127    8154    8257    8275    8446    8620    9031
ETX   = 000200          4553#   8101    8154    8258    8274    8485    8502    8578    8590    9016
EVL   = 000004 G        4478#
EVMCTS  017240          5059    5604#
EVMDCD  017250          5061    5606#
EVMDSR  017244          5060    5605#
EVMOCG  017120          5597#
EVMOHD  017143          5602#   6515
EVMOST  017223          5603#   6530
EVMRI   017260          5063    5608#
EVMRTS  017254          5062    5607#
EVMSQD  017264          5064    5609#
EVMTM   017270          5065    5610#
EVTADD  010614          5108#   6460*   6463    6484*   6487    6492*   6495    6499*   6502
EVTBCT  010616          5109#   6461*   6463    6485*   6487    6493*   6495    6500*   6502
EVTEND  010454          5040#   6195    6442    7249
EVTF0   016310          5586#   6452
EVTF1   016403          5587#   6457
EVTF2   016432          5588#   6463
EVTF3   016504          5589#   6470
EVTF3C  016516          5590#   5876    5881    6471
EVTF3D  016533          5591#   5871    5885
EVTF4   016555          5592#   6495
EVTF4A  016657          5593#   6502
EVTF4B  016755          5594#   6487
EVTF5A  017034          5595#   5850
EVTLOG  007552          5038    5039#   6199    6432    6440    7246
EVTLST  010514          5070#   6457
EVTMIN  010610          5106#   6456*   6457
EVTPTR  007550          5038#   6185    6200*   6431    6447    7247*
EVTSEC  010606          5105#   6455*   6457
EVTTCK  010612          5107#   6454*   6457
```

```
EVTTMP  010620  5110#  6467*  6470  6486*  6487  6494*  6495  6501*  6502
EXIT  = 000057  4620#  5158  7598  7755
E$END = 002100  4228#
E$LOAD= 000035  4228#  4306
FACSIM  032134  6728#  7917
FHDPLX  007474  4999#  7261*  8784
FLAG    007502  5009#  8039*  8077*  8101*  8127*  8154*  8205  8215  8227  8257*  8258*  8264*  8266
                8274*  8275*  8283*  8446*  8456*  8472*  8485*  8487  8489  8502*  8503*  8507  8578*
                8590*  8620*  8736*  8793*  8885  8888*  8970  8972*  8981  8996  9003*  9008  9012
                9014  9016  9027*  9028*  9029  9031  9042*  9043*  9072*  9086*  9167  9172  9174*
                9211*  9240  9245  9247*  9298*  9303*  9373  9383*  9385*
F$AU  = 000015  4228#  7436  7463
F$AUTO= 000020  4228#  7342  7351
F$BGN = 000040  4228#  4249  5849  5853  5857  5870  5875  5880  5884  5888  6048  7099  7140
                7169  7241  7315  7342  7360  7374  7399  7410  7436  7447  7489  7559  7601
                9071  9085  9223  9388  9405  9529
F$CLEA= 000007  4228#  7360  7390
F$DU  = 000016  4228#  7399  7426
F$END = 000041  4228#  4249  5851  5855  5859  5873  5878  5882  5886  5888  6072  7131  7241
                7315  7331  7351  7374  7390  7410  7426  7447  7463  7489  7559  7601  9073
                9087  9223  9388  9445  9529
F$HARD= 000004  4228#  9405  9445
F$HW  = 000013  4228#  4351  4400
F$INIT= 000006  4228#  7169  7331
F$JMP = 000050  4228#  5888  7241  7315  7374  7410  7447  7559  7601
F$MOD = 000000  4228#  4249  9529
F$MSG = 000011  4228#  5849  5851  5853  5855  5857  5859  5870  5873  5875  5878  5880  5882
                5884  5886
F$PROT= 000021  4228#  7140  7146
F$PWR = 000017  4228#
F$RPT = 000012  4228#  7099  7131
F$SEG = 000003  4228#
F$SOFT= 000005  4228#
F$SRV = 000010  4228#  6048  6072  9071  9073  9085  9087
F$SUB = 000002  4228#
F$SW  = 000014  4228#
F$TEST= 000001  4228#  7489  9388
GETCL   035310  7556  7565#  7576  7580  7584  7587  7589  7596  7603  7605  7612  7621  7635
                7641  7655  7669  8019
GETPRM  034310  7211  7254#  7259
GETRCL  027074  6224#  6237  6242  6246
GOOD    007446  4979#  5850  8363*
GTRA2   034716  7502#
GTRA3   034760  7505  7510#
GTRA4   034766  7515#
GTRA5   035224  7511  7549#  8411  8462  8467
GTREX   040470  8039#
GTRX2   040524  8048#  8409  8592  8639
GTR9    040334  7561  7592  8012#
GTXRXB  034716  7501#
G$CNTO= 000200  4228#
G$DELM= 000372  4228#
G$DISP= 000003  4228#
G$EXCP= 000400  4228#
G$HILI= 000002  4228#
G$LOLI= 000001  4228#
```

```
G$NO  = 000000          4228#  6228   7567   8441   8565
G$OFFS= 000400          4228#  6228   7237   7567   8441   8565   9419   9436   9437   9438   9441
G$OFSI= 000376          4228#  6228   7237   7567   8441   8565   9419   9436   9437   9438   9441
G$PRMA= 000001          4228#  9436   9437
G$PRMD= 000002          4228#  6228   7237   7567   8441   8565   9438   9441
G$PRML= 000000          4228#  9419
G$RADA= 000140          4228#  6228   7567   8565
G$RADB= 0000C0          4228#
G$RADD= 000040          4228#  7237
G$RADL= 000120          4228#  9419
G$RADO= 000020          4228#  8441   9436   9437   9438   9441
G$XFER= 000004          4228#
G$YES = 000010          4228#  7237   9419   9436   9437   9438   9441
HALFDB= 002000          4672#  8781   8786
HELP  = 000000          2195#  4219   4245   4288   4315   4335   4353   4433   4447   4463   5375   5414   5435
                        5798   5810   5833   5899   7101   7116   7124   7148   7171   7317   7344   7362   7376
                        7401   7412   7438   7449   7476   7483   7491   9317   9333   9407   9472   9494   9507
                        9521
HELPDC= 000000          2200#  2209   4308   4371   4404   4630   4655   5339   5388   5420   5748   5861   7263
                        7303   8645   8664   8803   8845   8900   9049   9060   9075   9422   9453
HLP   = 000005          4578#  5152   5154   7586   7753
HLPEND  003230          4808#  7751
HLPF    013225          5463#  6297   7750
HLPTAB  003210          4800#  7749
HLP0    013147          5462#  7548
HLP1    013232          4800   5464#
HLP2    013245          4801   5465#
HLP3    013362          4802   5469#
HLP3A   013447          4803   5470#
HLP4    013474          4804   5471#
HLP4A   013553          4805   5472#
HLP5    013631          4806   5473#
HLP6    013721          4807   5474#
HOE   = 100000 G        4478#
IBE   = 010000 G        4478#
IDU   = 00C 40 G        4478#
IEO   = 000100          4674#  8745   9368
IER   = 020000 G        4478#
INDEX   007450          4981#  6315*  6316*  6323*  6324*  6336*  6342*  6366*  6367*  6374*  6375*  6406
INDEXE  007452          4982#  6317*  6325*  6337*  6343*  6368*  6376*  6410   6412
ININT = 000001          4546#  8996   9003   9072   9172   9174
INIT1   033762          7197   7201#
INTPRI  012404          5370#  7295*  7311   7312
INVEC   012400          5368#  7292*  7311
ISR   = 000100 G        4478#
IXE   = 004000 G        4478#
I$AU  = 000041          4228#  7436#  7463#
I$AUTO= 000041          4228#  7342#  7351#
I$CLN = 000041          4228#  7360#  7374   7390#
I$DU  = 000041          4228#  7399#  7426#
I$HRD = 000041          9405#  9445#
I$INIT= 000041          4228#  7169#  7241   7315   7331#
I$MOD = 000041          4228#  4249#  9529#
I$MSG = 000041          4228#  5849#  5851#  5853#  5855#  5857#  5859#  5870#  5873#  5875#  5878#  5880#  5882#
                        5884#  5886#
I$PROT= 000040          4228#  7140#
```

CZCLKB.P11 16-JUN-81 14:38 CROSS REFERENCE TABLE -- USER SYMBOLS

```
I$PTAB= 000041            4228#
I$PWR = 000041            4228#
I$RPT = 000041            4228#   7099#   7131#
I$SEG = 000041            4228#   7489
I$SETU= 000041            4228#
I$SRV = 000041            4228#   6048#   6072#   9071#   9073#   9085#   9087#
I$SUB = 000041            4228#   7489
I$TST = 000041            4228#   7489#   7559    7601    9223    9388#
J$JMP = 000167            4228#   5888    7410    7447
KDPM    020303            5098    5664#
KDZM    020307            5099    5665#
KEYWD1  003202            4796#   6244    6300*   6304*   6309*   7583    7586    7588    7590    7593    7598    7602    7604
                          7606    7743*   7746*   7753*   7755*   7757*   7761*   7776    7795    7837*   7848*   7851*   7913
                          7918*
KLM     020313            5100    5666#
LCLKEN= 000100            4509#   7220
LGDVE   026434            6129#   8496    8726    8989    9002    9021    9036    9122    9161    9217    9253    9380
LIS   = 000006            4498#   7936    8591
LISCK   043110            5018    8614#
LISCKA  043136            8616#   8635    8637
LISMOD= 000032            4599#   5184
LISP    014753            5526#   8615
LMDLOP= 000046            4611#   5323
LNCNT   007372            4954#   6176    6178*   6182*   8043*
LOE   = 040000 G          4478#
LOGCMD  026554            6151#   8376
LOGCML  026536            6147#   8353
LOGCMP  026520            6143#   8347
LOGDVI  026452            6134#   8045
LOGEOP  026572            6155#   8404
LOGEX   027044            6165    6202#
LOGRXC  026424            6126#   8231    8513    8630
LOGRXQ  026406            6121#   8214    8478    8624
LOGS1   026610            6114    6119    6124    6128    6160#
LOGS2   027036            6196    6200#
LOGS3   026642            6132    6141    6146    6150    6154    6158    6169#
LOGS4   026716            6177    6181#
LOGS5   026742            6171    6173    6184#
LOGTXC  026370            6116#   8270    8501    8585
LOGTXQ  026352            6111#   8223    8484    8577
LOGUNT  007460            4989#   7252*   7254*   7255    7258
LOOPS   003532            4909#   6801
LOSDAM  025361            5784#   9266
LOT   = 000010 G          4478#
LP0     014635            4909    5077    5501#   6800    9254
LP00    014636            5502#   6797
LP1     014645            4910    5503#
LP2     014656            4911    5504#
LP3     014664            4912    5505#
LP4     014677            4913    5506#
LULOOP= 004000            4670#   8746    8749
L$ACP   002110 G          4306#
L$APT   002036 G          4306#
L$AU    034702 G          4306    7436#
L$AUT   002070 G          4306#
L$AUTO  034646 G          4306    7342#
```

L$CCP	002106	G	4306#	
L$CLEA	034650	G	4306	7360#
L$CO	002032	G	4306#	
L$DEPO	002011	G	4306#	
L$DESC	012424	G	4306	5432#
L$DESP	002076	G	4306#	
L$DEVP	002060	G	4306#	
L$DISP	002124	G	4306	4333#
L$DLY	002116	G	4306#	
L$DTP	002040	G	4306#	
L$DTYP	002034	G	4306#	
L$DU	034674	G	4306	7399#
L$DUT	002072	G	4306#	
L$DVTY	012410	G	4306	5412#
L$EF	002052	G	4306#	
L$ENVI	002044	G	4306#	
L$ETP	002102	G	4306#	
L$EXP1	002046	G	4306#	
L$EXP4	002064	G	4306#	
L$EXP5	002066	G	4306#	
L$HARD	046060	G	4306	9405#
L$HIME	002120	G	4306#	
L$HPCP	002016	G	4306#	
L$HPTP	002022	G	4306#	
L$HW	002130	G	4306	4351#
L$ICP	002104	G	4306#	
L$INIT	033746	G	4306	7169#
L$LADP	002026	G	4306#	
L$LAST	046442	G	4306	9528#
L$LOAD	002100	G	4306#	
L$LUN	002074	G	4306#	
L$MREV	002050	G	4306#	
L$NAME	002000	G	4306#	
L$PRIO	002042	G	4306#	
L$PROT	033740	G	4306	7140#
L$PRT	002112	G	4306#	
L$REPP	002062	G	4306#	
L$REV	002010	G	4306#	
L$RPT	033732	G	4306	7099#
L$SPC	002056	G	4306#	
L$SPCP	002020	G	4306#	
L$SPTP	002024	G	4306#	
L$STA	002030	G	4306#	
L$TEST	002114	G	4306#	
L$TIML	002014	G	4306#	
L$UNIT	002012	G	4306#	7255
L10000	002150		4351	4400#
L10001	025670		5851#	
L10002	025716		5855#	
L10003	025746		5859#	
L10004	026030		5873#	
L10005	026106		5878#	
L10006	026140		5882#	
L10007	026176		5886#	5888
L10010	026350		6072#	
L10011	033736		7131#	

```
L10013  034644            7241    7315    7331#
L10014  034646            7351#
L10015  034672            7374    7390#
L10016  034700            7410    7426#
L10017  034706            7447    7463#
L10020  046054            7559    7601    9223    9388#
L10021  044626            9073#
L10022  044636            9087#
L10023  046132            9405    9445#
L5060   014771            5528#   7237
MAINTB= 000400            4671#   8781
MCLR  = 040000            4669#   8708
MLTYP   007472            4996#   6138    7544*   7550    7560*   7829    7930*   7949*   7991*   7993*   7995*   7997*   7999*
                          8004*   8012    8747    8759    8762
MOBITE  010476            5055#   6528
MOBITS  010460            5048#   6516
MOCHK = 000010            4518#   6819    7976
MODE    007504            5012#   8055
MODES   003514            4901#   6794
MODLOC= 000003            4503#   8762
MODREM= 000004            4504#   8759
MODS    010456            5044#   6167    8837*   8891*   8975*
MODTYP  007470            4993#   6137    7543*   7549    7828    7925*   7928*   7933*   7936*   7939*   7942*   7945*   7959
                          8001    8053    8241    8255    8272    8591*   8638*   8779
MOMSGS  010476            5059#   6517
MOP   = 000043            4505#   4608#
MOSC  = 000056            4619#   5261
MO0     014544            4901    5494#
MO1     014554            4902    5495#
MO2     014565            4903    5496#
MO3     014575            4904    5497#
MO4     014604            4905    5498#
MO5     014621            4906    5499#
MO6     014626            4907    5500#
MSG     002734            4774    4782#
MSGLIM= 000017            4485#   6736    7521    7618    7646    7652    7767    7782    8024    8030
MSGTRN  015241            5541#   7611    7620    7640    7654
MSGTRU  015272            5542#   6610
MSGTYP  007410            4963#   6651    7527*   7539*   7806*   7874*   7893*   7895*   7897*   7899*   7901*   7903*   7906*
                          8051*
MSG0    002220            4688    4703    4713#
MSG0C   002150            4688#
MSG1    002221            4689    4704    4715#
MSG1C   002152            4689#
MSG2    002222            4690    4705    4717#
MSG2C   002154            4690#
MSG3    002223            4691    4706    4719#
MSG3C   002156            4691#
MSG4    002224            4692    4707    4721#
MSG4C   002160            4692#
MSG5    002324            4693    4708    4727#
MSG5C   002162            4693#   7528    7540    7807    7904
MSG6    002416            4694    4709    4732#
MSG6C   002164            4694#   7907
MSG8    002646            4696    4711    4748#
MSG8C   002170            4696#
```

```
NEW     034302          7210    7252#   7256
NO    = 000036          4603#   5224    7953    7956    7982
NOBUF   007374          4955#   8040*   8401    9235*
NOCLK   015102          5538#   7240    7508
NOD0    010660          5151#
NOD1    010664          5152#
NOD10   010742          5159#
NOD100  011560          5239#
NOD101  011564          5240#
NOD102  011600          5241#
NOD103  011604          5256#
NOD104  011620          5257#
NOD105  011624          5258#
NOD106  011640          5259#
NOD107  011644          5261#
NOD11   010744          5160#
NOD110  011660          5262#
NOD111  011664          5265#
NOD112  011670          5268#
NOD113  011704          5269#
NOD114  011710          5270#
NOD115  011726          5271#
NOD116  011732          5272#
NOD117  011746          5273#
NOD12   010756          5161#
NOD120  011752          5274#
NOD121  011766          5275#
NOD122  011772          5276#
NOD123  012006          5277#
NOD124  012012          5278#
NOD125  012026          5279#
NOD126  012032          5280#
NOD127  012046          5281#
NOD13   010762          5162#
NOD130  012052          5282#
NOD131  012072          5283#
NOD132  012076          5286#
NOD133  012102          5287#
NOD134  012106          5288#
NOD135  012112          5289#
NOD136  012116          5290#
NOD137  012122          5291#
NOD14   010776          5163#
NOD140  012126          5292#
NOD141  012130          5295#
NOD142  012134          5296#
NOD143  012140          5297#
NOD144  012154          5298#
NOD145  012160          5299#
NOD146  012174          5300#
NOD147  012200          5303#
NOD15   011002          5164#
NOD150  012204          5304#
NOD151  012210          5305#
NOD152  012214          5308#
NOD153  012220          5319#
```

```
NOD154  012242          5320#
NOD155  012246          5321#
NOD156  012262          5322#
NOD157  012266          5323#
NOD16   011016          5165#
NOD160  012310          5324#
NOD161  012314          5325#
NOD162  012336          5326#
NOD163  012342          5329#
NOD164  012346          5330#
NOD165  012352          5331#
NOD166  012356          5336#
NOD167  027266          6253#
NOD17   011022          5166#
NOD170  027272          6254#
NOD171  027276          6255#
NOD172  027300          6256#
NOD173  027314          6257#
NOD174  027316          6258#
NOD175  027332          6259#
NOD176  027334          6260#
NOD177  027346          6261#
NOD2    010670          5153#
NOD20   011026          5167#
NOD200  027350          6262#
NOD201  027364          6263#
NOD202  027370          6264#
NOD203  027374          6265#
NOD204  027410          6266#
NOD205  027412          6267#
NOD206  027426          6268#
NOD207  027430          6269#
NOD21   011040          5168#
NOD210  027446          6270#
NOD211  027452          6271#
NOD212  027456          6272#
NOD213  027460          6273#
NOD214  027462          6274#
NOD22   011044          5169#
NOD23   011056          5170#
NOD24   011062          5171#
NOD25   011064          5175#
NOD26   011070          5176#
NOD27   011104          5177#
NOD3    010672          5154#
NOD30   011110          5178#
NOD31   011126          5179#
NOD32   011132          5180#
NOD33   011150          5181#
NOD34   011154          5182#
NOD35   011172          5183#
NOD36   011176          5184#
NOD37   011214          5185#
NOD4    010706          5155#
NOD40   011220          5186#
NOD41   011244          5187#
```

```
NOD42   011250            5188#
NOD43   011254            5189#
NOD44   011272            5190#
NOD45   011276            5191#
NOD46   011310            5192#
NOD47   011314            5196#
NOD5    010710            5156#
NOD50   011320            5197#
NOD51   011336            5198#
NOD52   011340            5199#
NOD53   011360            5200#
NOD54   011362            5205#
NOD55   011366            5206#
NOD56   011404            5207#
NOD57   011410            5208#
NOD6    010724            5157#
NOD60   011430            5209#
NOD61   011434            5212#
NOD62   011440            5213#
NOD63   011444            5214#
NOD64   011450            5215#
NOD65   011454            5216#
NOD66   011460            5217#
NOD67   011464            5218#
NOD7    010726            5158#
NOD70   011470            5219#
NOD71   011474            5222#
NOD72   011500            5223#
NOD73   011504            5224#
NOD74   011516            5225#
NOD75   011522            5226#
NOD76   011536            5227#
NOD77   011542            5238#
NONE  = 000000            4500#
NOTNUF= 000050            4613#   5162    5165    5166    5176    5215    5217    5223    5296
NULEVT  016250            5585#   6434
NULL  = 000000            4573#
NUM   = 000014            4585#   5304
NXMM    025310            5781#   9275
N10$    010664            5151    5152#
N100$   011314            5165    5168    5196#
N102$   011320            5197#
N104$   011340            5197    5199#
N110$   011362            5170    5205#
N111$   011366            5206#
N112$   011410            5206    5208#
N114$   011500            5222    5223#
N115$   011474            5179    5181    5183    5185    5187    5190    5192    5222#   5227    5239    5241    5262    5320
                          5322    5324    5326    5331
N116$   011516            5224    5225#
N117$   011542            5225    5226    5238#
N118$   011564            5238    5240#
N120$   011664            5207    5209    5265#
N121$   012130            5269    5271    5273    5275    5277    5279    5281    5288    5295#   5305
N122$   012160            5297    5299#
N123$   012134            5295    5296#
```

CZCLKB.P11 16-JUN-81 14:38 CROSS REFERENCE TABLE -- USER SYMBOLS

```
N125$   012356          5215    5217    5219    5223    5283    5296    5336#
N126$   012200          5298    5300    5303#
N130$   011604          5240    5256#
N131$   011624          5256    5258#
N132$   011644          5258    5261#
N140$   012214          5259    5308#
N141$   012220          5319#
N142$   012246          5319    5321#
N143$   012266          5321    5323#
N144$   012314          5323    5325#
N150$   012342          5257    5329#
N20$    011022          5164    5166#
N25$    011044          5167    5169#
N30$    011062          5166    5169    5171#   5175    5176    5177    5188    5191    5196    5199    5205    5208    5214
                        5216    5218    5261    5265    5286    5299    5303    5304    5308    5325    5329    5330
N40$    011002          5162    5164#
N42$    010672          5152    5154#
N43$    010710          5154    5156#
N44$    010726          5156    5158#
N45$    010744          5158    5160#
N46$    010762          5160    5162#
N50$    011434          5163    5212#
N51$    011440          5212    5213#
N52$    011444          5213    5214#
N60$    011670          5268#
N61$    011710          5268    5270#
N62$    011732          5270    5272#
N63$    011752          5272    5274#
N64$    011772          5274    5276#
N65$    012012          5276    5278#
N66$    012032          5278    5280#
N67$    012052          5280    5282#
N68$    012076          5282    5286#
N70$    012102          5287#   5291
N71$    012112          5287    5289#
N72$    012116          5289    5290#
N73$    012126          5290    5292#
N80$    011064          5161    5175#
N81$    011070          5176#
N82$    011132          5178    5180#
N83$    011154          5180    5182#
N84$    011176          5182    5184#
N85$    011220          5184    5186#
N86$    011250          5186    5188#
N87$    011276          5189    5191#
OFSET   007424          4970#   5850    8356*   8369*
OPBFPT  002520          4741#   8633
OPBUF   002524          4710    4742#   7884    8561    8565    8567    8573    8574    8586    8588    8616    8617    8634
                        8636
OPCNT   002166          4695#   7882*   8571*   8575    8576
OPEND   002646          4743#   8563
OPRMM   014764          5527#   8565
OPRMSG= 000015          4586#   5286
OPTN    046272          9441    9466#
OPTYP   012406          5371#   6313    6334    6352    6364    7296*   8710    8713    8742    8754    8791    9238    9279
                        9365
```

CZCLKB.P11 16-JUN-81 14:38 CROSS REFERENCE TABLE -- USER SYMBOLS

```
OTINT = 000002          4547#   9008    9086    9167    9211    9373
OUTHDL  045070          9011    9171    9210#   9384
OUTHEX  045672          9236    9242    9248    9293    9301    9306#
OUTH1   045160          9213    9225#
OUTH2   045614          9227    9295#
OUTH3   045650          9297    9303#
OUTH4   045656          9304#
OUTH6   045222          9233    9238#
OUTVEC  012402          5369#   7293*   7294*   7312
O$APTS= 000000          4228#   4306
O$AU  = 000001          4228#   4286#   4306
O$BGNR= 000001          4228#   4286#   4306
O$BGNS= 000000          4228#   4306
O$DU  = 000001          4228#   4286#   4306
O$ERRT= 000000          4228#   4306
O$GNSW= 000000          4228#   4306
O$POIN= 000001          4228#   4286#   4306
O$SETU= 000000          4228#   4306    9528
PARAM   007476          5000#   6140    6172    7546*   7552    7831    7947*   7958*   7963*   7981*   7984*   8014    8098*
                        8232    8237    8284    8329    8447*   8560*   8614*
PAS   = 000002          4494#   7928    7959    8255    8272
PASC  = 000042          4607#   5330
PASMOD= 000030          4597#   5180
PASS1   002650          4755#   8442*
PASS2   002651          4756#   8443*
PASS3   002652          4757#   8444*
PASS4   002653          4758#   8445*
PCADD   007462          4990#   5872    5877    9109*   9153*   9210*
PCK     014726          5510#   6810
PCLKCT= 001600          4511#   7227
PCLKEN= 000111          4510#   7229
PCPM    015516          5554#   5872    5877
PEC     014736          5512#   6814
PLCK    040664          5014    8123#
PLCK2   040664          8124#
PLCK3   040700          8126#
PMS     014745          5525#   6818
PNCK    014724          5509#   6813
PNEC    014734          5511#   6817
PNMS    014743          5524#   6821
PNST    014713          5507#   6809
PNT   = 001000 G        4478#
PRI   = 002C00 G        4478#
PRIOR   046244          9438    9465#
PRI00 = 000000 G        4478#   7314
PRI01 = 000040 G        4478#
PRI02 = 000100 G        4478#
PRI03 = 000140 G        4478#
PRI04 = 000200 G        4478#
PRI05 = 000240 G        4478#
PRI06 = 000300 G        4478#
PRI07 = 000340 G        4306    4478#   7372
PRNT  = 000055          4618#   5156    7588    7757
PROEM   025270          5780#   9278
PROTO = 000041          4606#
PROTOB= 000040          4520#   7979
```

```
PSCNT   007376          4956# 8041* 8399* 8402
PST     014715          5508# 6806
PTREND  007332          4933#
PTRTAB  006562          4932# 6739  6744  7518  7616  7645  7766  7781  7792  7799  8022
P$ACT   003550          4919# 6231* 6910  7570*
P$BUFA  003544          4917# 6229* 6865  7568*
P$CNT   003552          4920# 7002* 7009  7066* 7073* 7078
P$EXIT  032760          6868  6903# 6927  6932
P$GDBD  003561          4924# 6224* 6234  6360* 6925* 6930* 6938  6984* 7022* 7565* 7573  7866* 7915*
                        7962* 800 *
P$NNUF  003560          4923# 6225* 6239  6292* 6351* 6417* 7566* 7577  7740* 7762* 7775* 7794* 7842*
                        7921* 7929* 7948* 8006*
P$NUM   003554          4921# 6357  6359  6362  6370  6372  6378  6380  6997* 7000* 7002  7003* 7005*
                        7009* 7011* 7017* 7835  7841  7863  7868  7870  7988
P$RADX  003556          4922# 6962* 6965* 6972* 6979  7007  7015
P$TREE  003546          4918# 6230* 6866  7569*
P$TRV   032642          6233  6864# 7572
P$TR5   032652          6867# 6876  6896  6901
QCOPY = 000013          4584# 5299  7858
QRX   = 000004          4548# 8077  8127  8154  8205  8227  8257  8264  8275  8472  8507  8620  8885
                        8888  9042
QTX   = 000010          4549# 8101  8154  8215  8258  8266  8274  8283  8485  8489  8502  8578  8590
                        8970  8972  9027
QUALFG  003204          4798# 6232* 7571* 7855* 7858* 7861  7953* 7956  7964* 7982  7985*
QUALVL  003206          4799# 7633* 7667* 7852* 7870* 7871*
RDI   = 000200          4666# 9115
RDO   = 000200          4667# 9212  9306
REC   = 000000          4492#
RECMOD= 000031          4598# 5182
REPLOG  030574          6308  6428#
REPORT  027046          6217# 7113  7758
RESFLG  007466          4992# 7201* 7213* 7510
RESTRT  034242          7205  7221  7230  7238  7243#
RHLPEN  003246          4817# 6298
RHLPTB  003230          4810# 6296
RHLP0   014020          5476# 6222
RHLP1   014057          4810  5477#
RHLP2   014102          4811  5478#
RHLP3   014135          4812  5479#
RHLP4   014166          4813  5480#
RHLP5   014220          4814  5481#
RHLP6   014257          4815  5482#
RHLP7   014316          4816  5483#
RI    = 000200          4651# 5052
RMDLOP= 000047          4612# 5325
RNOTNF= 000007          4629# 6262  6264  6269
RPASS   007500          5008# 6139  7545* 7551  7557  7763* 7830  7988* 8405  8407*
RPBASE  030336          6330  6347  6381  6403#
RPEXT = 000002          4624# 6244  6258  6304
RPHLP = 000001          4623# 6254  6256  6300
RPLOG = 000003          4625# 6260  6309
RPSWE = 000004          4626# 6265
RPSWF = 000005          4627# 6267
RPSWO = 000006          4628# 6271
RPT     030642          5121  6437# 6465  6472  6483  6489  6496  6504
RPTAA   030712          6452#
```

```
RPTDCK  031356          5120    6499#
RPTDDE  031302          5123    6492#
RPTDER  031066          5118    6467#
RPTDLE  031356          5122    6498#
RPTDSP  010622          5114#   6458
RPTDVI  031162          5119    6474#
RPTEOP  031226          5124    6484#
RPTIV   014370          5484#   6359
RPTMSB  031442          6464    6515#
RPTTXQ  031014          5114    5115    5116    5117    6460#
RPTO    030706          6433    6444    6448    6451#
RPT1    030674          6441    6447#
RQI   = 000040          4665#   9110
RSEL4   045706          9034    9040    9304*   9310#
RSEL6   045710          9035    9041    9305*   9311#
RTS   = 000040          4650#   5051
RUN   = 000004          4577#   5160    7590    7761
RUNSBM  025437          5788#   9249
RXBIT = 000004          4673#   9296
RXBUF   004562          4930#   8049    8473    8474    8493    8514    8520    8527
RXC   = 000006          4535#   6127    6170
RXIDM   025461          5789#   9283
RXMTOT  007370          4952#   8036*   8076    8152    8333
RXM1    025561          5794#   8517
RXM2    025604          5795#   8522
RXNC    025541          5793#   8509
RXONLY  040572          5012    8074#
RXON2   040572          8075#
RXPTR   007332          4935#   7520*   8033*   8034*   8050    8075    8126    8153    8240    8331
RXQ   = 000004          4534#   6123
R10$    027272          6253    6254#
R11$    027300          6254    6256#
R12$    027316          6256    6258#
R125$   027462          6264    6274#
R13$    027334          6258    6260#
R14$    027350          6260    6262#
R15$    027370          6263    6264#
R16$    027412          6265    6267#
R17$    027430          6267    6269#
R30$    027460          6262    6269    6270    6271    6273#
SCM     020047          5638#   6144
SCMD    020102          5641#   6152
SCML    020071          5640#   6148
SDVE    020036          5637#   6130
SDVI    020060          5639#   6135
SECRM   020135          5649#   8534
SEL0    012360          5354#   7276*   8708*   8717    8724    8746*   8749*   8987    9000    9110*   9115    9120    9160
                        9378
SEL2    012364          5357#   7279*   7280*   8725    8745*   8988    9001    9121    9159    9212    9215    9225    9251
                        9296    9306*   9368*   9379
SEL4    012370          5360#   7283*   7284*   8707*   8739*   8837    8891    8892*   8975    8976*   9299    9304
SEL6    012374          5363#   7287*   7288*   8706*   8741*   8744*   8758*   8778*   8781*   8786*   8788    8838*   8893*
                        8977*   9216    9229    9300    9305
SEOP    020113          5642#   6156
SETET = 000060          4621#   5282    7583    7918
SETEXP= 000010          4581#   5206    7606    7848    7913
```

CZCLKB.P11 16-JUN-81 14:38 CROSS REFERENCE TABLE -- USER SYMBOLS

```
SETTRN= 000011           4582#   5208    7851
SHF0    015355           5543#   6803
SHF1    015413           5551#   6841
SHMSG   014430           5485#   7824
SHOW  = 000002           4575#   5167    7604    7746    7776    7795
SHTAB   003504           4897#   7815    7821
SHTEND  003513           4898#   7818
SHTYP0  014464           4892    5486#
SHTYP1  014473           4892    5487#
SHTYP2  014500           4892    5488#
SHTYP3  014505           4892    5489#
SHTYP4  014512           4892    5490#
SHTYP5  014520           4892    5491#
SHTYP6  014525           4892    5492#
SHTYP7  014533           4892    5493#
SHTYTB  003464           4892#   7824
SHWOP   032340           6481    6792#   7553    7832
SIZE  = 000012           4583#   5297    7855    7861
SMSC    020124           5643#
SQD   = 040000           4652#   5053
SRXQ    020025           5636#   6122
STADD   007402           4958#   6562    7835*
START   034036           7203    7213#
STATB = 000001           4515#   6172    6807    7970
STATUS= 000016           4587#   5238
STXC    020014           5635#   6117
STXQ    020003           5634#   6112
SVCGBL= 000000           4228#   4237#   4306    4333    4351    5412    5432    5849    5853    5857    5870    5875    5880
                         5884    6048    7099    7140    7169    7342    7360    7399    7436    9071    9085    9405    9528#
SVCINS= 000001           4228#   4234#   4306    4333    4351    5412    5432    5850    5851    5854    5855    5858    5859
                         5871    5872    5873    5876    5877    5878    5881    5882    5885    5886    5888    6072    6180
                         6183    6222    6228    6236    6241    6297    6359    6405    6414    6434    6452    6457    6463
                         6470    6471    6487    6495    6502    6515    6530    6564    6568    6570    6610    6803    6841
                         6983    7021    7131    7199    7202    7203    7204    7205    7206    7207    7209    7210    7217
                         7218    7223    7224    7232    7233    7237    7240    7241    7258    7259    7299    7311    7312
                         7314    7315    7331    7351    7372    7374    7390    7410    7426    7447    7463    7508    7548
                         7555    7556    7559    7567    7575    7579    7601    7611    7620    7640    7654    7750    7824
                         7865    7890    7961    8005    8018    8352    8366    8375    8441    8461    8465    8534    8565
                         8615    8633    8719    8728    8991    8995    9005    9022    9037    9073    9087    9114    9124
                         9163    9166    9219    9223    9292    9372    9388    9405    9419    9436    9437    9438    9441
                         9445    9528
SVCSUB= 000001           4228#   4236#
SVCTAG= 000C01           4228#   4238#   4400    5851    5855    5859    5873    5878    5882    5886    6072    6228    7131
                         7237    7331    7351    7390    7426    7463    7567    8441    8565    9073    9087    9388    9445
SVCTST= 000001           4228#   4235#   7489
S$LSYM= 010000           4228#   4400#   5851#   5855#   5859#   5873#   5878#   5882#   5886#   6072#   6228#   7131#   7237#
                         7331#   7351#   7390#   7426#   7463#   7567#   8441#   8565#   9073#   9087#   9388#   9445#
S1      034024           7207    7209#
S2      034100           7218    7223#
S3      034150           7224    7232#
S4      034216           7233    7240#
TABEX   015163           5539#   7620    7654
TAL   = 000005           4497#   7945    8638
TALCK   042670           5017    8559#   8587    8589
TALMOD= 000035           4602#   5191
TCURAD  007360           4947#   7529*   7532    7626    7631*   7802*
```

```
TEMP    007426  4971# 6113* 6118* 6123* 6127* 6131* 6136* 6145* 6149* 6153* 6157* 6170  6186
                6187* 6188* 6189  6567* 6568  6612* 6613* 6614  6653* 6654* 6660  6794* 6803
                6806* 6809* 6841  7823* 7824  7875* 7876* 7880  7883  8246* 8247* 8251  8526*
                8532* 8534
TEMP1   007430  4972# 6112* 6117* 6122* 6130* 6135* 6144* 6148* 6152* 6156* 6183  6409* 6414
                6801* 6803  6810* 6813* 6841
TEMP2   007432  4973# 6137* 6138* 6192  6413* 6414  6656* 6658* 6662  6802* 6803  6814* 6817*
                6841  8208* 8218* 8229* 8235  8249* 8250  8268* 8337* 8402* 8474* 8481* 8495*
                8499* 8511* 8574* 8583* 8617* 8628* 8723* 8986* 8999* 9018* 9033* 9119* 9158*
                9214* 9250* 9377*
TEMP3   007434  4974# 5858  5871  5876  5881  5885  6139* 6193  6408* 6414  6797* 6800* 6803
                8210* 8220* 8230* 8236  8247  8248  8249  8269* 8339* 8349  8403* 8441  8442
                8443  8444  8445  8476* 8483* 8493* 8500* 8512* 8575* 8584* 8619* 8629* 8724*
                8987* 9000* 9019* 9034* 9120* 9160* 9215* 9251* 9378*
TEMP4   007436  4975# 5854  5871  5876  5881  5885  6140* 6167* 6194  8346* 8355* 8360* 8361
                8372  8401* 8494* 8725* 8988* 9001* 9020* 9035* 9121* 9159* 9216* 9229* 9232
                9243  9255  9258  9261  9264  9267  9270  9273  9276  9281  9284  9287  9379*
TEMP5   007440  4976# 6818* 6821* 6841
TIMERS  007546  5034# 6066  6070* 8980* 8984
TIMER1  007542  5032# 6060  6062* 7503* 7504  8705* 8720  9111* 9112  9154* 9155  9371* 9375
TIMER2  007544  5033# 6063  6065*
TIMMIN  007534  5028# 6057* 6191  7243*
TIMOM   025413  5786# 9260
TIMSEC  007536  5029# 6054* 6055  6058* 6190  7244*
TIMTCK  007540  5030# 6051* 6053* 6068  6188  7245*
TM    = 001000  4653# 5054
TOINOT  044204  8981# 9006  9009  9013
TOIN1   044266  8985  8995#
TOIN2   044352  8997  9008#
TOORIO  044746  8738  8757  8771  8777  8836  8890  8974  9153# 9164  9369
TOOR1   045032  9156  9166#
TOOR2   045050  9169  9172#
TOOR3   044760  9155# 9173
TOTCC   007422  4969# 6604* 6605  6611* 6613  6615* 7515* 7608* 7609  7630  7637* 7638  7666
TRA   = 000001  4493# 7942
TRAMOD= 000034  4601# 5189
TRVACT  032762  6897  6908# 6924  6929  6934  6937  6957  7018  7036  7057  7081
TRVALN  033554  6886  7040#
TRVALP  033510  6885  7026#
TRVBIF  033066  6882  6937#
TRVBR   033056  6881  6934#
TRVBRC  033002  6895  6915# 6935  6940  6959  7023  7038  7059  7085
TRVDEC  033162  6888  6962#
TRVERR  033020  6879  6924#
TRVEXI  033040  6880  6929#
TRVNMA  033202  6963  6966#
TRVNOB  033012  6920# 6951  6958  7019  7037  7058
TRVNUM  033174  6884  6965#
TRVOCT  033174  6887  6964#
TRVSPA  033110  6883  6943#
TRVSTR  033642  6889  7063#
TSEL4   045702  9019  9025  9299* 9308#
TSEL6   045704  9020  9026  9300* 9309#
TTL   = 000001  4501# 7560  8747
TTLLOP= 000044  4609# 5319
TTOTCC  007356  4946# 6763  7516* 7608  7613  7630* 7798*
```

```
TXBUF   003562          4929#  6731   7529   7801   8494
TXC   = 000002          4533#  6118
TXMTOT  007354          4945#  6751   7535*  7615*  7617   7632*  7793   7797*  8016   8100   8124   8150
TXNC    025521          5792#  8492
TXONLY  040624          5013   8098#
TXON2   040632          8099#
TXPTR   007334          4936#  6744*  6746*  6747   6756*  6758   7519*  7533   7616*  7624*  7625   7629*  7799*
                        7800   8023*  8099   8125   8151
TXQ   = 000000          4532#  6113
T$ARGC= 000001          4306#  5850#  5854#  5858#  5871#  5872#  5876#  5877#  5881#  5885#  6180#  6183#  6222#
                        6236#  6241#  6297#  6359#  6405#  6414#  6434#  6452#  6457#  6463#  6470#  6471#  6487#
                        6495#  6502#  6515#  6530#  6564#  6568#  6570#  6610#  6803#  6841#  6983#  7021#  7240#
                        7508#  7548#  7575#  7579#  7611#  7620#  7640#  7654#  7750#  7824#  7865#  7890#  7961#
                        8005#  8018#  8461#  8534#  8615#  8633#
T$CODE= 005032          6228#  7237#  7567#  8441#  8565#  9419#  9436#  9437#  9438#  9441#
T$ERRN= 000023          4228#  8352#  8366#  8375#  8465#  8728#  8991#  9005#  9022#  9037#  9124#  9163#  9219#
                        9292#
T$EXCP= 000000          6228#  7237#  7567#  8441#  8565#  9436#  9437#  9438#  9441#
T$FLAG= 000040          5888#  7241#  7315#  7374#  7410#  7447#  7559#  7601#  9223#
T$GMAN= 000000          4228#  6228#  7237#  7567#  8441#  8565#
T$HILI= 000007          6228#  7237#  7567#  8441#  8565#  9436#  9437#  9438#  9441#
T$LAST= 000001          4228#  9528#
T$LOLI= 000000          6228#  7237#  7567#  8441#  8565#  9436#  9437#  9438#  9441#
T$LSYM= 010000          4228#  4400   5851   5855   5859   5873   5878   5882   5886   6072   7131   7331   7351
                        7390   7426   7463   9073   9087   9388   9445
T$LTNO= 000001          9528#
T$NEST= 177777          4228#  4249#  4351#  4400#  5849#  5851#  5853#  5855#  5857#  5859#  5870#  5873#  5875#
                        5878#  5880#  5882#  5884#  5886#  6048#  6072#  7099#  7131#  7140#  7146#  7169#  7331#
                        7342#  7351#  7360#  7390#  7399#  7426#  7436#  7463#  7489#  9071#  9073#  9085#  9087#
                        9388#  9405#  9445#  9529#
T$NS0 = 000000          4249#  9529
T$NS1 = 000004          4351#  4400   5849#  5851   5853#  5855   5857#  5859   5870#  5873   5875#  5878   5880#
                        5882   5884#  5886   6048#  6072   7099#  7131   7140#  7146   7169#  7331   7342#  7351
                        7360#  7390   7399#  7426   7436#  7463   7489#  9388   9405#  9445
T$NS2 = 000010          9071#  9073   9085#  9087
T$PTNU= 000000          4228#
T$SAVL= 177777          4228#
T$SEGL= 177777          4228#
T$SUBN= 000000          4228#  7489#
T$TAGL= 177777          4228#
T$TAGN= 010024          4228#  4351#  5849#  5853#  5857#  5870#  5875#  5880#  5884#  6048#  7099#  7140#  7169#
                        7342#  7360#  7399#  7436#  7489#  9071#  9085#  9405#
T$TEMP= 000C00          4333#  4400#  5851#  5855#  5859#  5873#  5878#  5882#  5886#  5888#  6072#  6228#  7131#
                        7146#  7237#  7241#  7315#  7331#  7351#  7374#  7390#  7410#  7426#  7447#  7463#  7559#
                        7567#  7601#  8441#  8565#  9073#  9087#  9223#  9388#  9419#  9436#  9437#  9438#  9441#
                        9445#  9529#
T$TEST= 000001          4228#  7489#  9528
T$TSTM= 177777          4228#  5850   5851   5854   5855   5858   5859   5871   5872   5873   5876   5877   5878
                        5881   5882   5885   5886   6180   6183   6222   6228   6236   6241   6297   6359   6405
                        6414   6434   6452   6457   6463   6470   6471   6487   6495   6502   6515   6530   6564
                        6568   6570   6610   6803   6841   6983   7021   7131   7199   7202   7204   7206   7209
                        7217   7223   7232   7237   7240   7241   7258   7299   7311   7312   7314   7315   7331
                        7351   7372   7374   7390   7426   7463   7508   7548   7555   7559   7567   7575   7579
                        7601   7611   7620   7640   7654   7750   7824   7865   7890   7961   8005   8018   8352
                        8366   8375   8441   8461   8465   8534   8565   8615   8633   8719   8728   8991   8995
                        9005   9022   9037   9114   9124   9166   9219   9223   9292   9372   9388
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
T$TSTS= 000001          4228#  7489#
T$$AU = 010017          7436#  7447   7463
T$$AUT= 010014          7342#  7351
T$$CLE= 010015          7360#  7374   7390
T$$DU = 010016          7399#  7410   7426
T$$HAR= 010023          9405#  9445
T$$HW = 010000          4351#  4400
T$$INI= 010013          7169#  7241   7315   7331
T$$MSG= 010007          5849#  5851   5853#  5855   5857#  5859   5870#  5873   5875#  5878   5880#  5882   5884#
                        5886   5888
T$$PRO= 010012          7140#
T$$RPT= 010011          7099#  7131
T$$SRV= 010022          6048#  6072   9071#  9073   9085#  9087
T$$TES= 010020          7489#  7559   7601   9223   9388
T1      034710 G        4333   7489#
UAM   = 000200 G        4478#
UNKM    020273          5097   5663#  8526
UPTABL  041132          8232#
UPTA1   041220          8238   8245#
UPTA3   041216          8242   8244#
UPTA4   041156          8233   8237#
UPTEX   041270          8244   8254#
VECTOR  046211          9437   9464#
X$    = 000215          4252#  5151#  5152#  5153#  5154#  5155#  5156#  5157#  5158#  5159#  5160#  5161#  5162#
                        5163#  5164#  5165#  5166#  5167#  5168#  5169#  5170#  5171#  5175#  5176#  5177#  5178#
                        5179#  5180#  5181#  5182#  5183#  5184#  5185#  5186#  5187#  5188#  5189#  5190#  5191#
                        5192#  5196#  5197#  5198#  5199#  5200#  5205#  5206#  5207#  5208#  5209#  5212#  5213#
                        5214#  5215#  5216#  5217#  5218#  5219#  5222#  5223#  5224#  5225#  5226#  5227#  5238#
                        5239#  5240#  5241#  5256#  5257#  5258#  5259#  5261#  5262#  5265#  5268#  5269#  5270#
                        5271#  5272#  5273#  5274#  5275#  5276#  5277#  5278#  5279#  5280#  5281#  5282#  5283#
                        5286#  5287#  5288#  5289#  5290#  5291#  5292#  5295#  5296#  5297#  5298#  5299#  5300#
                        5303#  5304#  5305#  5308#  5319#  5320#  5321#  5322#  5323#  5324#  5325#  5326#  5329#
                        5330#  5331#  5336#  6253#  6254#  6255#  6256#  6257#  6258#  6259#  6260#  6261#  6262#
                        6263#  6264#  6265#  6266#  6267#  6268#  6269#  6270#  6271#  6272#  6273#  6274#
X$ALWA= 000000          4228#
X$FALS= 000040          4228#
X$OFFS= 000400          4228#
X$TRUE= 000020          4228#
$PATCH  046356          9518#
.     = 046442          4224#  4736#  4742#  4774   4781   4795#  4899#  4929#  4930#  4931#  4932#  5039#  5040#
                        5154#  5158#  5162#  5169#  5176#  5178#  5184#  5186#  5197#  5199#  5206#  5208#  5224#
                        5226#  5238#  5240#  5256#  5258#  5268#  5270#  5272#  5274#  5276#  5282#  5297#  5299#
                        5323#  5412#  5432#  5529#  5612#  5624#  5625#  5626#  5796#  5888   6256#  6258#  6262#
                        6267#  6269#  6424#  7241   7315   7374   7410   7447   7559   7601   9223   9470#  9519#
```

BCOMPL	15#	4228#	7203	7205	7210	7556									
BERROR	19#	4228#													
BGNAU	23#	4228#	7436												
BGNAUT	31#	4228#	7342												
BGNCLN	39#	4228#	7360												
BGNDU	47#	4228#	7399												
BGNHRD	55#	4228#	9405												
BGNHW	66#	4228#	4351												
BGNINI	77#	4228#	7169												
BGNMOD	85#	4228#	4249												
BGNMSG	98#	4228#	5849	5853	5857	5870	5875	5880	5884						
BGNPRO	106#	4228#	7140												
BGNPTA	114#	4228#													
BGNRPT	144#	4228#	7099												
BGNSEG	152#	4228#													
BGNSET	161#	4228#													
BGNSFT	182#	4228#													
BGNSRV	193#	4228#	6048	9071	9085										
BGNSUB	201#	4228#													
BGNSW	225#	4228#													
BGNTST	236#	4228#	7489												
BNCOMP	266#	4228#	7207	7218	7224	7233	7259								
BNERRO	270#	4228#													
BREAK	274#	4228#	8719	8995	9114	9166	9372								
BRESET	278#	4228#													
CKLOOP	282#	4228#													
CLI	4256#	5151	5152	5153	5154	5155	5156	5157	5158	5159	5160	5161	5162	5163	5164
	5165	5166	5167	5168	5169	5170	5171	5175	5176	5177	5178	5179	5180	5181	5182
	5183	5184	5185	5186	5187	5188	5189	5190	5191	5192	5196	5197	5198	5199	5200
	5205	5206	5207	5208	5209	5212	5213	5214	5215	5216	5217	5218	5219	5222	5223
	5224	5225	5226	5227	5238	5239	5240	5241	5256	5257	5258	5259	5261	5262	5265
	5268	5269	5270	5271	5272	5273	5274	5275	5276	5277	5278	5279	5280	5281	5282
	5283	5286	5287	5288	5289	5290	5291	5292	5295	5296	5297	5298	5299	5300	5303
	5304	5305	5308	5319	5320	5321	5322	5323	5324	5325	5326	5329	5330	5331	5336
	6253	6254	6255	6256	6257	6258	6259	6260	6261	6262	6263	6264	6265	6266	6267
	6268	6269	6270	6271	6272	6273	6274								
CLOCK	286#	4228#	7217	7223											
CLOSE	292#	4228#													
CLRVEC	296#	4228#													
COMMEN	301#	4228#													
DELAY	322#	4228#													
DESCRI	317#	4228#	5432												
DEVTYP	341#	4228#	5412												
DISPAT	346#	4228#	4333												
DISPLA	360#	4228#													
DOCLN	376#	4228#	7199												
DODU	380#	4228#													
DORPT	385#	4228#													
ENDAU	389#	4228#	7463												
ENDAUT	401#	4228#	7351												
ENDCLN	413#	4228#	7390												
ENDCOM	425#	4228#													
ENDDU	441#	4228#	7426												
ENDHRD	453#	4228#	9445												
ENDHW	465#	4228#	4400												
ENDINI	475#	4228#	7331												

```
ENDMOD   487#  4228#  9529
ENDMSG   500#  4228#  5851  5855  5859  5873  5878  5882  5886
ENDPRO   512#  4228#  7146
ENDPTA   520#  4228#
ENDRPT   529#  4228#  7131
ENDSEG   541#  4228#
ENDSET   555#  4228#
ENDSFT   568#  4228#
ENDSRV   580#  4228#  6072  9073  9087
ENDSUB   596#  4228#
ENDSW    614#  4228#
ENDTST   624#  4228#  9388
EQUALS   642#  4228#  4478
ERRDF    714#  4228#
ERRHRD   718#  4228#  8465
ERROR    722#  4228#
ERRSF    726#  4228#
ERRSOF   730#  4228#  8352  8366  8375  8728  8991  9005  9022  9037  9124  9163  9219  9292
ERRTBL   734#  4228#
ESCAPE   744#  4228#  9223
EXIT     771#  4228#  5888  7241  7315  7374  7410  7447  7559  7601
FEQUAL   810#  4228#
GETBYT   824#  4228#
GETPRI   834#  4228#
GETWOR   829#  4228#
GMANIA   839#  4228#
GMANID   848#  4228#  6228  7237  7567  8441  8565
GMANIL   859#  4228#
GPHARD   868#  4228#  7258
GPRMA    874#  4228#  9436  9437
GPRMD    903#  4228#  6228#  7237#  7567#  8441#  8565#  9438  9441
GPRML    934#  4228#  9419
HEADER   954#  4228#  4306
INLOOP   962#  4228#
IOSETU   966#  4228#
IOSTAR   974#  4228#
KT11     982#  4228#
LASTAD  1147#  4228#  9528
MANUAL  1162#  4228#  7555
MEMORY  1166#  4228#
MSBYTE  2000#  4228#  4306#
MSCHEC  2118#  4228#  5888#  7241#  7315#  7374#  7410#  7447#  7559#  7601#
MSCNTO  2182#  4228#  6228#  7237#  7567#  8441#  8565#  9419#  9436#  9437#  9438#  9441#
MSCOUN  2066#  4228#  5850#  5854#  5858#  5871#  5872#  5876#  5877#  5881#  5885#  6180#  6183#  6222#  6236#
        6241#  6297#  6359#  6405#  6414#  6434#  6452#  6457#  6463#  6470#  6471#  6487#  6495#  6502#  6515#
        6530#  6564#  6568#  6570#  6610#  6803#  6841#  6903#  7021#  7240#  7508#  7548#  7575#  7579#  7611#
        7620#  7640#  7654#  7750#  7824#  7865#  7890#  7961#  8005#  8018#  8461#  8534#  8615#  8633#
MSDATA  1867#  4228#  4306#  5412#  5432#
MSDECR  2029#  4228#  4400#  5851#  5855#  5859#  5873#  5878#  5882#  5886#  6072#  7131#  7146#  7331#  7351#
        7390#  7426#  7463#  9073#  9087#  9388#  9445#  9529#
MSDEFA  2170#  4228#  6228#  7237#  7567#  8441#  8565#  9419#  9436#  9437#  9438#  9441#
MSENDE  2074#  4228#  4400#  5851#  5855#  5859#  5873#  5878#  5882#  5886#  6072#  7131#  7331#  7351#  7390#
        7426#  7463#  9073#  9087#  9388#  9445#  9529#
MSERRI  1649#  4228#  8352#  8366#  8375#  8465#  8728#  8991#  9005#  9022#  9037#  9124#  9163#  9219#  9292#
MSESCA  2006#  4228#  9223#
MSESCS  2010#  4228#  9223#
```

M$EXCP	2101#	4228#	6228#	7237#	7567#	8441#	8565#	9436#	9437#	9438#	9441#				
M$EXIT	2014#	4228#	5888#	7241#	7315#	7374#	7410#	7447#	7559#	7601#					
M$EXSE	2022#	4228#	5888#	7241#	7315#	7374#	7410#	7447#	7559#	7601#					
M$EXTJ	2018#	4228#	5888#	7241#	7315#	7374#	7410#	7447#	7559#	7601#					
M$GEN	2038#	4228#	4306#	4333#	4351#	4400#	5412#	5432#	5849#	5851#	5853#	5855#	5857#	5859#	5870#
	5873#	5875#	5878#	5880#	5882#	5884#	5886#	6048#	6072#	6228#	7099#	7131#	7140#	7169#	7237#
	7331#	7342#	7351#	7360#	7390#	7399#	7426#	7436#	7463#	7489#	7567#	8441#	8565#	9071#	9073#
	9085#	9087#	9388#	9405#	9445#	9528#									
M$GENB	1938#	4228#	6228#	7237#	7567#	8441#	8565#								
M$GETS	2035#	4228#	4400#	5851#	5855#	5859#	5873#	5878#	5882#	5886#	6072#	7131#	7146#	7331#	7351#
	7390#	7426#	7463#	9073#	9087#	9388#	9445#	9529#							
M$GETT	1877#	4228#	5888#	7241#	7315#	7374#	7410#	7447#	7559#	7601#	9223#				
M$GNGB	1902#	4228#	4249#	4306#	4333#	4351#	5412#	5432#	5849#	5853#	5857#	5870#	5875#	5880#	5884#
	6048#	7099#	7140#	7169#	7342#	7360#	7399#	7436#	9071#	9085#	9405#	9528#			
M$GNIN	2049#	4228#	4306#	4333#	4351#	5412#	5432#	5850#	5851#	5854#	5855#	5858#	5859#	5871#	5872#
	5873#	5876#	5877#	5878#	5881#	5882#	5885#	5886#	5888#	6072#	6180#	6183#	6222#	6228#	6236#
	6241#	6297#	6359#	6405#	6414#	6434#	6452#	6457#	6463#	6470#	6471#	6487#	6495#	6502#	6515#
	6530#	6564#	6568#	6570#	6610#	6803#	6841#	6983#	7021#	7131#	7199#	7202#	7203#	7204#	7205#
	7206#	7207#	7209#	7210#	7217#	7218#	7223#	7224#	7232#	7233#	7237#	7240#	7241#	7258#	7259#
	7299#	7311#	7312#	7314#	7315#	7331#	7351#	7372#	7374#	7390#	7410#	7426#	7447#	7463#	7508#
	7548#	7555#	7556#	7559#	7567#	7575#	7579#	7601#	7611#	7620#	7640#	7654#	7750#	7824#	7865#
	7890#	7961#	8005#	8018#	8352#	8366#	8375#	8441#	8461#	8465#	8534#	8565#	8615#	8633#	8719#
	8728#	8991#	8995#	9005#	9022#	9037#	9073#	9087#	9114#	9124#	9163#	9166#	9219#	9223#	9292#
	9372#	9388#	9405#	9419#	9436#	9437#	9438#	9441#	9445#	9528#					
M$GNLS	1913#	4228#	6228#	7237#	7567#	8441#	8565#								
M$GNSU	1898#	4228#													
M$GNTA	1890#	4228#	4400#	5851#	5855#	5859#	5873#	5878#	5882#	5886#	6072#	7131#	7331#	7351#	7390#
	7426#	7463#	9073#	9087#	9388#	9445#									
M$GNTE	1894#	4228#	7489#												
M$HAPT	1739#	4228#	4306#												
M$HNAP	1824#	4228#	4306#												
M$INCR	2026#	4228#	4249#	4351#	5849#	5850#	5851#	5853#	5854#	5855#	5857#	5858#	5859#	5870#	5871#
	5872#	5873#	5875#	5876#	5877#	5878#	5880#	5881#	5882#	5884#	5885#	5886#	6048#	6180#	6183#
	6222#	6228#	6236#	6241#	6297#	6359#	6405#	6414#	6434#	6452#	6457#	6463#	6470#	6471#	6487#
	6495#	6502#	6515#	6530#	6564#	6568#	6570#	6610#	6803#	6841#	6983#	7021#	7099#	7131#	7140#
	7169#	7199#	7202#	7204#	7206#	7209#	7217#	7223#	7232#	7237#	7240#	7241#	7258#	7299#	7311#
	7312#	7314#	7315#	7331#	7342#	7351#	7360#	7372#	7374#	7390#	7399#	7426#	7436#	7463#	7489#
	7508#	7548#	7555#	7559#	7567#	7575#	7579#	7601#	7611#	7620#	7640#	7654#	7750#	7824#	7865#
	7890#	7961#	8005#	8018#	8352#	8366#	8375#	8441#	8461#	8465#	8534#	8565#	8615#	8633#	8719#
	8728#	8991#	8995#	9005#	9022#	9037#	9071#	9085#	9114#	9124#	9163#	9166#	9219#	9223#	9292#
	9372#	9388#	9405#												
M$IOSE	1700#	4228#													
M$LDRO	1942#	4228#	7202#	7204#	7206#	7209#	7217#	7223#	7258#	7314#	7372#				
M$MASK	1671#	4228#													
M$MCHI	4#	4228#													
M$MCLO	1624#	4228#													
M$MSK1	1677#	4228#													
M$POP	1881#	4228#	4400#	5851#	5855#	5859#	5873#	5878#	5882#	5886#	6072#	7131#	7146#	7331#	7351#
	7390#	7426#	7463#	9073#	9087#	9388#	9445#	9529#							
M$PRIN	1636#	4228#	5850#	5854#	5858#	5871#	5872#	5876#	5877#	5881#	5885#	6180#	6183#	6222#	6236#
	6241#	6297#	6359#	6405#	6414#	6434#	6452#	6457#	6463#	6470#	6471#	6487#	6495#	6502#	6515#
	6530#	6564#	6568#	6570#	6610#	6803#	6841#	6983#	7021#	7240#	7508#	7548#	7575#	7579#	7611#
	7620#	7640#	7654#	7750#	7824#	7865#	7890#	7961#	8005#	8018#	8461#	8534#	8615#	8633#	
M$PUSH	1631#	4228#	4249#	4351#	5849#	5853#	5857#	5870#	5875#	5880#	5884#	6048#	7099#	7140#	7169#
	7342#	7360#	7399#	7436#	7489#	9071#	9085#	9405#							
M$PUT	1972#	4228#	5850#	5854#	5858#	5871#	5872#	5876#	5877#	5881#	5885#	6180#	6183#	6222#	6236#

```
          6241#  6297#  6359#  6405#  6414#  6434#  6452#  6457#  6463#  6470#  6471#  6487#  6495#  6502#  6515#
          6530#  6564#  6568#  6570#  6610#  6803#  6841#  6983#  7021#  7240#  7299#  7311#  7312#  7508#  7548#
          7575#  7579#  7611#  7620#  7640#  7654#  7750#  7824#  7865#  7890#  7961#  8005#  8018#  8461#  8534#
          8615#  8633#
M$PUT1    1981#  4228#  5850#  5854#  5858#  5871#  5872#  5876#  5877#  5881#  5885#  6180#  6183#  6222#  6236#
          6241#  6297#  6359#  6405#  6414#  6434#  6452#  6457#  6463#  6470#  6471#  6487#  6495#  6502#  6515#
          6530#  6564#  6568#  6570#  6610#  6803#  6841#  6983#  7021#  7240#  7299#  7311#  7312#  7508#  7548#
          7575#  7579#  7611#  7620#  7640#  7654#  7750#  7824#  7865#  7890#  7961#  8005#  8018#  8461#  8534#
          8615#  8633#
M$RADI    2077#  4228#  6228#  7237#  7567#  8441#  8565#  9419#  9436#  9437#  9438#  9441#
M$RBR0    1952#  4228#
M$RNR0    1962#  4228#  7217#  7223#  7258#
M$SETS    2032#  4228#  4249#  4351#  5849#  5853#  5857#  5870#  5875#  5880#  5884#  6048#  7099#  7140#  7169#
          7342#  7360#  7399#  7436#  7489#  9071#  9085#  9405#
M$STAR    1733#  4228#
M$SVC     1933#  4228#  5850#  5851#  5854#  5855#  5858#  5859#  5871#  5872#  5873#  5876#  5877#  5878#  5881#
          5882#  5885#  5886#  5888#  6180#  6183#  6222#  6228#  6236#  6241#  6297#  6359#  6405#  6414#  6434#
          6452#  6457#  6463#  6470#  6471#  6487#  6495#  6502#  6515#  6530#  6564#  6568#  6570#  6610#  6803#
          6841#  6983#  7021#  7131#  7199#  7202#  7204#  7206#  7209#  7217#  7223#  7232#  7237#  7240#  7241#
          7258#  7299#  7311#  7312#  7314#  7315#  7331#  7351#  7372#  7374#  7390#  7410#  7426#  7447#  7463#
          7508#  7548#  7555#  7559#  7567#  7575#  7579#  7601#  7611#  7620#  7640#  7654#  7750#  7824#  7865#
          7890#  7961#  8005#  8018#  8352   8366   8375   8441#  8461#  8465   8534#  8565#  8615#  8633#  8719#
          8728   8991   8995#  9005   9022   9037   9114#  9124   9163   9166#  9219   9223#  9292   9372#  9388#
M$TLAB    1929#  4228#  5850#  5851#  5854#  5855#  5858#  5859#  5871#  5872#  5873#  5876#  5877#  5878#  5881#
          5882#  5885#  5886#  6180#  6183#  6222#  6228#  6236#  6241#  6297#  6359#  6405#  6414#  6434#  6452#
          6457#  6463#  6470#  6471#  6487#  6495#  6502#  6515#  6530#  6564#  6568#  6570#  6610#  6803#  6841#
          6983#  7021#  7131#  7199#  7202#  7204#  7206#  7209#  7217#  7223#  7232#  7237#  7240#  7241#  7258#
          7299#  7311#  7312#  7314#  7315#  7331#  7351#  7372#  7374#  7390#  7426#  7463#  7508#  7548#  7555#
          7559#  7567#  7575#  7579#  7601#  7611#  7620#  7640#  7654#  7750#  7824#  7865#  7890#  7961#  8005#
          8018#  8352#  8366#  8375#  8441#  8461#  8465#  8534#  8565#  8615#  8633#  8719#  8728#  8991#  8995#
          9005#  9022#  9037#  9114#  9124#  9163#  9166#  9219#  9223#  9292#  9372#  9388#
M$TSTL    1921#  4228#  5850#  5851#  5854#  5855#  5858#  5859#  5871#  5872#  5873#  5876#  5877#  5878#  5881#
          5882#  5885#  5886#  6180#  6183#  6222#  6228#  6236#  6241#  6297#  6359#  6405#  6414#  6434#  6452#
          6457#  6463#  6470#  6471#  6487#  6495#  6502#  6515#  6530#  6564#  6568#  6570#  6610#  6803#  6841#
          6983#  7021#  7131#  7199#  7202#  7204#  7206#  7209#  7217#  7223#  7232#  7237#  7240#  7241#  7258#
          7299#  7311#  7312#  7314#  7315#  7331#  7351#  7372#  7374#  7390#  7426#  7463#  7508#  7548#  7555#
          7559#  7567#  7575#  7579#  7601#  7611#  7620#  7640#  7654#  7750#  7824#  7865#  7890#  7961#  8005#
          8018#  8352#  8366#  8375#  8441#  8461#  8465#  8534#  8565#  8615#  8633#  8719#  8728#  8991#  8995#
          9005#  9022#  9037#  9114#  9124#  9163#  9166#  9219#  9223#  9292#  9372#  9388#
M$WORD    1994#  4228#  4306#  4333#  5888#  6228#  7237#  7241#  7315#  7374#  7410#  7447#  7559#  7567#  7601#
          8352#  8366#  8375#  8441#  8465#  8565#  8728#  8991#  9005#  9022#  9037#  9124#  9163#  9219#  9292#
          9419#  9436#  9437#  9438#  9441#  9528
M$XFER    1682#  4228#
NODCL     4261#  5151   5152   5153   5154   5155   5156   5157   5158   5159   5160   5161   5162   5163   5164
          5165   5166   5167   5168   5169   5170   5171   5175   5176   5177   5178   5179   5180   5181   5182
          5183   5184   5185   5186   5187   5188   5189   5190   5191   5192   5196   5197   5198   5199   5200
          5205   5206   5207   5208   5209   5212   5213   5214   5215   5216   5217   5218   5219   5222   5223
          5224   5225   5226   5227   5238   5239   5240   5241   5256   5257   5258   5259   5261   5262   5265
          5268   5269   5270   5271   5272   5273   5274   5275   5276   5277   5278   5279   5280   5281   5282
          5283   5286   5287   5288   5289   5290   5291   5292   5295   5296   5297   5298   5299   5300   5303
          5304   5305   5308   5319   5320   5321   5322   5323   5324   5325   5326   5329   5330   5331   5336
          6253   6254   6255   6256   6257   6258   6259   6260   6261   6262   6263   6264   6265   6266   6267
          6268   6269   6270   6271   6272   6273   6274
OPEN      1171#  4228#
POINTE    1176#  4228#  4286
PRINTB    1239#  4228#  5850   5854   5858   5871   5872   5876   5877   5881   5885
```

CZCLKB.P11 16-JUN-81 14:38 CROSS REFERENCE TABLE -- MACRO NAMES

PRINTF	1279#	4228#	6180	6183	6222	6236	6241	6297	6359	6405	6414	6564	6568	6570	6610
	6983	7021	7240	7508	7548	7575	7579	7611	7620	7640	7654	7750	7824	7865	7890
	7961	8005	8018	8461	8534	8615	8633								
PRINTS	1319#	4228#	6434	6452	6457	6463	6470	6471	6487	6495	6502	6515	6530	6803	6841
PRINTX	1359#	4228#													
READBU	1399#	4228#	7232												
READEF	1403#	4228#	7202	7204	7206	7209									
RFLAGS	1408#	4228#													
SETPRI	1413#	4228#	7314	7372											
SETVEC	1418#	4228#	7299	7311	7312										
SLASH	1424#	4228#													
STARS	1438#	4228#													
SVC	1452#	4227#	4228												
XFER	1612#	4228#	5888#	7241#	7315#	7374#	7410#	7447#	7559#	7601#					
XFERF	1616#	4228#													
XFERT	1620#	4228#													

. ABS. 046442 000

ERRORS DETECTED: 0

CZCLKB/I,CZCLKB.SEQ/CRF/DOC=SVC34R.MLB,CZCLKB.P11
RUN-TIME: 24 30 3 SECONDS
RUN-TIME RATIO: 68/59=1.1
CORE USED: 21K (41 PAGES)

DOCUMENT PAGES: 199