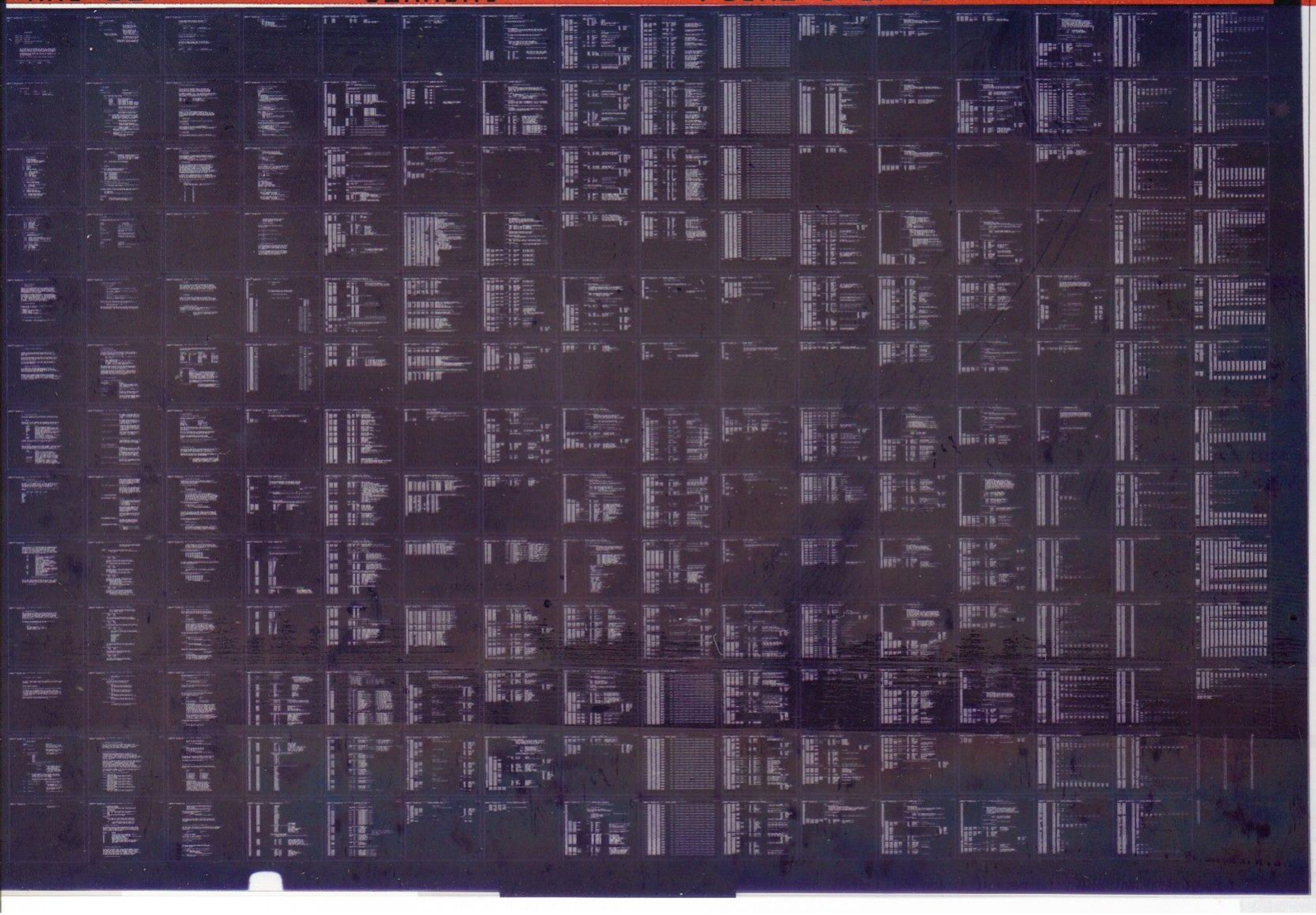
KMS-BL

KMS11-BL PDP11 DCLT
CZKMUAO

AH-T017A-MC
FICHE 1 OF 1



```
2207                                .TITLE  CZKMUA0 KMS11-BL PDP-11 DCLT
2208
```

.REM &

IDENTIFICATION

PRODUCT CODE: AC-T015A-MC

PRODUCT NAME: CZKMUA0 KMS11-BL PDP-11 DCLT

PRODUCT DATE: MARCH-1982

MAINTAINER: MERRIMACK DIAGNOSTIC ENGINEERING

AUTHOR: GLORIA MEREDITH

REVISION HISTORY:

REV	DATE	AUTHOR	REASON
A	24-MAR-82	G. MEREDITH	ORIGINAL ISSUE Of KMS11-BL PDP-11 DCLT

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 POINT TO POINT COMMUNICATION LINKS BETWEEN KMS11-BL OR KMS11-BM AND OTHER DDCMP SUPPORTED STATIONS. THIS DCLT PROGRAM WILL PROVIDE THE COVERAGE NECESSARY TO DETECT FAILURES OF 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 KMS11-BL 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
- A KMS11-BL OR KMS11-BM REMOTE SINGLE LINE DDCMP

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 DDCMP DEVICES 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 DDCVP DEVICES BUT TO TEST THE COMMUNICATION LINK TO WHICH THEY ARE CONNECTED.

1.5 ASSUMPTIONS - RESTRICTIONS

IT IS ASSUMED THAT THE KMS11 COMMUNICATIONS DEVICE 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 KMS11 SUPPORTS DDCMP OPERATION IN THE FIRMWARE, THE PDP-11 DCLT PROGRAM IS UNABLE TO CONTROL OR KNOW EXACTLY WHAT IS BEING TRANSMITTED AT ANY GIVEN TIME. ALL DATA MESSAGES ARE ENCLOSED IN A DDCMP ENVELOPE AND THERE MAY ALSO BE CONTROL MESSAGES (AKS, NAKS,....) BEING TRANSMITTED. BECAUSE OF THIS PLEASE BEWARE IF IF YOU ARE SCOPING DATA.

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

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

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

2.2 SWITCHES

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

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

EXAMPLE OF SWITCH USAGE:

```
        STAR1/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 KMS11-BL DCLT PROGRAM WILL NOT USE MORE THAN ONE UNIT.
THE HARDWARE INFORMATION REQUESTED WILL BE:

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

UNIT 0
FULL DUPLEX OPERATION :  (L) Y ?
KMS11 CSR ADDRESS : (O)  164100 ?
INTERRUPT VECTOR ADDRESS:  (O)  400  ?
INTERRUPT PRIORITY : (O)   5  ?
```

2.5 DATA COMM. LINK TEST COMMANDS

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

MESSAGE COMMANDS AVAILABLE:

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

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

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

2.5.1 MESSAGE COMMANDS

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

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

WHERE THE OPTIONAL "QUAL" IS:

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

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

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

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

FOR THE MESSAGES IN THE
TRANSMIT LIST

2.5.2 STATISTICAL COMMANDS

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

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

2.5.3 RUN COMMAND

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

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

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

=ACTIVE	(FORCES /NOECHO ,NO LOOPING)
=PASSIVE	(FORCES NO LOOPING)
=RECEIVE	(FORCES /NOECHO ,NO LOOPING)
=LISTEN	(FORCES /NOECHO ,NO LOOPING, /NOCHECK)
=TRANSMIT	(FORCES /NOECHO ,NO LOOPING, /NOCHECK)
=TALK	(FORCES /NOECHO ,NO LOOPING, /NOCHECK)
=DOWNLINELOAD	(FORCES /NOECHO ,NO LOOPING, /NOCHECK,

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

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

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

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

/STATUS/NOSTATUS	ENABLES/DISABLES PRINTING OF PROGRAM STATUS MESSAGES TO THE OPERATOR
/ECHO/NOECHO	ENABLES/DISABLES THE RETRANSMISSION OF THE DATA RECEIVED IN PASSIVE MODE. (IGNORED IN MODES OTHER THAN PASSIVE)
/MODEM/NOMODEM	ENABLES/DISABLES THE REPORTING OF MODEM STATUS INTERRUPT CHANGES. NOTE: THIS SWITCH CAUSES NO ACTION IN THIS DCLT PROGRAM BUT IT IS INCLUDED BECAUSE IT IS USED IN OTHER DCLT PROGRAMS.
/LOOP=LTYPE	SPECIFIES WHETHER MAINTENANCE LOOPBACK IS BEING USED. (IGNORED IN MODES OTHER THAN ACTIVE) MUST BE SPECIFIED EACH TIME ELSE NO LOOP IS USED.

```
                        "LTYPE" IS:
                         =INTERNALTTL

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

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


        2.5.4   DEFAULTS
                ---------

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

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

FOR THE RUN COMMAND THE DEFAULTS ARE:

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

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


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

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


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

2.5.5 PRINT

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

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

2.5.6 MISC COMMANDS

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

2.6 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

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

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

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

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

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

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

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

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

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

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

3.2 SPECIFIC ERROR MESSAGES

3.2.1 COMMAND LINE INTERPRETER ERRORS:

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

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

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

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

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

?TRANSMIT AND EXPECT LIST MUST BE IDENTICAL FOR LOOP?

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

3.2.2 DCLT ERROR MESSAGES:

BAD CLOCK - PROGRAM WILL HANG ON 'TIMEOUT''!!
THIS MEANS THAT EITHER NO CLOCK WAS ON THE SYSTEM OR THE ONE THAT WAS FOUND DID NOT INTERRUPT WHEN ASKED TO DO A ''TICK''.
THE PROGRAM WILL STILL RUN, BUT ANY OF THE PROGRAM THAT TIMES THE DEVICE WILL HANG IF THE DEVICE TIMES OUT.
ALSO, THE EVENT LOG WILL CONTAIN A ZERO EVENT TIME FOR ALL EVENTS LOGGED.

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

THIS MEANS THAT THE TRANSMIT OR EXPECT BUFFER IS FULL. NO MORE MESSAGES CAN BE ADDED TO THAT BUFFER.

BUFFER FULL - MSG. NOT BUILT !!
THIS MEANS THAT THE LAST MESSAGE YOU

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

CHAR. COUNT EXCEEDS BUFF LIMIT - MSG TRUNCATED

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

3.2.3 DEVICE ERROR MESSAGES

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

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

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

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

DATA COMPARISON DATA ERROR
TOTAL MISMATCHES IN MSG = NNN

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

DATA COMPARISON LENGTH ERROR
COMPARE COUNT= XXX RECEIVE COUNT= ZZZ

XXX= NUMBER OF BYTES IN THE COMPARE
MESSAGE
ZZZ= NUMBER OF BYTES IN THE RECEIVED
MESSAGE

THIS MEANS THAT THE MESSAGE RECEIVED
WAS A D'FFENT 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
              KMS11 WAS GIVEN IN HARDWARE P TABLE.

TIME OUT WAITING FOR OUTPUT INTERRUPT
          SEL0     SEL2
          XXXXXX   XXXXXX

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

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

INPUT INTERRUPT WHEN EXPECTING OUTPUT
SEL0 SEL2
XXXXXX XXXXXX

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

ILLEGAL OUTPUT INTERRUPT
SEL2 SEL6
XXXXXX XXXXXX

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

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

WHERE 'MMMMMM' 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

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

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

4.1 PRINTING OF EVENT LOG

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

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

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

THE TYPES OF EVENTS KEPT IN THE EVENT LOG ARE:

TRANSMIT MESSAGE QUEUED:
        EVENT TIME, ADDRESS OF 1ST BYTE OF MESSAGE,
        TOTAL NO. OF BYTES, MODEM STATUS AT THAT TIME.
TRANSMIT MESSAGE COMPLETED:
        EVENT TIME, ADDRESS OF 1ST BYTE OF MESSAGE,
        TOTAL NO. OF BYTES, MODEM STATUS AT THAT TIME.
RECEIVE SPACE QUEUED:
        EVENT TIME, ADDRESS OF 1ST BYTE OF MESSAGE,
        TOTAL NO. OF BYTES, MODEM STATUS AT THAT TIME.
RECEIVE MESSAGE COMPLETED:
        EVENT TIME, ADDRESS OF 1ST BYTE OF MESSAGE,
        TOTAL NO. OF BYTES, MODEM STATUS AT THAT TIME.
DATA COMPARISON STARTED:
        EVENT TIME, ADDRESS OF 1ST BYTE OF RECEIVED MSG.,
        TOTAL NO. OF BYTES IN RCV. MSG., TOTAL NO. OF BYTES
        IN EXPECT MSG.
DATA COMPARISON DATA ERROR:
        EVENT TIME, ADDRESS OF 1ST BYTE OF RECEIVED MSG.,
        TOTAL NO. OF BYTES IN RCV. MSG., TOTAL NO. OF

COMPARISON FAILURES
DATA COMPARISON LENGTH ERROR:
EVENT TIME, ADDRESS OF 1ST BYTE OF RECEIVED MSG., TOTAL NO. OF BYTES IN RCV. MSG., TOTAL NO. OF BYTES IN EXPECT MSG.
DEVICE INIT AND SETUP:
EVENT TIME, MODE OF OPERATION, TYPE OF MAINTENANCE LOOP, 'DCLT" PASS COUNT, 'RUN" PARAMETERS
DEVICE ERROR:
EVENT TIME, DEVICE ERROR MESSAGE, CONTENTS OF TWO REGISTERS RELATING TO THE ERROR.
END OF PASS:
^C ABORT:
EVENT TIME, 'DCLT" PASS COUNT, 'DCLT" ERROR COUNT, NO. OF 'NOBUFF'S"(NO. OF CONTROL-OUTS WITH THE NO-BUFFER SET SINCE THE LAST 'DCLT RUN" COMMAND.)

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

4.2 OPERATOR STATUS MESSAGES

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

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

4.3 PRINTING OF KMS11 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 PDP-11 MEMORY IS UPDATED BY THE KMS11 WHENEVER A FATAL ERROR OCCURS. THE ERROR COUNTER LOCATIONS OF THE BASE TABLE ARE UPDATED EVERY SECOND BY THE KMS11.

4.3.1 PRINTING ERROR LOCATIONS

TO PRINT ERROR COUNTER LOCATIONS, INPUT 'BASE/ERROR''.
LOCATIONS BASE+3..BASE+12 WILL BE DISPLAYED.
THE BASE ADDRESS IN THIS PROGRAM IS ALWAYS 17370.

EXAMPLE :

```
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''.
200 BYTES WILL BE DISPLAYED.

4.3.3 PRINTING SINGLE LOCATION

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

EXAMPLE :

```
RPT> (A) ? B/O=3

LOCATION        CONTENTS          DESCRIPTION
17373             006        NTLR - NAKS..RCVD NO BUFFERS

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   164100          ;[2] CSR ADDRESS
.WORD   400             ;[4] INTERRUPT VECTOR
.WORD   240             ;[6] INTERRUPT PRIORITY (5)
```

6.0 MODE AND MESSAGE DESCRIPTIONS

6.1 MODE DESCRIPTIONS

BECAUSE THE KMS11 SUPPORTS 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.

NOTE: KMS11 DEVICES CANNOT BE DOWN-LINE-LOADED.

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" MODE

REMEMBER THAT THE WHENEVER A "RUN" COMMAND IS TYPED, THE DEFAULT IS NO LOOPBACK AND THAT LOOP MODE MUST BE SPECIFIED BY A "/LOOP=INT" IF LOOP MODE IS DESIRED.
LOOP MODE IS VALID ONLY 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.

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 [illegible] THE RX BUFFER ON THE
ITEP SIDE IS ONLY 10 BYTES LARGER THAN THE TX BUFFER YOU
HAVE SELECTED, SO BE SURE TO SET THE TX BUFFER ON THE DCLT
NODE ACCORDINGLY.

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

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

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

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

7.2 TROUBLESHOOTING HINTS

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

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

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

7.2.1 INTERNAL LOOP AT EACH NODE

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

A POSSIBLE COMMMAND SEQUENCE IS:

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

WHAT THE ABOVE COMMAND SEQUENCE MEANS:

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

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

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

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

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

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

WHAT THIS SEQUENCE MEANS:

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

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

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

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

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

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

WHAT THIS SEQUENCE MEANS:

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

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

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

7.2.2 TRANSMIT ON ONE NODE RECEIVE ON THE OTHER

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

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

WHAT THIS SEQUENCE MEANS:

THE "C E " AND "C T" INITIALIZE BOTH THE TRANSMIT AND EXPECT LISTS. THE "SE T=1ALT/S=250" SETS THE TRANSMIT LIST ON NODE A TO BE 1 MESSAGE WITH A LENGTH OF 250 BYTES AND DATA OF ALTERNATING ONES AND ZEROS. THE "R M=TR/PAS=3" SETS THE RUN MODE OF NODE A TO BE TRANSMIT AND THE PASS COUNT IS SET TO 3. THE "R M=R/NOCH/PAS=3" SETS THE RUN MODE OF NODE B TO BE RECEIVE, NO DATA CHECKING IS TO BE DONE,AND THE PASS COUNT IS SET TO THREE.

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

FOR NODE A:

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

FOR NODE B:

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

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

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

WHAT THIS SEQUENCE MEANS:

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

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

NODE A: IS THE SAME AS ABOVE.

NODE B:

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

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

7.2.3 ONE NODE ACTIVE THE OTHER NODE PASSIVE

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

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

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

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

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

FOR NODE A:

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

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

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

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

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

7.2.4 BOTH NODES ACTIVE

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

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

WHAT THIS SEQUENCE MEANS:

NODE A SETS UP IS TRANSMIT LIST TO BE 3 MESSAGES. MESSAGE 1 IS 10 BYTES LONG AND CONTAINS DATA OF ALTENATING 0'S AND 1'S MESSAGE 2 IS 20 BYTES LONG AND CONTAINS DATA OF THE CCITT PATTERN. MESSAGE THREE IS 30 BYTES LONG AND CONTAINS ALPHANUMERICS FOR DATA. THE EXPECT LIST ALSO CONTAINS 3 MESSAGES. MESSAGE 1 IS 11 BYTES LONG AND CONTAINS 0'S FOR DATA. MESSAGE TWO IS 21 BYTES LONG AND CONTAINS 1'S FOR DATA. MESSAGE 3 IS 31 BYTES LONG AND CONTAINS THE ITEP DATA.

NODE B HAS THE SAME MESSAGES EXCEPT THAT THE
TRANSMIT MESSAGE LIST IS THE EXPECT MESSAGE LIST
AND VICE VERSA.
BOTH NODES ARE RUN IN THE ACTIVE MODE WITH
DATA CHECKING AND PASS COUNT EQUAL TO THREE.

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

```
MODE=ACTIVE/PASS=C0000
            /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 THF 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
                       X   X   X   X   X  X   X
>>> 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
                       X   X   X   X   X  X   X
                        .....
```

.....
.....

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

7.3.5 EXAMPLE EXIT COMMAND

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

```
EXIT
```

WILL RETURN YOU TO THE SUPERVISOR.

```
DR>
```

IF YOU ENTERED REPORT FROM THE DCLT LEVEL
THEN TYPING

```
EXIT
```

WILL RETURN YOU TO THE DCLT LEVEL.

```
DCLT>
```

7.4 THINGS TO WATCH OUT FOR

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

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

```
                                .SBTTL  PROGRAM HEADER

002000                                  BGNMOD

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

002000                                  POINTER BGNRPT,BGNAU,BGNDU

002000                                  HEADER  CZKMU,A,0,1800.,0,#PRI07
(4)  002000
(4)  002000     103                                                     LSNAME::
(4)  002001     132                                                             .ASCII /C/
(4)  002002     113                                                             .ASCII /Z/
(4)  002003     115                                                             .ASCII /K/
(4)  002004     125                                                             .ASCII /M/
(6)  002005     000                                                             .ASCII /U/
(6)  002006     000                                                             .BYTE   0
(5)  002007     000                                                             .BYTE   0
(5)  002010                                                                     .BYTE   0
(4)  002010     101                                                             .BYTE   0
(5)  002011                                                             LSREV::
(4)  002011     060                                                             .ASCII  /A/
(5)  002012                                                             LSDEPO::
(4)  002012  000000                                                             .ASCII  /0/
(5)  002014                                                             LSUNIT::
(4)  002014  003410                                                             .WORD   0
(5)  002016                                                             LSTIML::
(4)  002016  046266                                                             .WORD   1800.
(5)  002020                                                             LSHPCP::
(4)  002020  000000                                                             .WORD   LSHARD
(5)  002022                                                             LSSPCP::
(4)  002022  002130                                                             .WORD   0
(5)  002024                                                             LSHPTP::
(4)  002024  000000                                                             .WORD   LSHW
(5)  002026                                                             LSSPTP::
(4)  002026  046552                                                             .WORD   0
(5)  002030                                                             LSLADP::
(4)  002030  000000                                                             .WORD   LSLAST
(5)  002032                                                             LSSTA::
(4)  002032  000000                                                             .WORD   0
(5)  002034                                                             LSCO::
(4)  002034  000000                                                             .WORD   0
(5)  002036                                                             LSDTYP::
(4)  002036  000000                                                             .WORD   0
(5)  002040                                                             LSAPT::
(4)  002040  002124                                                             .WORD   0
                                                                        LSDTP::
                                                                                .WORD   LSDISPATCH
```

```
   (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  012230
   (5)  002062
   (4)  002062  030242
   (5)  002064
   (4)  002064  000000
   (5)  002066
   (4)  002066  000000
   (5)  002070
   (4)  002070  035562
   (5)  002072
   (4)  002072  035554
   (5)  002074
   (4)  002074  000000
   (5)  002076
   (4)  002076  012252
   (5)  002100
   (4)  002100  104035
   (5)  002102
   (4)  002102  000000
   (5)  002104
   (4)  002104  030256
   (5)  002106
   (4)  002106  035470
   (5)  002110
   (4)  002110  035466
   (5)  002112
   (4)  002112  030250
   (5)  002114
   (4)  002114  000000
   (5)  002116
   (4)  002116  000000
   (5)  002120
   (4)  002120  000000
4228
4235
```

```
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
002122                                                                  .WORD   1
(4)  002122  000001
                                                                LSDISPATCH::
(3)  002124                                                             .WORD   T1
(6)  002124  035570
```

```
                                .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  164100                         .WORD   164100          ;[2] CSR ADDRESS
002134  000400                         .WORD   400             ;[4] INTERRUPT VECTOR
002136  000240                         .WORD   240             ;[6] INTERRUPT PRIORITY (5)
002140  000000                         .WORD   0               ;[10] SPARE
002142  000000                         .WORD   0               ;[12] SPARE
002144  000000                         .WORD   0               ;[14] SPARE
002146  000000                         .WORD   0               ;[16] SPARE


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

```
(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
4397
```

```
4399                                    ;***** INDEPENDENT EQUATES
4400
4401            001000                          BUFLIM=512.             ;MAX BUFFER SIZE IN BYTES
4402                                            ;  APPLIES TO TX,RX AND CMP BUFFS
4403            000017                          MSGLIM=15.              ;MAX NO. OF MESSAGES PER BUFFER
4404                                            ;  (FOR EACH INCREMENT (+1) TO MSGLIM,
4405                                            ;   ADD 6 WORDS TO THE POINTER TABLE
4406                                            ;   (PTRTAB:) SINCE THIS MEANS 2 MORE
4407                                            ;   ''POINTER'' WORDS PER BUFFER.
4408
4409                                    ;MODE OF OPERATION EQUATES
4410            000000                          REC=0                   ;RECEIVE MODE
4411            000001                          TRA=1                   ;TRANSMIT MODE
4412            000002                          PAS=2                   ;PASSIVE MODE
4413            000003                          ACT=3                   ;ACTIVE MODE
4414            000004                          DOW=4                   ;DOWN-LINE-LOAD MODE
4415            000005                          TAL=5                   ;TALK MODE
4416            000006                          LIS=6                   ;LISTEN MODE
4417                                    ;MAINT LOOP TYPE EQUATES
4418            000000                          NONE=   0               ;NO LOOP
4419            000001                          TTL=    1               ;INTERNAL TTL
4420            000002                          CABLE=  2               ;CABLE LOOP
4421            000003                          MODLOC= 3               ;MODMEM LOCAL
4422            000004                          MODREM= 4               ;MODEM REMOTE
4423            000005                          MOP=    5               ;MOP
4424
4425
4426                                    ;CLOCK ENABLE VALUES TO BE LOADED IN CLK'S CSR
4427            000100                          LCLKEN= 100             ;L-CLOCK CSR VALUE TO ENABLE THE CLOCK
4428            000111                          PCLKEN= 111             ;P-CLOCK CSR VALUE TO ENABLE THE CLOCK
4429            001600                          PCLKCT= 1600            ;P-CLOCK COUNT SET REGISTER FOR COUNTER
4430
4431                                    ;PARAM WORD EQUATES
4432
4433            000001                          STATB=  BIT0            ;OPERATOR AWAKE ASKED FOR
4434            000002                          DATCKB= BIT1            ;DATA CHECK BIT
4435            000004                          ECHOB=  BIT2            ;ECHO BIT
4436            000010                          MOCHK=  BIT3            ;MODEM CHECK/NO CHECK
4437            000020                          CRCB=   BIT4            ;CRC CALCUALTE ASKED FOR
4438            000040                          PROTOB= BIT5            ;PROTOCOL PROCESSING ASKED FOR
4439
4440
4441                                    ;EVENT LOG MESSAGE TYPES (USED TO LOCATE EVENT DESCRIPTION IN EVENT TABLE
4442                                    ;  AND DISPATCHING TO SEPERATE SECTIONS OF THE EVENT REPORTING SECTION)
4443            000000                          TXQ=    0               ;TRANSMIT MESSAGE QUEUED
4444            000002                          TXC=    2               ;TRANSMIT COMPLETE
4445            000004                          RXQ=    4               ;RECEIVE BUFFER QUEUED
4446            000006                          RXC=    6               ;RECEIVE COMPLETE
4447            000010                          DER=    10              ;DEVICE INFORMATION
4448            000012                          DVI=    12              ;DEVICE ABOUT TO INIT
4449            000014                          DCK=    14              ;DATA COMPARISON RESULTS
4450
4451            000020                          DLE=    20              ;DATA COMPARISON LENGH ERROR
4452            000022                          DDE=    22              ;DATA COMPARISON DATA ERROR
4453            000024                          EOP=    24              ;END OF PASS
4454            000026                          ABO=    26              ;^C ABORT
```

```
                                ;;;;EQUATES FOR FLAG WORD;;;;;;

000001                                  ININT= 1                ;INPUT INT. REC.
000002                                  OTINT= 2                ;OUTPUT INT REC
000004                                  QRX=   4                ;RX QUED /COMPL
000010                                  QTX=    10              ;TX QUED/COMPL
000020                                  CTX=    20              ;TX COMPL AND IN TXSEL4 AND TSEL6
000040                                  CRX=    40              ;RX COMPL AND IN TSEL4 AND TSEL6
000100                                  ERX=    100             ;EXPECT TO GET A RX COMPLED
000200                                  ETX=    200             ;EXPECT TO GET A TX COMPLETED
000400                                  DLLGA=  400             ;DOWN LINE LOAD GO AHEAD BIT
002000                                  BTUP=   2000            ;BASE TABLE UPDATE REQUESTED

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

                                ; DEFS FOR COMMAND LINE INTERPRETATION ACTION VALUES
000000                                  NULL=0
000001                                  CLEAR=1
000002                                  SHOW=2
000003                                  CHECK=3
000004                                  RUN=4
000005                                  HLP=5
000006                                  CSHEXP=6
000007                                  CSHTRN=7
000010                                  SETEXP=10
000011                                  SETTRN=11
000012                                  SIZE=12
000013                                  QCOPY=13
000014                                  NUM=14
000015                                  OPRMSG=15
000016                                  STATUS=16
000017                                  ENDQQ=17
000020                                  CMSG0=20
000021                                  CMSG1=21
000022                                  CMSG2=22
000023                                  CMSG3=23
000024                                  CMSG4=24
000025                                  CMSG5=25
000026                                  CMSG6=26
000027                                  ATVMOD:27
000030                                  PASMOD=30
000031                                  RECMOD=31
000032                                  LISMOD=32
```

```
4511    000033                  DLLMOD=33
4512    000034                  TRAMOD=34
4513    000035                  TALMOD=35
4514    000036                  NO=36
4515    000037                  ECHO=37
4516    000040                  CRC=40
4517    000041                  PROTO=41
4518    000042                  PASC=42
4519    000043                  MOP=43
4520    000044                  TTLLOP=44
4521    000045                  CBLLOP=45
4522    000046                  LMDLOP=46
4523    000047                  RMDLOP=47
4524    000050                  NOTNUF=50
4525    000051                  BADCHR=51
4526    000052                  DMPS=52
4527    000053                  DMPE=53
4528    000054                  DMPQ=54
4529    000055                  PRNT=55
4530    000056                  MOSC=56                 ;MODEM/NOMODEM
4531    000057                  EXIT=57                 ;EXIT COMMAND
4532    000060                  SETET=60                ;S E=T COMMAND
4533                    ;FOLLOWING EQUATES USED IN REPORT CLI
4534    000001                  RPHLP=1                 ;HELP COMMAND
4535    000002                  RPEXT=2                 ;EXIT COMMAND
4536    000003                  RPLOG=3                 ;PRINT EVENT LOG COMMAND
4537    000004                  RPSWE=4                 ;BASE/ERROR COMMAND
4538    000005                  RPSWF=5                 ;BASE/FULL COMMAND
4539    000006                  RPSWO=6                 ;BASE/OFFSET
4540    000007                  RNOTNF=7                ;MORE COMMAND NEEDED
4552
4553                    ;***** DEVICE DEPENDENT EQUATES
4554                    ; MODEM SIGNAL BIT DEFINITONS
4555                    ;       IF SIGNAL AVAILABLE IN DEVICE, EQUATE NAME TO BIT POSITION.
4556                    ;       ELSE EQUATE IT TO = 0
4557
4558
4559    000004                  CTS=    BIT2            ;CLEAR TO SEND (CIRCUIT CB)
4560    000010                  DSR=    BIT3            ;DATA SET READY (CIRCUIT CC)
4561    000001                  DCD=    BIT0            ;DATA CARRIER DETECT (CIRCUIT CF)
4562    000040                  RTS=    BIT5            ;REQUEST TO SEND (CIRCUIT CA)
4563    000200                  RI=     BIT7            ;RING INDICATOR (CIRCUIT CE)
4564    040000                  SQD=    BIT14           ;SIGNAL QUALITY DETECT (CIRCUIT CG)
4565    001000                  TM=     BIT9            ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)
4566
4567
4575
4576
4577                    ; DEVICE SIGNALS
4578    100000                  KRUN=   BIT15           ;RUN BIT
4579    040000                  MCLR=   BIT14           ;MASTER CLEAR
4580    004000                  LULOOP= BIT11           ;LINE UNIT LOOP(TTL)
4581    002000                  RAMO=   BIT10           ;LOAD/VERIFY CRAM (KMS11)
4582    002000                  HALFDB= BIT10           ;HALF DUPLEX BIT
4583    000400                  MAINTB= BIT8            ;MAINT MODE BIT
4584    000200                  RDI=    BIT7            ;READY IN
```

```
4585            000200                          RDO=    BIT7
4586            000040                          RQI=    BIT5            ;REQUEST IN
4587            000004                          RXBIT=  BIT2            ;RX BIT
4588            000100                          IEO=    BIT6            ;ENABLE OUTPUT INTERRUPT BIT
4589            000001                          BACC=   BIT0            ;BUFFER ADDR. CHAR COUNT
4590
```

```
                                .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
4642            002324                          EMSG4:
4643            002324                          MSG5:                   ;''INTERPROCESSOR TEST PROGRAM'S (ITEP)'' MESSAGE
4644                                                                    ; #1, (DP1:)
4645            002324  077577  040444  052040          .ASCII  <177><177>/$A THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG./
                002332  042510  050440  044525
                002340  045503  041040  047522
                002346  047127  043040  054117
                002354  045040  046525  042520
                002362  020104  053117  051105
                002370  052040  042510  046040
                002376  055101  020131  047504
                002404  027107
4646            002406  005015  077401  077577          .ASCIZ  <15><12><001><177><177><177><177>
                002414  000177
4647            002416                          EMSG5:
4648            002416                          MSG6:                   ;ALPHA-NUMERICS (OR FUTURE COMM TURNAROUND MSG)
4649            002416  022043  021041  023040          .ASCII  /#$!'' &'()*+,-.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
4650            002507     057  056133  057135          .ASCIZ  ?/[\]^_%?
                002514  022537     000
4651            002517                          EMSG6:
4652                    002520                  .EVEN
4653
4654                                            ; ************************************************************************
4655                                            ;THESE THREE STORAGE AREAS MUST NOT BE SEPERATED !!!!
4656
4657            002520  047045  040445          OPBFPT: .ASCII  /%N%A/
4658            002524  000122                  OPBUF:  .BLKB   82.             ;BUFFER FOR OPERATOR SPEC'D MESSAGES
4659            002646                          OPEND:
4660
4661                                            ; THE ABOVE THREE LINES MUST BE KEPT TOGETHER
4662                                            ; ************************************************************************
4663
4664            002646      033                 MSG8:   .BYTE   33              ;RECEIVE BUFFER FILL PATTERN
4665            002647                          EMSG8:
4666
4667                                            ; DOWN-LINE-LOAD MESSAGE DEFINITIONS
4668                                                    ;;;;ENTER MOP MODE MESSAGE FORMAT
4669                                                    ;;;;THE NODE WILL ENTER MAINTENANCE MODE ONLY IF THE PASSWORD MATCHES.
4670            002647      006                 DLLM1:  .BYTE   6       ;BINARY CODE FOR MAINTENANCE MODE
4671            002650      000                 PASS1:  .BYTE   0       ;PASSWORD BYTE #1   LEGAL VALUE 0 - 255
4672            002651      000                 PASS2:  .BYTE   0       ;;VALUE IN BYTE 1 IS DUPLICATED HERE
4673            002652      000                 PASS3:  .BYTE   0       ;;;AND HERE
4674            002653      000                 PASS4:  .BYTE   0       ;;;;AND HERE.
4675            002654                          DLLM1E:                 ;END ENTER MOP MODE MESSAGE FORMAT
4676                                                    ;;;;MEMORY LOAD WITH TRANSFER ADDRESS MESSAGE FORMAT
```

```
 4677  002654  000                    DLLM2:  .BYTE   0       ;CODE
 4678  002655  000                            .BYTE   0       ;LOAD NUMBER
 4679  002656  006                            .BYTE   6       ;LOAD ADDRESS LSB
 4680  002657  000                            .BYTE   0
 4681  002660  000                            .BYTE   0
 4682  002661  000                            .BYTE   0       ;LOAD ADDRESS
 4683                                    ;
 4684                                    ;IMAGE DATA
 4685                                    ;
 4686  002662  005037  000006                   CLR     @#6
 4687  002666  012706  001000                   MOV     #1000,SP
 4688  002672  012701  177560                   MOV     #177560,R1      ;SET UP TTY
 4689  002676  010700                           MOV     PC,R0           ;MAKE ADDR.PIC
 4690  002700  062700  000034                   ADD     #<MSG-.>,R0     ;ADDRESS MSG.
 4691  002704  105761  000004           1$:     TSTB    4(R1)           ;TTY READY?
 4692  002710  100375                           BPL     1$              ;WAIT TIL YES
 4693  002712  112061  000006                   MOVB    (R0)+,6(R1)     ;TYPE A CHAR
 4694  002716  001372                           BNE     1$              ;KEEP GOING
 4695  002720  012737  000026  000024           MOV     #26,@#24        ;SET UP POWER FAIL
 4696  002726  005037  000026                   CLR     @#26            ;MAKE SURE T BIT CLAER
 4697  002732  000777                           BR      .               ;JUMP ON YOURSELF
 4698  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
 4699  003023     012  027015  027056          .ASCIZ  <12><15>/....RELOAD PROGRAM..../
       003030  051056  046105  040517
       003036  020104  051120  043517
       003044  040522  027115  027056
       003052  000056
 4700  003054     006                          .BYTE   6               ;NEXT FOUR BYTES CONTAINS TRANSFER ADDRESS
 4701  003055     000                          .BYTE   0               ;;OF PROGRAM JUST DOWNLINE LOADED.
 4702  003056     000                          .BYTE   0               ;;;THIS PROGRAM STARTS AT ADDRESS 6.
 4703  003057     000                          .BYTE   0
 4704  003060                          DLLM2E:                 ;END MEMORY LOAD MESSAGE FORMAT
 4705
 4706                                          .EVEN
 4707
```

```
 4709                                   ;COMMAND LINE BUFFER, DATA LOCATIONS AND MESSAGES FOR ACTION ROUTINES
 4710
 4711  003060  000122                   CMDBUF: .BLKB   82.              ;BUFFER FOR OPERATOR COMMANDS
 4712  003202  000000                   KEYWD1: .WORD   0                ;THIS LOC WILL =1 IF CLEAR TYPED, 2 FOR SHOW,
 4713                                                                    ; A 4 IF RUN WAS TYPED, 5 IF HELP WAS TYPED
 4714  003204  000000                   QUALFG: .WORD   0                ;THIS LOC HOLDS QUALIFIER VALUE (SIZE OR COPY)
 4715  003206  000000                   QUALVL: .WORD   0
 4716  003210  013044                   HLPTAB: .WORD   HLP1
 4717  003212  013057                           .WORD   HLP2
 4718  003214  013174                           .WORD   HLP3
 4719  003216  013261                           .WORD   HLP3A
 4720  003220  013306                           .WORD   HLP4
 4721  003222  013365                           .WORD   HLP4A
 4722  003224  013443                           .WORD   HLP5
 4723  003226  013533                           .WORD   HLP6
 4724  003230                           HLPEND:
 4725                                   ;INDEX TABLE FOR REPORT 'RPT>' HELP MESSAGES
 4726  003230  013671                   RHLPTB: .WORD   RHLP1
 4727  003232  013714                           .WORD   RHLP2
 4728  003234  013747                           .WORD   RHLP3
 4729  003236  014000                           .WORD   PHLP4
 4730  003240  014032                           .WORD   RHLP5
 4731  003242  014071                           .WORD   RHLP6
 4732  003244  014130                           .WORD   RHLP7
 4733  003246  000000                   RHLPEN: .WORD   0                ;END OF REPORT HELP TABLE
 4734
 4735                                   ;INDEX  TABLE FOR KMS11 BASE TABLE DATA DESCRIPTION MESSAGES
 4736  003250  020322                   DMCIND: .WORD   DMUNKN
 4737  003252  020322                           .WORD   DMUNKN
 4738  003254  020342                           .WORD   DMC002
 4739  003256  020363                           .WORD   DMC003
 4740  003260  020420                           .WORD   DMC004
 4741  003262  020461                           .WORD   DMC005
 4742  003264  020514                           .WORD   DMC006
 4743  003266  020551                           .WORD   DMC007
 4744  003270  020606                           .WORD   DMC010
 4745  003272  020641                           .WORD   DMC011
 4746  003274  020663                           .WORD   DMC012
 4747  003276  020705                           .WORD   DMC013
 4748  003300  020744                   DMCEND: .WORD   DMC377           ;NO KMS11 MESSAGES MUST FOLLOW DMCEND
 4749
 4750  003302  014276  014305  014312  SHTYTB: .WORD   SHTYP0,SHTYP1,SHTYP2,SHTYP3,SHTYP4,SHTYP5,SHTYP6,SHTYP7
       003310  014317  014324  014532
       003316  014337  014345
 4751
 4752                                   ; THE LIST OF BYTES BELOW ARE THE FIRST BYTES OF THE PREDEFINED MESSAGES
 4753                                   ;  USED TO "SHOW" THE TRANSMIT AND COMPARE BUFFER CONTENTS.
 4754
 4755  003322     000     377     252  SHTAB:  .BYTE   0,377,252,125,203,177,043
       003325     125     203     177
       003330     043
 4756  003331                           SHTEND:
 4757          003332                           .EVEN
 4758
 4759  003332  014356                   MODES:  .WORD   M00      ;ADDRESSES OF MODE TYPES IN ASCII
 4760  003334  014366                           .WORD   M01
```

```
 4761  003336  014377                          .WORD   MO2
 4762  003340  014407                          .WORD   MO3
 4763  003342  014416                          .WORD   MO4
 4764  003344  014433                          .WORD   MO5
 4765  003346  014440                          .WORD   MO6
 4766
 4767  003350  014447                LOOPS:    .WORD   LP0     ;ADDRESSES OF LOOP TYPES IN ASCII
 4768  003352  014457                          .WORD   LP1
 4769  003354  014470                          .WORD   LP2
 4770  003356  014476                          .WORD   LP3
 4771  003360  014511                          .WORD   LP4
 4772
 4773                                ;COMMAND LINE TRAVERSE LOCATIONS (USED BY 'P$TRV')
 4774
 4775  003362  000000                P$BUFA: .WORD   0               ;LOC. TO HOLD ADDR. OF CMD LINE BUFFER
 4776  003364  000000                P$TREE: .WORD   0               ;LOC. TO HOLD ADDR. OF PARSING TREE
 4777  003366  000000                P$ACT:  .WORD   0               ;LOC. TO HOLD ADDR. OF ACTION ROUTINE
 4778  003370  000000                P$CNT:  .WORD   0               ;LOC. TO BE A COUNTER LOCATION
 4779  003372  000000                P$NUM:  .WORD   0               ;LOC. TO HOLD NUMERIC VALUE FROM PARSE
 4780  003374  000000                P$RADX: .WORD   0               ;LOC. TO HOLD RADIX USED(LO) AND +/-(HI BYTE)
 4781  003376     000                P$NNUF: .BYTE   0               ;RETURN =0 IF ENOUGH OF COMMAND FOUND
 4782  003377     000                P$GDBD: .BYTE   0               ;RETURN CODE 0 IF NO ERROR FOUND
 4783
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 23-18
CZKMUA.P11    30-MAR-82 09:13                 MESSAGE BUFFERS AND POINTER TABLES

 4785                                 .SBTTL          MESSAGE BUFFERS AND POINTER TABLES
 4786
 4787  003400  001000                 TXBUF:  .BLKB   BUFLIM  ;TRANSMITTER BUFFERS
 4788  004400  001000                 RXBUF:  .BLKB   BUFLIM  ;RECEIVER BUFFERS
 4789  005400  001000                 CMPBUF: .BLKB   BUFLIM  ;COMPARISON BUFFERS
 4790  006400  000264                 PTRTAB: .BLKW   180.    ;TABLE FOR MESSAGE ADDRS. & BYTE COUNTS
 4791  007150                         PTREND:                 ; END OF MSG. PTR. TABLE
 4792
 4793  007150  000000                 RXPTR:  .WORD   0       ;RECEIVER MESSAGE POINTER
 4794  007152  000000                 TXPTR:  .WORD   0       ;TRANMITTER BUFFER POINTER
 4795  007154  000000                 CMPPTR: .WORD   0       ;COMPARISON BUFFER POINTER
 4796  007156  000000                 CMPTOT: .WORD   0       ;CMP MSG TOTAL
 4797  007160  000000                 CTOTCC: .WORD   0       ;COMPARE BUFFER CHAR. COUNT
 4798  007162  000000                 CCURAD: .WORD   0       ;CURRENT ADDR OF CMP BUFF TO ADD AT
 4799
 4800  007164  000000                 DVTXA:  .WORD   0       ;DEVICE TX ADDR
 4801  007166  000000                 DVTCC:  .WORD   0       ;DEVICE TX CHAR COUNT
 4802  007170  000000                 DVTCT:  .WORD   0       ;DEVICE TX MESSAGE COUNT
 4803  007172  000000                 TXMTOT: .WORD   0       ;TX MSG TOTAL
 4804  007174  000000                 TTOTCC: .WORD   0       ;TX BUFFER CHAR. COUNT
 4805  007176  000000                 TCURAD: .WORD   0       ;CURRENT ADDR. OF TX BUFF TO ADD AT
 4806
 4807  007200  000000                 DVRXA:  .WORD   0       ;DEVICE RX ADDR
 4808  007202  000000                 DVRCC:  .WORD   0       ;DEVICE RX CHAR COUNT
 4809  007204  000000                 DVRCT:  .WORD   0       ;DEVICE RX MESSAGE COUNT
 4810  007206  000000                 RXMTOT: .WORD   0       ;RX MSG TOTAL
 4811
 4812  007210  000000                 LNCNT:  .WORD   0       ;NUMBER OF OPERATOR AWAKE MSGS
 4813  007212  000000                 NOBUF:  .WORD   0       ;NUMBER OF NO BUFFS
 4814  007214  000000                 PSCNT:  .WORD   0       ;PASS COUNTER
 4815  007216  000000                 ERRCNT: .WORD   0       ;ERROR COUNTER
 4816  007220  000000                 STADD:  .WORD   0       ;START ADDR.
 4817  007222  000000                 ENADD:  .WORD   0       ;END ADDR. FOR DUMP
 4818  007224  000000                 BYTBIT: .WORD   0       ;BYTE BIT FOR DUMP ROUTINE
 4819
 4820                                 ;OTHER MESSSAGE RELATED STORAGE LOCATIONS
 4821  007226  000000                 MSGTYP: .WORD   0       ;TYPE OF DATA 0=0'S,1=1'S,2=10'S,3=01'S
 4822                                                         ;4=CCITT,5=QUICK FOX,6=ALPHA/NUM,7=OPER
 4823  007230  000000                 CURCC:  .WORD   0       ;TX/RX/CMP CHAR COUNT
 4824  007232  000000                 CPTRR:  .WORD   0       ;CURRENT RX POINTER
 4825  007234  000000                 CPTR:   .WORD   0       ;CURRENT POINTER
 4826  007236  000000                 CURADD: .WORD   0       ;CURRENT TX/RX/CMP START ADDD
 4827  007240  000000                 TOTCC:  .WORD   0       ;TOTAL CHAR COUNT NOT MORE THEN 'BUFLIM'
 4828  007242  000000                 OFSET:  .WORD   0       ;OFFSET COUNT
 4829  007244  000000                 TEMP:   .WORD   0       ;TEMPORARY LOCATIONS (USED A LOT)
 4830  007246  000000                 TEMP1:  .WORD   0
 4831  007250  000000                 TEMP2:  .WORD   0
 4832  007252  000000                 TEMP3:  .WORD   0
 4833  007254  000000                 TEMP4:  .WORD   0
 4834  007256  000000                 TEMP5:  .WORD   0
 4835  007260  000000                 CONOTM: .WORD   0       ;CONTROL OUT ERROR MSG. ADDRESS
 4836  007262  000000                 CONTIN: .WORD   0       ;WORD FOR CONTORL IN
 4837  007264     000                 GOOD:   .BYTE   0       ;BYTE TO HOLD EXPECTED MESSAGE DATA BYTE FOR ERR REPORT
 4838  007265     000                 BAD:    .BYTE   0       ;BYTE TO HOLD RECEIVED MESSAGE DATA BYTE FOR ERR REPORT
 4839  007266  000000                 INDEX:  .WORD   0       ;WILL CONTAIN POINTER TO KMS11 MESSAGES
 4840  007270  000000                 INDEXE: .WORD   0       ;WILL CONTAIN POINTER TO LAST OF KMS11 MESSAGES
```

```
4841  007272  000000                  BEND:   .WORD   0       ;LAST LOCATION IN BASE TABLE TO BE PRINTED
4842
4843
4844
4845                                  ;MORE INDEPENDENT CODE STORAGE LOCATIONS
4846  007274  000000                  BDATA:  .WORD   0       ;POINTER TO BASE TABLE
4847  007276  000000                  LOGUNT: .WORD   0       ;LOC. TO HOLD LOGICAL UNIT NUMBER
4848  007300  000000                  PCADD:  .WORD   0       ;LOC. HOLD PC OF CALLIN ROUTINE
4849  007302  000000                  DCLFLG: .WORD   0       ;CLEANUP AND EXIT FLAG. 1=DO CLEANUP ROUTINE&EXIT
4850  007304  000000                  RESFLG: .WORD   0       ;LOC TO HOLD FLAG (-1) THAT A RESTART WAS GIVEN
4851  007306  000000                  MODTYP: .WORD   0       ;DCLT MODE OF OPERATION TYPE
4852                                                          ;  (0=REC-ONLY, 1=TX-ONLY, 2=PASSIVE-LOOPBK,
4853                                                          ;   3=ACTIVE-LOOPBK, 4=DOWN L.L., 5=TALK, 6=LISTEN)
4854  007310  000000                  MLTYP:  .WORD   0       ;MAINTENANCE LOOP TYPE (0=NONE, 1=INTERNAL TTL,
4855                                                          ;  2=CABLE, 3=MODEM-ANALOG LOOPBK (LOCAL),
4856                                                          ;  4=MODEM-DIGITAL LOOPBK (REMOTE), 5=MOP)
4857  007312  000000                  FHDPLX: .WORD   0       ;FULL OR HALF DUPLEX FLAG (1=FULL 0=HAL. DUPLEX)
4858  007314  000002                  PARAM:  .WORD   2       ;PROGRAM PARAMETERS
4859                                                          ; BIT0= STATUS MSGS TO OPR PRINTED (1=YES)
4860                                                          ; BIT1= DATA CHECKING DONE ON RCVD MSGS (1=YES)
4861                                                          ; BIT2= ECHO (TRANSMIT) RCV'D MSG.(PASSIVE)(1=YES)
4862                                                          ; BIT3= SPARE
4863                                                          ; BIT4= CRC CALC./CHECK DONE (1=YES)
4864                                                          ; BIT5= PROTOCOL EMULATION (1=YES)
4865                                                          ; BIT6= SPARE
4866  007316  000000                  RPASS:  .WORD   0       ;PASS NUMBER FROM RUN COMMAND
4867  007320  000000                  FLAG:   .WORD   0       ;DEVICE FLAG WORD
4868
4869                                  ;MODE DISPATCH TABLE
4870  007322  041452                  MODE:   .WORD   RXONLY  ;RX ONLY DISPATCH
4871  007324  041504                          .WORD   TXONLY  ;TX ONLY DISPATCH
4872  007326  041544                          .WORD   PLCK    ;PASSIVE LOOP BACK DISP
4873  007330  041600                          .WORD   ALCK    ;ACTIVE LOOP BACK DISP
4874  007332  042724                          .WORD   DLL     ;DOWN LINE LOAD DISP
4875  007334  043544                          .WORD   TALCK   ;TALK MODE DISPATCH
4876  007336  043764                          .WORD   LISCK   ;LISTEN MODE DISPATCH
4877
4878
4879                                  .SBTTL          CLOCK TABLES, EVENT LOG AND POINTERS
4880  007340  000000                  CLKCSR: .WORD   0       ;CLOCK CSR ADDRESS
4881  007342  000000                  CLKBR:  .WORD   0       ;CLOCK INTERRUPT LEVEL
4882  007344  000000                  CLKVEC: .WORD   0       ;CLOCK INTERRUPT VECTOR
4883  007346  000074                  CLKHZ:  .WORD   60.     ;CLOCK'S HERTZ RATE
4884  007350  000000                  CLKEN:  .WORD   0       ;CLOCK'S CSR VALUE TO INTRPT. ENABLE IT
4885
4886  007352  000000                  TIMMIN: .WORD   0       ;PLACE TO KEEP TIME-SINCE-START
4887  007354  000000                  TIMSEC: .WORD   0
4888  007356  000000                  TIMTCK: .WORD   0       ;PLACE TO KEEP # OF TICKS/SEC
4889
4890  007360  000000                  TIMER1: .WORD   0       ;EVENT TIMER #1 (TICKS)
4891  007362  000000                  TIMER2: .WORD   0       ;EVENT TIMER #2 (TICKS)
4892  007364  000000                  TIMERS: .WORD   0       ;EVENT TIMER #3 (SECONDS)
4893
```

```
 4895
 4896  007366  007370                    ;EVENT LOG TABLE AND ITS NEXT ENTRY POINTER
 4897  007370  000341                    EVTPTR: .WORD   EVTLOG  ;POINTER TO NEXT FREE SPACE IN EVENT LOG
 4898  010272  000001                    EVTLOG: .BLKW   225.    ;EVENT LOG BUFFER
 4899                                    EVTEND: .BLKW   1.      ;APPROXIMATE END OF EVENT TABLE (ALLOWS CIRCULAR QUE)
 4900
 4901                                    .SBTTL          MODEM DATA SECTION
 4902  010274  000000                    MODS:   .WORD   0       ;MODEM STATUS
 4903
 4904                                    ;TABLE OF MODEM SIGNAL BIT DEFINITIONS
 4905
 4906  010276  000004                    MOBITS: .WORD   CTS             ;CLEAR TO SEND (CIRCUIT CB)
 4907  010300  000010                            .WORD   DSR             ;DATA SET READY (CIRCUIT CC)
 4908  010302  000001                            .WORD   DCD             ;DATA CARRIER DETECT (CIRCUIT CF)
 4909  010304  000040                            .WORD   RTS             ;REQUEST TO SEND (CIRCUIT CA)
 4910  010306  000200                            .WORD   RI              ;RING INDICATOR (CIRCUIT CE)
 4911  010310  040000                            .WORD   SQD             ;SIGNAL QUALITY DETECT (CIRCUIT CG)
 4912  010312  001000                            .WORD   TM              ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)
 4913  010314                            MOBITE:
 4914
 4915                                    ;TABLE OF ADDRESSES OF MODEM SIGNAL MESSAGE POSITIONS
 4916
 4917  010314  017063                    MOMSGS: .WORD   EVMCTS           ;CLEAR TO SEND (CIRCUIT CB)
 4918  010316  017067                            .WORD   EVMDSR           ;DATA SET READY (CIRCUIT CC)
 4919  010320  017073                            .WORD   EVMDCD           ;DATA CARRIER DETECT (CIRCUIT CF)
 4920  010322  017077                            .WORD   EVMRTS           ;REQUEST TO SEND (CIRCUIT CA)
 4921  010324  017103                            .WORD   EVMRI            ;RING INDICATOR (CIRCUIT CE)
 4922  010326  017107                            .WORD   EVMSQD           ;SIGNAL QUALITY DETECT (CIRCUIT CG)
 4923  010330  017113                            .WORD   EVMTM            ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)
 4924
 4925                                    ;TABLE OF ADDRESSES OF EVENT DESCRIPTION MESSAGES
 4926                                    ;       ORDER CORRESPONDS TO MESSAGE TYPE VALUES
 4927
 4928  010332  015437                    EVTLST: .WORD   EDTXQ   ;TRANSMIT MESSAGE QUEUED
 4929  010334  015463                            .WORD   EDTXC   ;TRANSMIT OF MESSAGE COMPLETE
 4930  010336  015512                            .WORD   EDRXQ   ;RECEIVE MESSAGE SPACE QUEUED
 4931  010340  015537                            .WORD   EDRXC   ;MESSAGE RECEIVED - RECEIVE COMPLETE
 4932  010342  015565                            .WORD   EDDER   ;DEVICE INFORMATION
 4933  010344  015632                            .WORD   EDDVI   ;DEVICE INITIALIZE STARTED
 4934  010346  015602                            .WORD   EDDCK   ;DATA COMPARISON DONE
 4935  010350  014447                            .WORD   LPO     ;NULL STRING
 4936  010352  015660                            .WORD   EDDLE   ;DATA COMPARE LENGTH ERROR
 4937  010354  015715                            .WORD   EDDDE   ;DATA COMPARE DATA ERROR
 4938  010356  015750                            .WORD   EDEOP   ;END OF PASS
 4939  010360  016021                            .WORD   EDABO   ;^C ABORT
 4940
 4941                                    ;;;FOLLOWING TABLE USED IN DOWNLINE LOAD ROUTINE.
 4942                                    ;;;;CONTAINS POINTERS TO ASCIZ DEVICE DESCRIPTIONS
 4943  010362  020217                    DLLIND: .WORD   DPM
 4944  010364  020222                            .WORD   DUM
 4945  010366  020225                            .WORD   DLM
 4946  010370  020230                            .WORD   DQM
 4947  010372  020233                            .WORD   DAM
 4948  010374  020236                            .WORD   DUPM
 4949  010376  020242                            .WORD   DMCM
 4950  010400  020246                            .WORD   DNM
```

```
4951  010402  020251                          .WORD   DLVM
4952  010404  020255                          .WORD   DMPM
4953  010406  020261                          .WORD   DTEM
4954  010410  020265                          .WORD   DVM
4955  010412  020270                          .WORD   DZM
4956  010414  020273                          .WORD   UNKM
4957  010416  020303                          .WORD   KDPM
4958  010420  020307                          .WORD   KDZM
4959  010422  020313                          .WORD   KLM
4960  010424  020316                          .WORD   DMVM
4961
4962                                  ;LOCATIONS USED DURING EVENT REPORTING
4963
4964  010426  000000                  EVTSEC: .WORD   0       ;TEMPORARY LOCS TO KEEP EVENT TIME WHILE REPORTING
4965  010430  000000                  EVTMIN: .WORD   0
4966  010432  000000                  EVTTCK: .WORD   0
4967  010434  000000                  EVTADD: .WORD   0       ;TEMP. LOC. TO HOLD ADDRESS DURING EVENT REPORTING
4968  010436  000000                  EVTBCT: .WORD   0       ;       ''      '' BYTE COUNT   ''      ''      ''
4969  010440  000000                  EVTTMP: .WORD   0       ;       ''      '' OTHER DATA   ''      ''      ''
4970
4971                                  ;REPORT CODING DISPATCH TABLE
4972
4973  010442  025324                  RPTDSP: .WORD   RPTTXQ  ;TRANSMIT QUEUED ENTRY DECODING
4974  010444  025324                          .WORD   RPTTXQ  ;TRANSMIT COMPLETE ENTRY DECODING
4975  010446  025324                          .WORD   RPTTXQ  ;RECEIVER QUEUED ENTRY DECODING
4976  010450  025324                          .WORD   RPTTXQ  ;RECEIVER COMPLETE ENTRY DECODING
4977  010452  025376                          .WORD   RPTDER  ;DEVICE ERROR ENTRY DECODING
4978  010454  025472                          .WORD   RPTDVI  ;DEVICE INIT ENTRY DECODING
4979  010456  025666                          .WORD   RPTDCK  ;DATA COMPARISON ENTRY DECODING
4980  010460  025152                          .WORD   RPT     ;PLACE HOLDER
4981  010462  025666                          .WORD   RPTDLE  ;DATA COMPARISON LENGH ERROR
4982  010464  025612                          .WORD   RPTDDE  ;DATA COMPARISON DATA ERROR
4983  010466  025536                          .WORD   RPTEOP  ;END OF PASS
4984  010470  025536                          .WORD   RPTABO  ;^C ABORT
4985
4986
4987  010472  000000          DEV1:   .WORD   0               ;TEMP LOCS TO HOLD DATA FOR EVENT REPORTING
4988  010474  000000          DEV2:   .WORD   0               ;  AND SHOW MODE,... SUBROUTINE
4989  010476  000000          DEV3:   .WORD   0
4990  010500  000000          DEV4:   .WORD   0
4991
```

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


010502                          CLITRE:

                                ;FIRST KEYWORD
010502                                  CLI     CLISPA,0,N10$                   ;SKIP ANY LEADIN SPACES
010506                          N10$:   CLI     <'?>,HLP,N42$                   ;IS THE FIRST NON-SP CHAR A ''?''
010512                                  CLI     CLIEXI,0                        ;  IF YES DO ''HLP'' AND EXIT
010514                          N42$:   CLI     CLISTR,HLP,N43$,<'HELP'>        ;ELSE, IS FIRST WORD A ''HELP''
010530                                  CLI     CLIEXI,0                        ;  IF YES DO ''HLP'' AND EXIT
010532                          N43$:   CLI     CLISTR,PRNT,N44$,<'PRINT'>      ;ELSE, IS FIRST WORD A ''PRINT''
010546                                  CLI     CLIEXI,0                        ;  IF YES DO ''PRINT'' AND EXIT
010550                          N44$:   CLI     CLISTR,EXIT,N45$,<'EXIT'>       ;ELSE,IS FIRST WORD AN ''EXIT''
010564                                  CLI     CLIEXI,0                        ;  IF YES DO ''EXIT'' AND EXIT
010566                          N45$:   CLI     CLISTR,RUN,N46$,<'RUN'>         ;ELSE, IS FIRST WORD A ''RUN''
010600                                  CLI     CLIBR,0,N80$                    ;  IF YES DO ''RUN'' & GOTO N80$
010604                          N46$:   CLI     CLISTR,NOTNUF,N40$,<'DUMP'>     ;ELSE, IS FIRST WORD A ''DUMP''
010620                                  CLI     CLIBR,0,N50$                    ;  IF YES GOTO N80$
010624                          N40$:   CLI     CLISTR,CLEAR,N20$,<'CLEAR'>     ;ELSE, IS FIRST WORD A ''CLEAR''
010640                                  CLI     CLIBR,NOTNUF,N100$              ;  IF YES DO ''CLR'' & GOTO N100$
010644                          N20$:   CLI     <'S>,NOTNUF,N30$                ;ELSE, IS FIRST CHAR. A ''S''
010650                                  CLI     CLISTR,SHOW,N25$,<'HOW'>        ;  IF YES IS REST OF WORD ''HOW''
010662                                  CLI     CLIBR,0,N100$                   ;    IF YES, DO ''SHOW'',BR N100$
010666                          N25$:   CLI     CLISTR,0,N30$,<'ET'>            ;  ELSE, IS REST OF WORD ''ET''
010700                                  CLI     CLIBR,0,N110$                   ;    IF YES, DO ''SET'', BR N110$
010704                          N30$:   CLI     CLIERR,0                        ;OTHERWISE ''ILL CMD'' - EXIT

                                ;SECOND KEYWORD (MODE=) FOR RUN COMMAND

010706                          N80$:   CLI     CLISPA,0,N30$                   ;SKIP LEADING SPS, IF NONE-ERR
010712                          N81$:   CLI     CLISTR,NOTNUF,N30$,<'MODE'>     ;IS NEXT WORD ''MODE=''
010726                                  CLI     <'=>,0,N30$                     ;  IF NO, IT'S WRONG -ERR -EXIT
010732                                  CLI     CLISTR,ATVMOD,N82$,<'ACTIVE'>   ;IS NEXT WORD ''ACTIVE''
010750                                  CLI     CLIBR,0,N115$                   ;  IF YES, DO ''ACTIVE'',BR N115$
010754                          N82$:   CLI     CLISTR,PASMOD,N83$,<'PASSIVE'>  ;IS NEXT WORD ''PASSIVE''
010772                                  CLI     CLIBR,0,N115$                   ;  IF YES, DO ''PASSVE'',BR N115$
010776                          N83$:   CLI     CLISTR,RECMOD,N84$,<'RECEIVE'>  ;IS NEXT WORD ''RECEIVE''
011014                                  CLI     CLIBR,0,N115$                   ;  IF YES, DO ''RECVE'',BR N115$
011020                          N84$:   CLI     CLISTR,LISMOD,N85$,<'LISTEN'>   ;IS NEXT WORD ''LISTEN''
011036                                  CLI     CLIBR,0,N115$                   ;  IF YES, DO ''LISTEN'',BR N115$
011042                          N85$:   CLI     CLISTR,DLLMOD,N86$,<'DOWNLINELOAD'>     ;IS NEXT WORD ''DOW...''
011066                                  CLI     CLIBR,0,N115$                   ;  IF YES, DO ''DWNLL'',BR N115$
011072                          N86$:   CLI     <'T>,0,N30$                     ;IS NEXT CHAR A ''T''
```

```
 5049  011076                          CLI     CLISTR,TRAMOD,N87$,<'RANSMIT'>  ; IS REST OF WORD 'RANSMIT"
 5050  011114                          CLI     CLIBR,0,N115$                   ;  IF YES, DO "TRANSM",BR N115$
 5051  011120                N87$:     CLI     CLISTR,TALMOD,N30$,<'ALK'>      ; IS REST OF WORD "ALK"
 5052  011132                          CLI     CLIBR,0,N115$                   ;  IF YES, DO "TALK",BR N115$
 5053                                                                          ; IF NO, ERROR - EXIT
 5054
 5055                                ;SECOND KEYWORD (FOR CLEAR OR SHOW)
 5056  011136                N100$:    CLI     CLISPA,0,N30$                   ;SKIP LEADING SPACES, NONE=ERR
 5057  011142                N102$:    CLI     CLISTR,CSHEXP,N104$,<'EXPECT'>  ;IS NEXT WORD "EXPE..."
 5058  011160                          CLI     CLIEXI,0                        ; IF YES, DO CLR-EXP,EXIT
 5059  011162                N104$:    CLI     CLISTR,CSHTRN,N30$,<'TRANSMIT'> ;IS NEXT WORD "TRANS.."
 5060  011202                          CLI     CLIEXI,0                        ; IF YES, DO CLR-TRN,EXIT
                                                                               ;IF NO - ERROR - EXIT
 5061
 5062
 5063
 5064                                ;SECOND KEYWORD (FOR SET)
 5065  011204                N110$:    CLI     CLISPA,0,N30$
 5066  011210                N111$:    CLI     CLISTR,SETEXP,N112$,<'EXPECT'>
 5067  011226                          CLI     CLIBR,0,N120$
 5068  011232                N112$:    CLI     CLISTR,SETTRN,N30$,<'TRANSMIT'>
 5069  011252                          CLI     CLIBR,0,N120$
 5070
 5071                                ;GET ADDRESSES FOR DUMP COMMAND
 5072  011256                N50$:     CLI     CLIALP,0,N51$
 5073  011262                N51$:     CLI     CLISPA,0,N52$
 5074  011266                N52$:     CLI     CLIOCT,DMPS,N30$
 5075  011272                          CLI     <'->,NOTNUF,N125$
 5076  011276                          CLI     CLIOCT,DMPE,N30$
 5077  011302                          CLI     <'/>,NOTNUF,N125$
 5078  011306                          CLI     <'B>,DMPQ,N30$
 5079  011312                          CLI     CLIBR,0,N125$
 5080
 5081                                ;QUALIFIERS FOR THE RUN COMMAND
 5082  011316                N115$:    CLI     CLIALP,0,N114$
 5083  011322                N114$:    CLI     <'/>,NOTNUF,N125$
 5084  011326                          CLI     CLISTR,NO,N116$,<'NO'>
 5085  011340                N116$:    CLI     <'C>,0,N117$
 5086  011344                          CLI     CLISTR,CHECK,N117$,<'HECK'>
 5087  011360                          CLI     CLIBR,0,N115$
 5088
 5094
 5095                                ;N113$: CLI     CLISTR,CRC,N30$,<'RC16'>
 5096                                ;       CLI     CLIBR,0,N115$
 5097
 5098  011364                N117$:    CLI     CLISTR,STATUS,N118$,<'STATUS'>
 5099  011402                          CLI     CLIBR,0,N115$
 5100  011406                N118$:    CLI     CLISTR,ECHO,N130$,<'ECHO'>
 5101  011422                          CLI     CLIBR,0,N115$
 5102
 5115
 5116  011426                N130$:    CLI     CLISTR,0,N131$,<'PASS'>
 5117  011442                          CLI     CLIBR,0,N150$
 5118  011446                N131$:    CLI     CLISTR,0,N132$,<'LOOP'>
 5119  011462                          CLI     CLIBR,0,N140$
 5120
 5121  011466                N132$:    CLI     CLISTR,MOSC,N30$,<'MODEM'>      ;MODEM ACTION
```

```
5122  011502                            CLI     CLIBR,0,N115$
5123
5124                                  ;GET MESSAGE TYPE FOR SET MESSAGE COMMANDS
5125  011506                          N120$:  CLI     <'=>,0,N30$
5126
5127                                  ;   LOOK FOR DEFAULT MESSAGE NAME
5128  011512                          N60$:   CLI     CLISTR,CMSG1,N61$,<'ONES'>
5129  011526                                  CLI     CLIBR,0,N121$
5130  011532                          N61$:   CLI     CLISTR,CMSG0,N62$,<'ZEROES'>
5131  011550                                  CLI     CLIBR,0,N121$
5132  011554                          N62$:   CLI     CLISTR,CMSG2,N63$,<'1ALT'>
5133  011570                                  CLI     CLIBR,0,N121$
5134  011574                          N63$:   CLI     CLISTR,CMSG3,N64$,<'0ALT'>
5135  011610                                  CLI     CLIBR,0,N121$
5136  011614                          N64$:   CLI     CLISTR,CMSG5,N65$,<'ITEP'>
5137  011630                                  CLI     CLIBR,0,N121$
5138  011634                          N65$:   CLI     CLISTR,CMSG4,N66$,<'CCITT'>
5139  011650                                  CLI     CLIBR,0,N121$
5140  011654                          N66$:   CLI     CLISTR,CMSG6,N67$,<'ALPHA'>
5141  011670                                  CLI     CLIBR,0,N121$
5142  011674                          N67$:   CLI     CLISTR,SETET,N68$,<'TRANSMIT'>
5143  011714                                  CLI     CLIBR,0,N125$
5144
5145                                  ;   LOOK FOR QUOTED MESSAGE
5146  011720                          N68$:   CLI     <''>,OPRMSG,N30$
5147  011724                          N70$:   CLI     <''>,ENDQO,N71$
5148  011730                                  CLI     CLIBR,0,N121$
5149  011734                          N71$:   CLI     CLISPA,0,N72$
5150  011740                          N72$:   CLI     CLIALN,0,N73$           ;ONLY A-Z,SP,TAB, OR 0-9 BETWEEN '"S
5151  011744                                  CLI     CLIBR,0,N70$
5152  011750                          N73$:   CLI     CLIERR,BADCHR           ;PRINT ERROR IF NONE LEGAL CHAR FOR '"S
5153
5154                                  ;GET QUALIFIERS (SIZE OR COPY) FOR SET MESSAGE COMMANDS
5155  011752                          N121$:  CLI     CLIALP,0,N123$
5156  011756                          N123$:  CLI     <'/>,NOTNUF,N125$
5157  011762                                  CLI     CLISTR,SIZE,N122$,<'SIZE'>
5158  011776                                  CLI     CLIBR,0,N126$
5159  012002                          N122$:  CLI     CLISTR,QCOPY,N30$,<'COPY'>
5160  012016                                  CLI     CLIBR,0,N126$
5161
5162                                  ;NUMER FOR SIZE OR COPY
5163  012022                          N126$:  CLI     <'=>,0,N30$
5164  012026                                  CLI     CLIDEC,NUM,N30$
5165  012032                                  CLI     CLIBR,0,N121$
5166
5167                                  ;GET MAINTENANCE LOOP TYPE FOR RUN 'LOOP" QUIALIFIER
5168  012036                          N140$:  CLI     <'=>,0,N30$
5169
5178
5179  012042                          N141$:  CLI     CLISTR,TTLLOP,N142$,<'INTERNALTTL'>
5180  012064                                  CLI     CLIBR,0,N115$
5181  012070                          N142$:  CLI     CLISTR,CBLLOP,N143$,<'CABLE'>
5182  012104                                  CLI     CLIBR,0,N115$
5183  012110                          N143$:  CLI     CLISTR,LMDLOP,N144$,<'LOCALMODEM'>
5184  012132                                  CLI     CLIBR,0,N115$
5185  012136                          N144$:  CLI     CLISTR,RMDLOP,N30$,<'REMOTEMODEM'>
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 23-25
CZKMUA.P11    30-MAR-82 09:13                    COMMAND LINE ACTION TREE

  5186  012160                                   CLI     CLIBR,0,N115$
  5187
  5188                                 ;GET LINE NUMBER FOR "PASS" RUN QUALIFIER
  5189  012164                         N150$:  CLI     <'=>,0,N30$
  5190  012170                                 CLI     CLIDEC,PASC,N30$
  5191  012174                                 CLI     CLIBR,0,N115$
  5192
  5193
  5194
  5195                                 ;END-OF-LINE
  5196  012200                         N125$:  CLI     CLIEXI,0
  5197
```

```
 5209
 5210
 5211                                   ;DEVICE DEPENDENT STORAGE LOCATIONS FOR
 5212                                   ;  CURRENT DEVICE PARAMTERS
 5213
 5214  012202                           SEL0:
 5215  012202  000000                   BSEL0:  .WORD   0               ;ADDRESSES OF REGISTERS SEL0 THRU BSEL7
 5216  012204  000000                   BSEL1:  .WORD   0
 5217  012206                           SEL2:
 5218  012206  000000                   BSEL2:  .WORD   0
 5219  012210  000000                   BSEL3:  .WORD   0
 5220  012212                           SEL4:
 5221  012212  000000                   BSEL4:  .WORD   0
 5222  012214  000000                   BSEL5:  .WORD   0
 5223  012216                           SEL6:
 5224  012216  000000                   BSEL6:  .WORD   0
 5225  012220  000000                   BSEL7:  .WORD   0
 5226
 5227
 5228  012222  000000                   INVEC:  .WORD   0               ;INPUT INTERRUPT VECTOR ADDRESS
 5229  012224  000000                   OUTVEC: .WORD   0               ;OUTPUT INTERRUPT VECTOR ADDRESS
 5230  012226  000000                   INTPRI: .WORD   0               ;INTERRUPT PRIORITY
 5231
 5232
 5245
 5255
 5256                                   ;       ERRTBL
```

```
                                .SBTTL  GLOBAL TEXT SECTION

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

                                .SBTTL          DEVICE SUPPORTED
                                ;
                                ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                                ;
012230                                  DEVTYP  <KMS11-BL,KMS11-BM>
012230                                                                  L$DVTYP::
012230  046513  030523  026461                                                   .ASCIZ  /KMS11-BL,KMS11-
012236  046102  045454  051515
012244  030461  041055  000115
                                                                                .EVEN


                                .SBTTL          PROGRAM IDENTIFICATION
                                ; TEST DESCRIPTION
                                ;
012252                                  DESCRIPT        <CZKMUA0 KMS11-BL PDP-11 DCLT>
012252                                                                  L$DESC::
012252  055103  046513  040525                                                   .ASCIZ  /CZKMUA0 KMS11-B
012260  020060  046513  030523
012266  026461  046102  050040
012274  050104  030455  020061
012302  041504  052114     000
012310                                                                          .EVEN
                                        .EVEN
                                .EVEN
```

```
 5305                                          .SBTTL          GLOBAL FORMAT STATEMENTS, MESSAGES, AND ASCII INFO
 5309
 5310  012310  041504  052114  000076  CLI$PM: .ASCIZ  /DCLT>/
 5311  012316  050122  037124  000040  CLI$RP: .ASCIZ  /RPT> /
 5312  012324  047045  040445  044477  CLIERM: .ASCIZ  /%N%A?ILL CMD-BAD SYNTX?/
 5313  012354  047045  040445  044477  CLINUF: .ASCIZ  /%N%A?INCMPLTE CMD?/
 5314  012377     045  022516  037501  CLINBG: .ASCIZ  /%N%A?NUM TOO BIG?/
 5315  012421     045  022516  037501  CLIBRX: .ASCIZ  /%N%A?BAD RADIX?/
 5316  012441     045  022516  037501  CLIBDL: .ASCIZ  /%N%A?''LOOP'' VALID ONLY IN ACTIVE?/
 5317  012503     045  022516  037501  CLINPS: .ASCIZ  /%N%A?''ECHO'' VALID ONLY IN PASSIVE?/
 5318  012546  047045  040445  044477  CLIBCR: .ASCIZ  /%N%A?ILL CHR- ''A-Z,0-9,SP,TAB'' ONLY?/
 5319  012613     045  022516  037501  CLISE0: .ASCIZ  /%N%A?''SIZE=0'' NOT VALID?/
 5320  012644  047045  040445  052077  CLIPW:  .ASCIZ  /%N%A?TRANSMIT & EXPECT LIST MUST BE IDENTICAL FOR LOOP?/
 5321  012734  040523  052124  046105  DLLQ1:  .ASCIZ  /SATTELITE PASSWORD= /
 5322  012761     045  022516  052101  HLP0:   .ASCIZ  /%N%ATHIS IS DCLT. TYPE 'H'' OR ''?'' FOR DETAILS/
 5323  013037     045  022516  000124  HLPF:   .ASCIZ  /%N%T/
 5324  013044  041504  052114  041440  HLP1:   .ASCIZ  /DCLT CMDS:/
 5325  013057     040  046103  040505  HLP2:   .ASCII  / CLEAR OR SHOW  EXPECTLIST OR TRANSMITLIST/<15><12>
 5326  013133     040  051120  04711'          .ASCII  / PRINT/<15><12>
 5327  013143     040  054105  052111          .ASCII  / EXIT/<15><12>
 5328  013152  042040  046525  020120          .ASCIZ  ? DUMP START-END/B?
 5329  013174  051440  052105  042440  HLP3:   .ASCIZ  ? SET EXPECTMSG OR TRANSMITMSG=TYPE/SIZE=N OR /COPY=N?
 5330  013261     040  042523  020124  HLP3A:  .ASCIZ  / SET EXPECT=TRANSMIT/
 5331  013306  020040  052040  050131  HLP4:   .ASCIZ  ?   TYPE=ONES,ZEROES,1ALT,0ALT,ITEP,CCITT,ALPHA?
 5332  013365     040  020040  020040  HLP4A:  .ASCIZ  /       OR 'OPR SPCD=A-Z,SP,TAB,0-9 IN QUOTES''/
 5333  013443     040  052522  020116  HLP5:   .ASCIZ  ? RUN MODE=MTYP/LOOP=LTYP/CHECK,STATUS,ECHO,MODEM,PASS=N?
 5334  013533     040  020040  052115  HLP6:   .ASCII  /   MTYP=TRAN,REC,ACT,PAS,TAL,LIS,DOWN/<15><12>
 5335  013602  020040  046040  054524          .ASCIZ  /   LTYP=INT,CAB,LOC,REM/
 5336  013632  047045  040445  054524  RHLP0:  .ASCIZ  /%N%ATYPE 'H'' OR ''?'' FOR HELP !/
 5337  013671     104  046103  020124  RHLP1:  .ASCIZ  /DCLT REPORT CMDS :/
 5338  013714  047514  020107  020055  RHLP2:  .ASCIZ  /LOG - PRINT DCLT EVENT LOG/
 5339  013747     105  044530  020124  RHLP3:  .ASCIZ  /EXIT - EXIT REPORT LEVEL/
 5340  014000  042510  050114  026440  RHLP4:  .ASCIZ  /HELP - PRINT THIS MESSAGE/
 5341  014032  040502  042523  042457  RHLP5:  .ASCIZ  !BASE/ERROR - PRINT ONLY ERRORS!
 5342  014071     102  051501  027505  RHLP6:  .ASCIZ  !BASE/FULL - PRINT ENTIRE TABLE!
 5343  014130  040502  042523  (.7457  RHLP7:  .ASCIZ  !BASE/OFFSET=NNN - PRINT SINGLE LOCATION!<15><12>
 5344  014202  047045  040445  040502  RPTIV:  .ASCIZ  /%N%ABASE OFFSET=%03%A TOO BIG !/
 5345  014242  047045  040445  051515  SHMSG:  .ASCIZ  ?%N%AMSG: TYPE=%T%A/SIZE=%D3?
 5346  014276  042532  047522  051505  SHTYP0: .ASCIZ  /ZEROES/
 5347  014305     117  042516  000123  SHTYP1: .ASCIZ  /ONES/
 5348  014312  040461  052114     000  SHTYP2: .ASCIZ  /1ALT/
 5349  014317     060  046101  000124  SHTYP3: .ASCIZ  /0ALT/
 5350  014324  041503  052111  000124  SHTYP4: .ASCIZ  /CCITT/
 5351  014332  052111  050105     000  SHTYP5: .ASCIZ  /ITEP/
 5352  014337     101  050114  040510  SHTYP6: .ASCIZ  /ALPHA/
 5353  014345     117  051120  051440  SHTYP7: .ASCIZ  /OPR SPEC/
 5354  014356  042522  042503  053111  M00:    .ASCIZ  /RECEIVE/
 5355  014366  051124  047101  046523  M01:    .ASCIZ  /TRANSMIT/
 5356  014377     120  051501  044523  M02:    .ASCIZ  /PASSIVE/
 5357  014407     101  052103  053111  M03:    .ASCIZ  /ACTIVE/
 5358  014416  047504  047127  044514  M04:    .ASCIZ  /DOWNLINELOAD/
 5359  014433     124  046101  000113  M05:    .ASCIZ  /TALK/
 5360  014440  044514  052123  047105  M06:    .ASCIZ  /LISTEN/
 5361  014447     000                  LP0:    .ASCIZ  //
 5362  014450  046057  047517  036520  LP00:   .ASCIZ  ?/LOOP=?
 5363  014457     111  052116  051105  LP1:    .ASCIZ  ?INTERNAL?
```

```
 5364  014470  040503  046102  000105  LP2:    .ASCIZ  ?CABLE?
 5365  014476  047514  040503  046514  LP3:    .ASCIZ  ?LOCALMODEM?
 5366  014511     122  046505  052117  LP4:    .ASCIZ  ?REMOTEMODEM?
 5367  014525     116     117          PNST:   .ASCII  /NO/
 5368  014527     123  040524  052524  PST:    .ASCIZ  /STATUS/
 5369  014536  047516                  PNCK:   .ASCII  /NO/
 5370  014540  044103  041505  000113  PCK:    .ASCIZ  /CHECK/
 5371  014546  047516                  PNEC:   .ASCII  /NO/
 5372  014550  041505  047510     000  PEC:    .ASCIZ  /ECHO/
 5373
 5384  014555     116     117          PNMS:   .ASCII  /NO/
 5385  014557     115  042117  046505  PMS:    .ASCIZ  /MODEM/
 5386  014565     045  022516  046101  LISP:   .ASCIZ  /%N%ALIS>/
 5387  014576  046124  037113     000  OPRMM:  .ASCIZ  /TLK>/
 5388  014603     124  044510  020123  L5060:  .ASCIZ  /THIS A 50. OR 60. HZ. LSI-11:/
 5389          014642                  .EVEN
 5390
 5391
 5392
 5393                                  ;
 5394                                  ; FORMAT STATEMENTS USED IN PRINT CALLS
 5395                                  ;
 5396
 5397  014642  047045  040445  047504  DLLCM:  .ASCIZ  /%N%ADOWN LINE LOAD COMPLETED SUCCESSFULLY/
 5398  014714  047045  040445  040502  NOCLK:  .ASCIZ  /%N%ABAD CLOCK - PROGRAM WILL HANG ON 'TIMEOUT'!!/
 5399  014775     115  054101  020056  TABEX:  .ASCIZ  /MAX. CHAR. MSG COUNT EXCEEDED -/
 5400  015035     102  043125  042506  BUFEX:  .ASCIZ  /BUFFER FULL -/
 5401  015053     045  022516  022524  MSGTRN: .ASCIZ  /%N%T%A MSG. NOT BUILT !!/
 5402  015104  047045  040445  044103  MSGTRU: .ASCIZ  /%N%ACHAR. COUNT EXCEEDS BUFF LIMIT - MSG TRUNCATED/
 5403  015167     045  022516  032523  SHF0:   .ASCIZ  ?%N%S5%AMODE=%T%T%T%A/PASS=%Z5?
 5404
 5410
 5411  015225     045  022516  032523  SHF1:   .ASCIZ  ?%N%S5%S5%S5%A/%T%A/%T%A/%T%A/%T?
 5412
 5413  015265     045  032523  040445  EFM2:   .ASCIZ  /%S5%ATOTAL MISMATCHES IN MSG = %D5/
 5414  015330  047045  051445  022463  PCPM:   .ASCIZ  /%N%S3%ACALLED FROM PC=%O6/
 5415  015362  051445  022465  041501  EFM11:  .ASCIZ  /%S5%ACOMPARE COUNT=%D5%S3%ARECEIVE COUNT=%D5/
 5416
 5417
 5418                                  ;EVENT DESCRIPTION MESSAGES
 5419
 5420  015437     124  040522  051516  EDTXQ:  .ASCIZ  /TRANSMIT MSG QUEUED/
 5421  015463     124  040522  051516  EDTXC:  .ASCIZ  /TRANSMIT MSG COMPLETED/
 5422  015512  042522  042503  053111  EDRXQ:  .ASCIZ  /RECEIVE SPACE QUEUED/
 5423  015537     122  041505  044505  EDRXC:  .ASCIZ  /RECEIVE MSG COMPLETED/
 5424  015565     104  053105  041511  EDDER:  .ASCIZ  /DEVICE ERROR/
 5425  015602  040504  040524  041440  EDDCK:  .ASCIZ  /DATA COMPARISON STARTED/
 5426  015632  042504  044526  042503  EDDVI:  .ASCIZ  /DEVICE INIT AND SETUP/
 5427  015660  040504  040524  041440  EDDLE:  .ASCIZ  /DATA COMPARISON LENGTH ERROR/
 5428  015715     104  052101  020101  EDDDE:  .ASCIZ  /DATA COMPARISON DATA ERROR/
 5429  015750  047105  020104  043117  EDEOP:  .ASCIZ  /END OF PASS/
 5430  015764  041101  047516  046522  EDMOS:  .ASCIZ  /ABNORMAL MODEM STATUS CHANGE/
 5431  016021     136  020103  041101  EDABO:  .ASCIZ  /^C ABORT/
 5432
 5433                                  ;*****************************************************************
 5434                                  ;THESE TWO STORAGE AREAS MUST NOT BE SEPERATED !!!!
```

```
 5435
 5436                                   ;EVENT REPORTING MESSAGES
 5437  016032  020040  040502  042523   BASM1A: .ASCIZ  /  BASE TABLE/
 5438
 5439  016047     045  031523  047445   BASM3:  .ASCIZ  /%S3%03/
 5440  016056  051445  022463  033117   BASM2:  .ASCIZ  /%S3%06/
 5441  016065     045  022516  033117   BASM1:  .ASCIZ  /%N%06/
 5442
 5443
 5444
 5445
 5446  016073     045  022516  052101   NULEVT: .ASCIZ  ?%N%ATHE DCLT EVENT LOG IS EMPTY?
 5447  016133     045  022516  037101   EVTF0:  .ASCIZ  ?%N%A>>>DCLT EVENT LOG ENTRY <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<?
 5448  016226  047045  042045  022465   EVTF1:  .ASCIZ  /%N%D5%A:%Z2%A:%Z2%S3%T/
 5449  016255     045  022516  031523   EVTF2:  .ASCIZ  /%N%S3%AADDR OF MSG=%06%S3%ABYTE COUNT=%D5/
 5450  016327     045  022516  031523   EVTF3:  .ASCIZ  /%N%S3%T%N/
 5451  016341     045  031523  047445   EVTF3C: .ASCIZ  /%S3%06%S3%06/
 5452  016356  051445  022463  033117   EVTF3D: .ASCIZ  /%S3%06%S3%06%S3%T/
 5453  016400  047045  051445  022463   EVTF4:  .ASCIZ  /%N%S3%AADDR OF MSG=%06%S3%ABYTE COUNT=%D5%S3%ANO. OF CMP ERRS=%D5/
 5454  016502  047045  051445  022463   EVTF4A: .ASCIZ  /%N%S3%AADDR OF MSG=%06%S3%ARX BYTES=%D5%S3%ACOMPARE BYTES=%D5/
 5455  016600  047045  051445  022463   EVTF4B: .ASCIZ  /%N%S3%APASS=%D5%S3%AERRORS=%D5%S3%ANOBUFFS=%D5/
 5456  016657     045  032523  040445   EVTF5A: .ASCIZ  /%S5%ABYTE # IN MSG.=%D5%S3%AEXPTD=%03%S3%ARECVD=%03/
 5457
 5458  016743     045  022516  034523   EVMOCG: .ASCIZ  /%N%S9%ACHANGED TO:/
 5459
 5460                                   ; ************************************************************************
 5461                                   ;DO NOT SEPERATE THE NEXT LIST OF MESSAGES - MODEM SIGNAL HEADER AND REPORT
 5462
 5463  016766  047045  051445  022470   EVMOHD: .ASCIZ  /%N%S8%AMODEM STATUS: CTS DSR DCD RTS RI  SQD TM/
 5464  017046  047045  051445  022471   EVMOST: .ASCII  /%N%S9%S9%S5%A/
 5465  017063     130     040     040   EVMCTS: .BYTE   'X,40,40,40
 5466  017067     130     040     040   EVMDSR: .BYTE   'X,40,40,40
 5467  017073     130     040     040   EVMDCD: .BYTE   'X,40,40,40
 5468  017077     130     040     040   EVMRTS: .BYTE   'X,40,40,40
 5469  017103     130     040     040   EVMRI:  .BYTE   'X,40,40,40
 5470  017107     130     040     040   EVMSQD: .BYTE   'X,40,40,40
 5471  017113     130     040     040   EVMTM:  .BYTE   'X,40,40,40
 5472  017117     000                           .BYTE   0
 5473                                           .EVEN
 5474
 5478
 5479
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 23-31
CZKMUA.P11    30-MAR-82 09:13           BASE TABLE ADDRESS

  5481                                  .SBTTL  BASE TABLE ADDRESS
  5482                                  ;THIS SECTION IS USED BY A M9301-YJ BOOT ROM FOR DOING DOWN-LINE-LOAD.
  5483                                  ;MUST BE IN THE AREA OF ''017370 + 256. BYTES'' + A FEW
  5484
  5485                                  ;;;;;!!!!!! BEWARE !!!! DO NOT ALLOW THE ABOVE ASCIZ MESSAGES TO EXPAND INTO
  5486                                  ;;;;;!!!!!!!!!!!!!!!!!!!!  THIS REGION.
  5487                                  .EVEN
  5488            017370                        .=17370
  5489  017370   000400                BASE:   .BLKB   256.            ;BASE TABLE ADDRESS
  5490            020000                        .=20000
  5491
```

```
                                        .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


                                        ;;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
```

```
 5539                                    .SBTTL  KMS11 BASE TABLE DATA DESCRIPTION MESSAGES
 5540  020322  047125  042504  044506  DMUNKN: .ASCIZ  /UNDEFINED DATA /       ;LOCATION UNDEFINED BY SPEC
 5541  020342  047101  020120  020055  DMC002: .ASCIZ  /ANP - CONSTANT 0/
 5542  020363     116  046124  020122  DMC003: .ASCIZ  /NTLR - NAKS RCVD..NO BUFFERS/
 5543  020420  044116  051104  026440  DMC004: .ASCIZ  /NHDR - NAKS RCVD..MSG HEADER BAD/
 5544  020461     104  052101  020122  DMC005: .ASCIZ  /DATR - NAKS RCVD..DATA BAD/
 5545  020514  052116  051514  026440  DMC006: .ASCIZ  /NTLS - NAKS SENT..NO BUFFERS/
 5546  020551     116  042110  020123  DMC007: .ASCIZ  /NHDS - NAKS SENT..BAD HEADER/
 5547  020606  040504  051524  026440  DMC010: .ASCIZ  /DATS - NAKS SENT..BAD DATA/
 5548  020641     122  050105  051503  DMC011: .ASCIZ  /REPCS - REPS SENT/
 5549  020663     122  050105  051103  DMC012: .ASCIZ  /REPCR - REPS RECD/
 5550  020705     102  051501  020105  DMC013: .ASCIZ  /BASE - CORE TABLE BASE ADDRESS/
 5551  020744  042523  020105  046504  DMC377: .ASCIZ  /SEE DMC TECH MANUAL FOR DESCRIPTION/
```

```
5560
5561                                    ;DEVICE ERROR MESSAGES
5562
5563  021010  044524  042515  047440  DVEM0:  .ASCII  /TIME OUT WAITING FOR RDI TO CLEAR/
5564  021051     015  020012  020040          .ASCIZ  <15><12>/   SEL0     SEL2  /
5565  021076  044524  042515  047440  DVEM1:  .ASCII  /TIME OUT WAITING FOR RDI TO SET/
5566  021135     015  020012  020040          .ASCIZ  <15><12>/   SEL0     SEL2  /
5567  021162  044524  042515  047440  DVEM3:  .ASCII  /TIME OUT WAITING FOR RUN TO SET/
5568  021221     015  020012  020040          .ASCIZ  <15><12>/   SEL0     SEL2  /
5569  021246  044524  042515  047440  DVEM4:  .ASCII  /TIME OUT WAITING FOR OUTPUT INTERRUPT/
5570  021313     015  020012  020040          .ASCIZ  <15><12>/   SEL0     SEL2  /
5571  021340  047111  052520  020124  DVEM5:  .ASCII  /INPUT INTERRUPT WHEN EXPECTING OUTPUT/
5572  021405     015  020012  020040          .ASCIZ  <15><12>/   SEL0     SEL2  /
5573  021432  046111  042514  040507  DVEM6:  .ASCII  /ILLEGAL OUTPUT INTERRUPT/
5574  021462  005015  020040  051440          .ASCIZ  <15><12>/   SEL2     SEL6  /
5575  021507     103  047117  051124  DVEM7:  .ASCII  /CONTROL OUT INSTEAD OF BA-CC OUT/
5576  021547     015  020012  020040          .ASCIZ  <15><12>/   SEL2     SEL6  /
5577
5578  021574  054124  041040  043125  DVEM8:  .ASCII  /TX BUFF COMPLETED AND SHOULD BE RX/
5579  021636  005015  020040  051440          .ASCIZ  <15><12>/   SEL4     SEL6  /
5580  021663     122  020130  052502  DVEM9:  .ASCII  /RX BUFF COMPLETED AND SHOULD BE TX/
5581  021725     015  020012  020040          .ASCIZ  <15><12>/   SEL4     SEL6  /
5582  021752  042040  053517  020116  DLLAB:  .ASCII  / DOWN LINE LOAD ABORTED/
5583  022001     015  020012  020040          .ASCIZ  <15><12>/   RXBUF    TXBUF /
5584
5585  022026  051120  041517  042105  PROEM:  .ASCIZ  /PROCEDURE ERROR/
5586  022046  047516  020116  054105  NXMM:   .ASCIZ  /NON EXIST MEM/
5587  022064  042104  046503  020120  DDCSRM: .ASCIZ  /DDCMP START REC/
5588  022104  044504  041523  047117  DISCOM: .ASCIZ  /DISCONNECT/
5589  022117     114  051517  020124  LOSDAM: .ASCIZ  /LOST DATA/
5590  022131     104  041504  050115  DDCMRM: .ASCIZ  /DDCMP MAINT REC/
5591  022151     124  046511  020105  TIMOM:  .ASCIZ  /TIME OUT/
5592  022162  040504  040524  041440  DATCKM: .ASCIZ  /DATA CHECK/
5593  022175     122  047125  051440  RUNSBM: .ASCIZ  /RUN SET ILLEGALLY/
5594  022217     122  020130  042111  RXIDM:  .ASCIZ  /RX IDLE/
5595  022227     103  020104  046107  CDGLM:  .ASCIZ  /CD GLITCHED/
5596  022243     103  051524  043040  CTSFM:  .ASCIZ  /CTS FALILED/
5597  022257     124  020130  047516  TXNC:   .ASCIZ  /TX NOT COMPLETE/
5598  022277     122  020130  047516  RXNC:   .ASCIZ  /RX NOT COMPLETE/
5599  022317     123  041505  051040  RXM1:   .ASCIZ  /SEC REQ ERR WORD 1/
5600  022342  042523  020103  042522  RXM2:   .ASCIZ  /SEC REQ ERR WORD 2/
5601          022366                          .EVEN
5602
5606
5617
5618
5626
5627
5628
5629
```

```
                                .SBTTL  GLOBAL ERROR REPORT SECTION

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


022366                          BGNMSG  ERR1
022366                                                                  ERR1::
022366                          PRINTB  #EVTF5A,OFSET,<B,GOOD>,<B,BAD>  ;INDIVIDUAL DATA COMPARE ERROR
022366  005046                                                                  CLR     -(SP)
022370  153716  007265                                                          BISB    BAD,(SP)
022374  005046                                                                  CLR     -(SP)
022376  153716  007264                                                          BISB    GOOD,(SP)
022402  013746  007242                                                          MOV     OFSET,-(SP)
022406  012746  016657                                                          MOV     #EVTF5A,-(SP)
022412  012746  000004                                                          MOV     #4,-(SP)
022416  010600                                                                  MOV     SP,R0
022420  104414                                                                  TRAP    C$PNTB
022422  062706  000012                                                          ADD     #12,SP
022426                          ENDMSG
022426                                                                  L10001:
022426  104423                                                                  TRAP    C$MSG

022430                          BGNMSG  ERR2
022430                                                                  ERR2::
022430                          PRINTB  #EFM2,TEMP4                     ;TOTAL DATA COMPARE FAILS ERROR
022430  013746  007254                                                          MOV     TEMP4,-(SP)
022434  012746  015265                                                          MOV     #EFM2,-(SP)
022440  012746  000002                                                          MOV     #2,-(SP)
022444  010600                                                                  MOV     SP,R0
022446  104414                                                                  TRAP    C$PNTB
022450  062706  000006                                                          ADD     #6,SP
022454                          ENDMSG
022454                                                                  L10002:
022454  104423                                                                  TRAP    C$MSG

022456                          BGNMSG  ERR10
022456                                                                  ERR10::
022456                          PRINTB  #EFM11,R4,TEMP3
022456  013746  007252                                                          MOV     TEMP3,-(SP)
022462  010446                                                                  MOV     R4,-(SP)
022464  012746  015362                                                          MOV     #EFM11,-(SP)
022470  012746  000003                                                          MOV     #3,-(SP)
022474  010600                                                                  MOV     SP,R0
022476  104414                                                                  TRAP    C$PNTB
022500  062706  000010                                                          ADD     #10,SP
022504                          ENDMSG
022504                                                                  L10003:
022504  104423                                                                  TRAP    C$MSG
```

```
5679  022506                          BGNMSG  ERR8
 (3)  022506                                                          ERR8::
5680  022506                          PRINTB  #EVTF3D,TEMP3,TEMP4,CONOTM
 (10) 022506  013746  007260                                                   MOV     CONOTM,-(SP)
 (9)  022512  013746  007254                                                   MOV     TEMP4,-(SP)
 (8)  022516  013746  007252                                                   MOV     TEMP3,-(SP)
 (7)  022522  012746  016356                                                   MOV     #EVTF3D,-(SP)
 (6)  022526  012746  000004                                                   MOV     #4,-(SP)
 (3)  022532  010600                                                           MOV     SP,R0
 (4)  022534  104414                                                           TRAP    C$PNTB
 (4)  022536  062706  000012                                                   ADD     #12,SP
5681  022542                          PRINTB  #PCPM,PCADD
 (8)  022542  013746  007300                                                   MOV     PCADD,-(SP)
 (7)  022546  012746  015330                                                   MOV     #PCPM,-(SP)
 (6)  022552  012746  000002                                                   MOV     #2,-(SP)
 (3)  022556  010600                                                           MOV     SP,R0
 (4)  022560  104414                                                           TRAP    C$PNTB
 (4)  022562  062706  000006                                                   ADD     #6,SP
5682  022566                          ENDMSG
 (3)  022566                                                          L10004:
 (3)  022566  104423                                                           TRAP    C$MSG
5683
5684  022570                          BGNMSG  ERR9
 (3)  022570                                                          ERR9::
5685  022570                          PRINTB  #EVTF3C,TEMP3,TEMP4
 (9)  022570  013746  007254                                                   MOV     TEMP4,-(SP)
 (8)  022574  013746  007252                                                   MOV     TEMP3,-(SP)
 (7)  022600  012746  016341                                                   MOV     #EVTF3C,-(SP)
 (6)  022604  012746  000003                                                   MOV     #3,-(SP)
 (3)  022610  010600                                                           MOV     SP,R0
 (4)  022612  104414                                                           TRAP    C$PNTB
 (4)  022614  062706  000010                                                   ADD     #10,SP
5686  022620                          PRINTB  #PCPM,PCADD
 (8)  022620  013746  007300                                                   MOV     PCADD,-(SP)
 (7)  022624  012746  015330                                                   MOV     #PCPM,-(SP)
 (6)  022630  012746  000002                                                   MOV     #2,-(SP)
 (3)  022634  010600                                                           MOV     SP,R0
 (4)  022636  104414                                                           TRAP    C$PNTB
 (4)  022640  062706  000006                                                   ADD     #6,SP
5687  022644                          ENDMSG
 (3)  022644                                                          L10005:
 (3)  022644  104423                                                           TRAP    C$MSG
5688
5689  022646                          BGNMSG  ERR13
 (3)  022646                                                          ERR13::
5690  022646                          PRINTB  #EVTF3C,TEMP3,TEMP4
 (9)  022646  013746  007254                                                   MOV     TEMP4,-(SP)
 (8)  022652  013746  007252                                                   MOV     TEMP3,-(SP)
 (7)  022656  012746  016341                                                   MOV     #EVTF3C,-(SP)
 (6)  022662  012746  000003                                                   MOV     #3,-(SP)
 (3)  022666  010600                                                           MOV     SP,R0
 (4)  022670  104414                                                           TRAP    C$PNTB
 (4)  022672  062706  000010                                                   ADD     #10,SP
5691  022676                          ENDMSG
 (3)  022676                                                          L10006:
 (3)  022676  104423                                                           TRAP    C$MSG
```

```
 5692
 5693  022700                          BGNMSG  ERR14                    ERR14::
  (3)  022700
 5694  022700                          PRINTB  #EVTF3D,TEMP3,TEMP4,CONOTM
  (10) 022700  013746  007260                                                    MOV     CONOTM,-(SP)
  (9)  022704  013746  007254                                                    MOV     TEMP4,-(SP)
  (8)  022710  013746  007252                                                    MOV     TEMP3,-(SP)
  (7)  022714  012746  016356                                                    MOV     #EVTF3D,-(SP)
  (6)  022720  012746  000004                                                    MOV     #4,-(SP)
  (3)  022724  010600                                                            MOV     SP,R0
  (4)  022726  104414                                                            TRAP    C$PNTB
  (4)  022730  062706  000012                                                    ADD     #12,SP
 5695  022734                          ENDMSG
  (3)  022734                                                          L10007:
  (3)  022734  104423                                                            TRAP    C$MSG
 5696
 5697  022736                          EXIT    MSG
  (4)  022736  000167                                                            .WORD   J$JMP
  (3)  022740  177772                                                            .WORD   L10007-2-.
 5698
 5699
```

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

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


                                .SBTTL          CLOCK SETUP SUBROUTINE

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

                                CLKSET:
022742                                  MOV     (R1)+,(R2)+             ;LOAD CLOCK'S CSR ADDR. INTO "CLKCSR"
022742  012122                          MOV     (R1)+,(R2)              ;LOAD CLOCK'S INT. LEVEL INTO "CLKBR"
022744  012112                          ASL     (R2)                    ;ADJUST THE INT. LEVEL FOR LOADING INTO
022746  006312                          ASL     (R2)                    ;    THE PSW WITH A "SETVEC" CALL
022750  006312                          ASL     (R2)
022752  006312                          ASL     (R2)
022754  006312                          ASL     (R2)+
022756  006322                          MOV     (R1)+,(R2)+             ;LOAD CLOCK'S INT. VECTOR INTO "CLKVEC"
022760  012122                          MOV     (R1)+,(R2)+             ;LOAD CLOCK'S HERTZ RATE INTO "CLKHZ"
022762  012122
022764  000207                          RTS     PC
```

```
5825
5826                            .SBTTL          CLOCK INTERRUPT SERVICE ROUTINE
5827                            ;++
5828                            ; FUNCTIONAL DESCRIPTION:
5829                            ;       THIS IS THE CLOCK INTERRUPT SERVICE ROUTINE WHICH TAKES CARE OF
5830                            ;       KEEPING THE "TIME-SINCE-START" AND COUNTING DOWN ANY OF THE
5831                            ;       "EVENT" TIMERS. THE TIMERS ARE USED TO TIME COMPLETION OF  DEVICE
5832                            ;       REQUESTS. THE "TIME-SINCE-START" IS USED TO BE LOGGED WITH EACH ENTRY
5833                            ;       INTO THE EVENT LOG.
5834                            ;
5835                            ; IMPLICIT INPUTS:
5836                            ;       TIMTCK: THE CURRENT NO. OF TICKS LEFT TO BE COUNTED UNTIL A SECOND
5837                            ;               HAS BEEN COUNTED OFF
5838                            ;       CLKHZ:  THE NO. OF TICKS IN A SECOND, DETERMINED BY THE SYS. LINE FREQ.
5839                            ;       TIMMIN & TIMSEC: CURRENT VALUE OF "TIME-SINCE-START"
5840                            ;                       IN MINUTES & SECONDS
5841                            ;       TIMER 1,2, & S: CURRENT VALUES OF THE "EVENT TIMERS"
5842                            ;
5843                            ; IMPLICIT OUTPUTS:
5844                            ;       NEW VALUE OF EVENT TIMER "1" DECREMENTED BY 1 TICK IF IT WAS NON-ZERO
5845                            ;       NEW VALUE OF EVENT TIMER "2" DECREMENTED BY 1 TICK IF IT WAS NON-ZERO
5846                            ;       NEW VALUE OF EVENT TIMER "S" DECREMENTED BY 1 SECOND IF IT WAS NON-ZERO
5847                            ;
5848                            ; FUNCTIONAL SIDE EFFECTS:
5849                            ;       THE CLOCK IS DISABLED UPON ENTRY AND REENABLED WHEN LEAVING
5850                            ;
5851                            ; CALLING SEQUENCE:
5852                            ;       THIS ROUTINE IS CALLED WHEN THE CLOCK INTERRUPTS THRU "CLKVEC".
5853                            ;       THE ADDRESS OF THIS ROUTINE WAS LOADED INTO THE CLOCK'S INTERRUPT
5854                            ;       VECTOR WITH A SUPERVISOR "SETVEC" CALL.
5855                            ;--
5856
5857  022766                                        BGNSRV  CLKINT
 (3)  022766                                                                    CLKINT::
5858
5859  022766  005077  164346                        CLR     @CLKCSR         ;DISABLE THE CLOCK FORM INTERRUPTING
5860  022772  005337  007356                        DEC     TIMTCK          ;DECREMENT THE # OF TICKS/SEC.
5861  022776  001015                                BNE     1$              ;GO CHECK TIMERS (1&2-TICKS, 3-SECONDS)
5862  023000  013737  007346  007356                MOV     CLKHZ,TIMTCK    ;RESET THE # OF TICKS/SEC.
5863  023006  005237  007354                        INC     TIMSEC          ;INC # OF SECS-SINCE-START
5864  023012  022737  000074  007354                CMP     #60.,TIMSEC     ;SEE IF WE'VE COUNTED 60 SECS. YET
5865  023020  001004                                BNE     1$              ;IF NOT, GO CHECK TIMERS
5866  023022  005237  007352                        INC     TIMMIN          ; ELSE INC MINUTES-SINCE-START
5867  023026  005037  007354                        CLR     TIMSEC          ; AND RESTART SECOND COUNTER
5868
5869  023032  005737  007360              1$:       TST     TIMER1          ;SEE IF TIMER #1, TIMING ANYTHING
5870  023036  001402                                BEQ     2$              ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
5871  023040  005337  007360                        DEC     TIMER1          ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
5872  023044  005737  007362              2$:       TST     TIMER2          ;SEE IF TIMER #2, TIMING ANYTHING
5873  023050  001402                                BEQ     3$              ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
5874  023052  005337  007362                        DEC     TIMER2          ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
5875  023056  005737  007364              3$:       TST     TIMERS          ;SEE IF TIMER #3, TIMING ANYTHING
5876  023062  001406                                BEQ     4$              ; IF=0, NOTHING BEING TIMED, LEAVE
5877  023064  023737  007346  007356                CMP     CLKHZ,TIMTCK    ;SEE IF A SECOND HAS BEEN COUNTED OFF
5878  023072  001002                                BNE     4$              ; BR IF NO
5879  023074  005337  007364                        DEC     TIMERS          ; ELSE DECREMENT THE TIMER VALUE (BY 1 SEC.)
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 23-40
CZKMUA.P11    30-MAR-82 09:13                    CLOCK INTERRUPT SERVICE ROUTINE

  5880  023100  013777  007350  164232  4$:     MOV     CLKEN,@CLKCSR   ;REENABLE THE CLOCK TO INTERRUPT
  5881  023106                                  ENDSRV
   (3)  023106                                                                  L10010:
   (2)  023106  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
                                ;--
023110                          LOGTXQ:
023110  012737  020003  007216          MOV     #STXQ,TEMP1     ;SET UP MSG. TO PRINT
023116  012737  000000  007244          MOV     #TXQ,TEMP       ;SET UP EVENT TYPE
023124  000510                          BR      LOGS1           ;GO LOG EVENT AND TIME

023126                          LOGTXC:
023126  012737  020014  007246          MOV     #STXC,TEMP1     ;SET UP MSG. TO PRINT
023134  012737  000002  007244          MOV     #TXC,TEMP       ;SET UP EVENT TYPE
023142  000501                          BR      LOGS1           ;GO LOG EVENT AND TIME

023144                          LOGRXQ:
023144  012737  020025  007276          MOV     #SRXQ,TEMP1     ;SET UP MSG. TO PRINT
023152  012737  000004  007244          MOV     #RXQ,TEMP       ;SET UP EVENT TYPE
023160  000472                          BR      LOGS1           ;GO LOG EVENT AND TIME

023162                          LOGRXC:
023162  012737  000006  007244          MOV     #RXC,TEMP       ;SET UP EVENT TYPE
023170  000466                          BR      LOGS1           ;GO LOG EVENT AND TIME
023172                          LGDVE:
```

```
 5939  023172  012737  020036  007246          MOV     #SDVE,TEMP1     ;SET UP MSG. TO PRINT
 5940  023200  012737  000010  007244          MOV     #DER,TEMP       ;SET UP EVENT TYPE
 5941  023206  000474                          BR      LOGS3           ;GO LOG EVENT AND TIME
 5942
 5943  023210                          LOGDVI:
 5944  023210  012737  020060  007246          MOV     #SDVI,TEMP1     ;SET UP MSG. TO PRINT
 5945  023216  012737  000012  007244          MOV     #DVI,TEMP       ;SET UP EVENT TYPE
 5946  023224  113737  007306  007250          MOVB    MODTYP,TEMP2
 5947  023232  113737  007310  007251          MOVB    MLTYP,TEMP2+1
 5948  023240  013737  007316  007252          MOV     RPASS,TEMP3
 5949  023246  013737  007314  007254          MOV     PARAM,TEMP4     ;SET UP EVNT ENTRIES
 5950  023254  000451                          BR      LOGS3           ;GO LOG EVENT AND TIME
 5951
 5952  023256                          LOGCMP:
 5953  023256  012737  020047  007246          MOV     #SCM,TEMP1      ;SET UP MSG. TO PRINT
 5954  023264  012737  000014  007244          MOV     #DCK,TEMP       ;SET UP EVENT TYPE
 5955  023272  000442                          BR      LOGS3
 5956  023274                          LOGCML:
 5957  023274  012737  020071  007246          MOV     #SCML,TEMP1
 5958  023302  012737  000020  007244          MOV     #DLE,TEMP       ;SET UP MSG. AND TYPE
 5959  023310  000433                          BR      LOGS3           ;GO LOG EVENT AND TIME
 5960  023312                          LOGCMD:
 5961  023312  012737  020102  007246          MOV     #SCMD,TEMP1
 5962  023320  012737  000022  007244          MOV     #DDE,TEMP
 5963  023326  000424                          BR      LOGS3           ;GO LOG MSG TYPE AND TIME
 5964  023330                          LOGEOP:
 5965  023330  012737  020113  007246          MOV     #SEOP,TEMP1
 5966  023336  012737  000024  007244          MOV     #EOP,TEMP
 5967  023344  000415                          BR      LOGS3           ;GO LOG MSG TYPE AND TIME
 5968
 5969  023346  013746  007216          LOGS1:  MOV     ERRCNT,-(SP)    ;SAVE CURRENT ERROR COUNT
 5970  023352  004737  044474                  JSR     PC,DVMODS       ; GO GET MODEM STATUS
 5971  023356  012604                          MOV     (SP)+,R4        ;GET SAVED ERRCNT VALUE
 5972  023360  020437  007216                  CMP     R4,ERRCNT       ;WHERE ANY ERRORS FOUND
 5973  023364  001402                          BEQ     1$              ; BR IF NONE
 5974  023366  000137  023602                  JMP     LOGEX           ; ELSE, LEAVE WITHOUT LOGGING ANYTHING
 5975                                                                  ;   BUT THE DEVICE ERROR FROM 'DVMODS'
 5976  023372  013737  010274  007254  1$:     MOV     MODS,TEMP4      ;AND PUT IT IN TEMP4
 5977
 5978  023400                          LOGS3:
 5979  023400  022737  000006  007244          CMP     #RXC,TEMP
 5980  023406  001434                          BEQ     LOGS5           ;IF RXC DONT PRINT
 5981  023410  032737  000001  007314          BIT     #STATB,PARAM
 5982  023416  001430                          BEQ     LOGS5           ;IF NO STATUS SELECTED
 5983                                                                  ;GO TO 5
 5984
 5985  023420  022737  000010  007210          CMP     #10,LNCNT       ;HAVE WE DONE 10?
 5986  023426  001012                          BNE     LOGS4           ;IF NOT GO TO 4
 5987  023430  005037  007210                  CLR     LNCNT           ;ESLE CLEAR IT
 5988
 5989  023434                          PRINTF  #CR                     ;ELSE PRINT CR
  (7)  023434  012746  020000                                                    MOV     #CR,-(SP)
  (6)  023440  012746  000001                                                    MOV     #1,-(SP)
  (3)  023444  010600                                                            MOV     SP,R0
  (4)  023446  104417                                                            TRAP    C$PNTF
  (4)  023450  062706  000004                                                    ADD     #4,SP
```

```
 5990  023454
 5991  023454  005237  007210                LOGS4:
 5992  023460                                       INC     LNCNT           ;INC COUNTER OF # OF AWAKE MSGS
  (7)  023460  013746  007246                         PRINTF  TEMP1           ;PRINT OPERATOR AWAKE MSG.
  (6)  023464  012746  000001                                                                 MOV     TEMP1,-(SP)
  (3)  023470  010600                                                                         MOV     #1,-(SP)
  (4)  023472  104417                                                                         MOV     SP,R0
  (4)  023474  062706  000004                                                                 TRAP    C$PNTF
 5993  023500  010346                LOGS5:  MOV     R3,-(SP)        ;SAVE R3 ON THE STACK   ADD     #4,SP
 5994  023502  013703  007366                MOV     EVTPTR,R3
 5995  023506  113723  007244                MOVB    TEMP,(R3)+      ;LOG EVENT
 5996  023512  013737  007346  007244        MOV     CLKHZ,TEMP
 5997  023520  163737  007356  007244        SUB     TIMTCK,TEMP
 5998  023526  113723  007244                MOVB    TEMP,(R3)+      ;LOG TIME SINCE START
 5999  023532  113723  007354                MOVB    TIMSEC,(R3)+
 6000  023536  113723  007352                MOVB    TIMMIN,(R3)+    ;TICKS,SECS AND MINS.
 6001  023542  013723  007250                MOV     TEMP2,(R3)+     ;LOG EVNT ENTRY 3
 6002  023546  013723  007252                MOV     TEMP3,(R3)+     ;LOG EVNT ENTRY 4
 6003  023552  013723  007254                MOV     TEMP4,(R3)+     ;LOG EVNT ENTRY 5
 6004  023556  020327  010272                CMP     R3,#EVTEND
 6005  023562  103404                        BLO     LOGS2           ;IF EVENT LOG FULL GO
 6006                                                                 ;CONTINUE;ELSE GO TO 2
 6007  023564  012713  177777                MOV     #-1,(R3)        ;LOG A TABLE END
 6008  023570  012703  007370                MOV     #EVTLOG,R3      ;PUT R3 TO START OF TABLE
 6009  023574  010337  007366        LOGS2:  MOV     R3,EVTPTR       ;RESTORE POINTER
 6010  023600  012603                        MOV     (SP)+,R3        ;RESTORE R3
 6011  023602  000207                LOGEX:  RTS     PC
 6012
 6013
```

```
 6015                                   .SBTTL  REPORT BASE TABLE OR EVENT LOG
 6016                                           ;;RPT> LOG
 6017                                           ;;     BASE/ERROR
 6018                                           ;;     BASE/FULL
 6019                                           ;;     BASE/OFFSET=NN
 6020                                           ;;     HELP
 6021                                           ;;     EXIT
 6022
 6023  023604  010246                  REPORT: MOV     R2,-(SP)        ;SAVE R2,R3,R4 ON THE STACK
 6024  023606  010346                          MOV     R3,-(SP)
 6025  023610  010446                          MOV     R4,-(SP)
 6026
 6027                                           ;PRINT HELP MESSAGE
 6028  023612                                   PRINTF  #RHLP0          ;BASIC HELP MESSAGE
   (7) 023612  012746  013632                                                    MOV     #RHLP0,-(SP)
   (6) 023616  012746  000001                                                    MOV     #1,-(SP)
   (3) 023622  010600                                                            MOV     SP,R0
   (4) 023624  104417                                                            TRAP    C$PNTF
   (4) 023626  062706  000004                                                    ADD     #4,SP
 6029
 6030  023632  105037  003377          GETRCL: CLRB    P$GDBD          ;INIT GOOD/BAD FLAG  -1=BAD INPUT
 6031  023636  105037  003376                  CLRB    P$NMUF          ;INIT MORE COMMAND LINE INPUT NEEDED
 6032
 6033                                           ;PRINT PROMPT 'RPT>'
 6034  023642                                   GMANID  CLI$RP,CMDBUF,A,-1,1,72.,NO
   (3) 023642  104443                                                            TRAP    C$GMAN
   (3) 023644  000406                                                            BR      10000$
   (4) 023646  003060                                                            .WORD   CMDBUF
   (5) 023650  000142                                                            .WORD   T$CODE
   (5) 023652  012316                                                            .WORD   CLI$RP
   (5) 023654  177777                                                            .WORD   -1
   (5) 023656  000001                                                            .WORD   T$LOLIM
   (5) 023660  000110                                                            .WORD   T$HILIM
   (3) 023662                                                           10000$:
 6035  023662  012737  003060  003362          MOV     #CMDBUF,P$BUFA  ;INPUT BUFFER
 6036  023670  012737  024024  003364          MOV     #CLIRT,P$TREE   ;REPORT CLI TREE
 6037  023676  012737  024222  003366          MOV     #CLIRAC,P$ACT   ;ACTION ROUTINES
 6038  023704  005037  003204                  CLR     QUALFG
 6039  023710  004737  027152                  JSR     PC,P$TRV        ;GO PARSE COMMAND LINE
 6040  023714  105737  003377                  TSTB    P$GDBD          ;COMMAND OK ?
 6041  023720  001412                          BEQ     1$              ;YES,BRANCH
 6042  023722                                   PRINTF  #CLIERM         ;PRINT INVALID INPUT MESSAGE
   (7) 023722  012746  012324                                                    MOV     #CLIERM,-(SP)
   (6) 023726  012746  000001                                                    MOV     #1,-(SP)
   (3) 023732  010600                                                            MOV     SP,R0
   (4) 023734  104417                                                            TRAP    C$PNTF
   (4) 023736  062706  000004                                                    ADD     #4,SP
 6043  023742  000137  023632                  JMP     GETRCL          ;TRY AGAIN
 6044
 6045  023746  105737  003376          1$:     TSTB    P$NMUF          ;MORE COMMAND NEEDED ?
 6046  023752  001412                          BEQ     10$             ;NO,BRANCH
 6047  023754                                   PRINTF  #CLINUF         ;INCOMPLETE MESSAGE
   (7) 023754  012746  012354                                                    MOV     #CLINUF,-(SP)
   (6) 023760  012746  000001                                                    MOV     #1,-(SP)
   (3) 023764  010600                                                            MOV     SP,R0
   (4) 023766  104417                                                            TRAP    C$PNTF
```

```
                                                                                    ADD     #4,SP
  (4)  023770  062706  000004
6048   023774  000137  023632                    JMP     GETRCL                  ;TRY AGAIN
6049
6050   024000  023727  003202  000002  10$:      CMP     KEYWD1,#RPEXT           ;EXIT COMMAND ?
6051   024006  001402                            BEQ     20$                     ;YES,BRANCH
6052   024010  000137  023632                    JMP     GETRCL                  ;GET ANOTHER COMMAND
6053   024014  012604                    20$:    MOV     (SP)+,R4                ;RESTORE R4
6054   024016  012603                            MOV     (SP)+,R3                ;RESTORE R3
6055   024020  012602                            MOV     (SP)+,R2                ;RESTORE R2
6056   024022  000207                            RTS     PC                      ;RETURN
```

```
 6058                                .SBTTL  COMMAND LINE PARSING TREE FOR REPORT
 6059  024024                        CLIRT:  CLI CLISPA,0,R10$                 ;SKIP SPACES IN COMMAND LINE
 6060  024030                        R10$:   CLI <'?>,RPHLP,R11$               ;IF INPUT = ? THEN PRINT HELP MESSAGE
 6061  024034                                CLI CLIEXI,0                      ;;AND EXIT PARSER
 6062  024036                        R11$:   CLI CLISTR,RPHLP,R12$,<'HELP'>    ;IF INPUT = 'HELP' THEN PRINT HELP
 6063  024052                                CLI CLIEXI,0                      ;;MESSAGE AND EXIT PARSER
 6064  024054                        R12$:   CLI CLISTR,RPEXT,R13$,<'EXIT'>    ;IF INPUT = 'EXIT' THEN SET KEYWORD =
 6065  024070                                CLI CLIEXI,0                      ;;RPEXT AND EXIT PARSER
 6066  024072                        R13$:   CLI CLISTR,RPLOG,R14$,<'LOG'>     ;IF INPUT = 'LOG' THEN GO PRINT EVENT
 6067  024104                                CLI CLIEXI,0                      ;;LOG AND EXIT PARSER
 6068  024106                        R14$:   CLI CLISTR,RNOTNF,R30$,<'BASE'>;IF INPUT = 'BASE' THEN MORE COMMAND
 6069  024122                                CLI CLIBR,0,R15$                  ;;LINE IS NEEDED
 6070  024126                        R15$:   CLI <'/>,RNOTNF,R125$             ;IF INPUT = '/' THEN LOOK FOR MORE
 6071  024132                                CLI CLISTR,RPSWE,R16$,<'ERROR'>   ;IF INPUT = 'ERROR' THEN GO PRINT
 6072  024146                                CLI CLIEXI,0                      ;;ERROR INFORMATION
 6073  024150                        R16$:   CLI CLISTR,RPSWF,R17$,<'FULL'>    ;IF INPUT = 'FULL' THEN GO PRINT
 6074  024164                                CLI CLIEXI,0                      ;;ENTIRE BASE TABLE
 6075  024166                        R17$:   CLI CLISTR,RNOTNF,R30$,<'OFFSET'>;IF INPUT = 'OFFSET' THEN LOOK FOR
 6076  024204                                CLI <'=>,0,R30$                   ;;'='
 6077  024210                                CLI CLIOCT,RPSWO,R30$             ;IF INPUT = OCTAL VALUE THEN GO
 6078  024214                                CLI CLIEXI,0                      ;;PRINT SINGLE BASE TABLE ITEM
 6079  024216                        R30$:   CLI CLIERR,0
 6080  024220                        R125$:  CLI CLIEXI,0
```

```
                                        .SBTTL  CLI ACTION DISPATCHER AND ROUTINES
024222  006302                  CLIRAC: ASL     R2              ;SET UP INDEX
024224  016202  024240                  MOV     10$(R2),R2      ;
024230  062702  024240                  ADD     #10$,R2         ;
024234  004712                          JSR     PC,(R2)         ;GO DO ACTION
024236  000207                          RTS     PC              ;RETURN
024240  000026                  10$:    .WORD   ACTRNL-10$      ;NULL
024242  000030                          .WORD   ACTRHL-10$      ;HELP ROUTINE
024244  000074                          .WORD   ACTREX-10$      ;EXIT ROUTINE
024246  000104                          .WORD   ACTRLG-10$      ;REPORT EVENT LOG ROUTINE
024250  000120                          .WORD   ACTSWE-10$      ;REPORT ERRORS ROUTINE
024252  000200                          .WORD   ACTSWF-10$      ;REPORT ENTIRE BASE TABLE
024254  000244                          .WORD   ACTSWO-10$      ;REPORT SINGLE BASE ADDRESS
024256  000020                          .WORD   ACTRNF-10$      ;MORE COMMAND NEEDED

                                        ;MORE COMMAND NEEDED
024260  112737  177777  003376  ACTRNF: MOVB    #-1,PSNNUF      ;MORE COMMAND NEEDED
024266  000207                  ACTRNL: RTS     PC              ;NULL

                                        ;PRINT HELP MESSAGE
024270  012702  003230          ACTRHL: MOV     #RHLPTB,R2      ;INDEX FOR HELP MESSAGES
024274                          1$:     PRINTF  #HLPF,(R2)+     ;PRINT IT
                                                                        MOV     (R2)+,-(SP)
                                                                        MOV     #HLPF,-(SP)
                                                                        MOV     #2,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTF
                                                                        ADD     #6,SP
024316  020227  003246                  CMP     R2,#RHLPEN      ;LAST MESSAGE ?
024322  001364                          BNE     1$              ;NO,BRANCH
024324  012737  000001  003202          MOV     #RPHLP,KEYWD1   ;SET KEYWORD
024332  000207                          RTS     PC              ;RETURN

                                        ;EXIT REPORT LEVEL
024334  012737  000002  003202  ACTREX: MOV     #RPEXT,KEYWD1   ;SET KEYWORD AND RETURN
024342  000207                          RTS     PC

                                        ;PRINT ERROR LOG
024344  004737  025104          ACTRLG: JSR     PC,REPLOG       ;GO PRINT EVENT LOG
024350  012737  000003  003202          MOV     #RPLOG,KEYWD1   ;SET KEYWORD
024356  000207                          RTS     PC              ;RETURN

                                        ;PRINT ONLY ERROR LOCATIONS
024360  012737  003250  007266  ACTSWE: MOV     #DMCIND,INDEX   ;SETUP KMS11 MESSAGES
024366  062737  000006  007266          ADD     #6,INDEX        ;POINT TO CORRECT MESSAGE
024374  012737  003300  007270          MOV     #DMCEND,INDEXE  ;LAST KMS11 ADDRESS
024402  012737  017370  007272          MOV     #BASE,BEND      ;SET UP LAST ADDRESS
024410  062737  000012  007272          ADD     #12,BEND        ;;TO BE PRINTED
024416  012737  017370  007274          MOV     #BASE,BDATA     ;BASE TABLE START ADDRESS
024424  062737  000003  007274          ADD     #3,BDATA        ;ERROR START ADDRESS
024432  004737  024646                  JSR     PC,RPBASE       ;GO PRINT DATA
024436  000207                          RTS     PC              ;RETURN

                                        ;PRINT FULL BASE TABLE
024440  012737  003250  007266  ACTSWF: MOV     #DMCIND,INDEX   ;SETUP KMS11 MESSAGES
024446  012737  003300  007270          MOV     #DMCEND,INDEXE  ;LAST MESSAGE
```

```
 6132  024454  012737  017370  007272          MOV     #BASE,BEND      ;TABLE START ADDRESS
 6133  024462  062737  000377  007272          ADD     #377,BEND       ;PRINT 256. BYTES OF DATA
 6134  024470  012737  017370  007274          MOV     #BASE,BDATA     ;FIRST ADDRESS TO PRINT
 6135  024476  004737  024646          20$:    JSR     PC,RPBASE       ;GO PRINT DATA
 6136  024502  000207                          RTS     PC              ;RETURN
 6137
 6138                                          ;PRINT SINGLE TABLE LOCATION
 6139  024504  105037  003376          ACTSWO: CLRB    P$NNUF          ;INIT NOT ENOUGH FLAG
 6140  024510  012737  000377  007272          MOV     #377,BEND       ;BASE TABLE FOR KMS11 = 256 BYTES
 6141  024516  023737  003372  007272          CMP     P$NUM,BEND      ;KMS11 = 256 BYTES
 6142  024524  101416                          BLOS    10$             ;YES,BRANCH
 6143  024526                                  PRINTF  #RPTIV,P$NUM    ;PRINT ILLEGAL VALUE
   (8)  024526  013746  003372                                                  MOV     P$NUM,-(SP)
   (7)  024532  012746  014202                                                  MOV     #RPTIV,-(SP)
   (6)  024536  012746  000002                                                  MOV     #2,-(SP)
   (3)  024542  010600                                                          MOV     SP,R0
   (4)  024544  104417                                                          TRAP    C$PNTF
   (4)  024546  062706  000006                                                  ADD     #6,SP
 6144  024552  112737  177777  003377          MOVB    #-1,P$GDBD      ;SET BAD DATA
 6145  024560  000431                          BR      30$             ;RETURN
 6146  024562  013701  003372          10$:    MOV     P$NUM,R1        ;OFFSET VALUE
 6147  024566  006301                          ASL     R1              ;MULTIPLY BY 2
 6148  024570  012737  003250  007266          MOV     #DMCIND,INDEX   ;KMS11 MESSAGES
 6149  024576  060137  007266                  ADD     R1,INDEX        ;GET RIGHT MESSAGE
 6150  024602  012737  003300  007270          MOV     #DMCEND,INDEXE  ;LAST KMS11 MESSAGE
 6151  024610  012737  017370  007272          MOV     #BASE,BEND      ;TABLE ADDRESS
 6152  024616  063737  003372  007272          ADD     P$NUM,BEND      ;LAST ADDRESS
 6153  024624  012737  017370  007274          MOV     #BASE,BDATA     ;BASE ADDRESS
 6154  024632  063737  003372  007274          ADD     P$NUM,BDATA     ;ADD OFFSET
 6155  024640  004737  024646          20$:    JSR     PC,RPBASE       ;GO PRINT SINGLE LOCATION
 6156  024644  000207                  30$:    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 KMS11. 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 KMS11 DATA
                                ;                       DESCRIPTION MESSAGES.
                                ;               INDEXE - CONTAINS POINTER TO LAST KMS11
                                ;                       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.


024646  010146                  RPBASE: MOV     R1,-(SP)        ;SAVE R1
024650  010246                          MOV     R2,-(SP)        ;SAVE R2
024652                                  PRINTF  #BTHEAD         ;PRINT BRIEF HEADER MESSAGE
  (7)   024652  012746  025006                                                  MOV     #BTHEAD,-(SP)
  (6)   024656  012746  000001                                                  MOV     #1,-(SP)
  (3)   024662  010600                                                          MOV     SP,R0
  (4)   024664  104417                                                          TRAP    C$PNTF
  (4)   024666  062706  000004                                                  ADD     #4,SP
024672  013702  007266                          MOV     INDEX,R2        ;POINTER TO MESSAGES
024676  013701  007274                          MOV     BDATA,R1        ;ADDRESS OF DATA
024702  010137  007252                  10$:    MOV     R1,TEMP3        ;SAVE CURRENT ADDRESS OF DATA
024706  112137  007246                          MOVB    (R1)+,TEMP1     ;READ DATA
024712  020237  007270                          CMP     R2,INDEXE       ;END OF MESSAGES?
024716  002402                              BLT     20$             ;NO,BRANCH
024720  013702  007270                          MOV     INDEXE,R2       ;'SEE MANUAL' MESSAGE
024724  012237  007250                  20$:    MOV     (R2)+,TEMP2     ;READ MESSAGE ADDRESS
024730                                  PRINTF  #DMFMT,TEMP3,<B,TEMP1>,TEMP2    ;PRINT DATA AND MESSAGE
 (10)   024730  013746  007250                                                  MOV     TEMP2,-(SP)
  (9)   024734  005046                                                          CLR     -(SP)
  (9)   024736  153716  007246                                                  BISB    TEMP1,(SP)
  (8)   024742  013746  007252                                                  MOV     TEMP3,-(SP)
  (7)   024746  012746  025057                                                  MOV     #DMFMT,-(SP)
  (6)   024752  012746  000004                                                  MOV     #4,-(SP)
  (3)   024756  010600                                                          MOV     SP,R0
  (4)   024760  104417                                                          TRAP    C$PNTF
  (4)   024762  062706  000012                                                  ADD     #12,SP
024766  020137  007272                          CMP     R1,BEND         ;LAST ADDRESS ?
024772  101743                              BLOS    10$             ;NO,BRANCH
024774  105037  003376                          CLRB    P$NNUF          ;CLEAR ENOUGH FLAG
025000  012602                              MOV     (SP)+,R2        ;RESTORE R2
025002  012601                              MOV     (SP)+,R1        ;RESTORE R1
025004  000207                              RTS     PC              ;RETURN

025006  047045  040445  042101  BTHEAD: .ASCIZ /%N%AADDRESS%S2%ACONTENTS%S6%ADESCRIPTION/
025014  051104  051505  022523
025022  031123  040445  047503
025030  052116  047105  051524
```

```
                025036  051445  022466  042101
                025044  051505  051103  050111
                025052  044524  047117     000
 6197           025057     045  022516  032523  DMFMT:  .ASCIZ /%N%S1%06%S5%03%S5%T/
                025064  047445  022466  032523
                025072  047445  022463  032523
                025100  052045     000
 6198           025104                          .EVEN
```

```
6200                                    .SBTTL  PRINT EVENT LOG
6201                                    ;PRINT THE EVENT LOG
6202  025104  010246                    REPLOG: MOV     R2,-(SP)        ;SAVE R2
6203  025106  010346                            MOV     R3,-(SP)        ;SAVE R3
6204  025110  010446                            MOV     R4,-(SP)        ;SAVE R4
6205  025112  013702  007366                    MOV     EVTPTR,R2       ;MAKE R2 A POINTER TO EVENT TABLE
6206  025116  023727  007370  177777            CMP     EVTLOG,#-1      ;SEE IF EVENT TABLE IS EMPTY
6207  025124  001034                            BNE     RPT0            ;BR IF NO
6208  025126                                    PRINTS  #NULEVT         ;IF EMPTY TELL OPRERATOR.
  (7) 025126  012746  016073                                                    MOV     #NULEVT,-(SP)
  (6) 025132  012746  000001                                                    MOV     #1,-(SP)
  (3) 025136  010600                                                            MOV     SP,R0
  (4) 025140  104416                                                            TRAP    C$PNTS
  (4) 025142  062706  000004                                                    ADD     #4,SP
6209  025146  000137  025742                    JMP     ENDEVT          ;AND END
6210
6211  025152  162702  000012          RPT:      SUB     #12,R2          ;NOW POINT BACK TO TOP OF ENTRY U
6212                                            ;JUST PRINTED
6213
6214  025156  020227  007370                    CMP     R2,#EVTLOG      ;POINTING TO TOP OF EVNT LOG QUEUE?
6215  025162  001010                            BNE     RPT1            ; BR IF NO
6216  025164  012702  010272                    MOV     #EVTEND,R2      ;SET R2 TO POINT TO BOTTOM OF LOG
6217  025170  026227  177776  177777            CMP     -2(R2),#-1
6218  025176  001007                            BNE     RPT0            ;IF END OF LOG IS NOT EMPTY
6219  025200  000137  025742                    JMP     ENDEVT          ;CONTINUE...ELSE EXIT
6220
6221  025204  020237  007366          RPT1:     CMP     R2,EVTPTR       ;ARE WE BACK TO POINTER?
6222  025210  001002                            BNE     RPT0            ;IF NOT CONTINUE
6223  025212  000137  025742                    JMP     ENDEVT          ;IF SO EXIT....
6224
6225  025216  162702  000012          RPT0:     SUB     #12,R2          ;POINT R2 TO START OF ENTRY
6226  025222                          RPTAA:    PRINTS  #EVTF0          ;PRINT EVENT ENTRY HEADER
  (7) 025222  012746  016133                                                    MOV     #EVTF0,-(SP)
  (6) 025226  012746  000001                                                    MOV     #1,-(SP)
  (3) 025232  010600                                                            MOV     SP,R0
  (4) 025234  104416                                                            TRAP    C$PNTS
  (4) 025236  062706  000004                                                    ADD     #4,SP
6227  025242  112203                            MOVB    (R2)+,R3        ;PUT EVENT TYPE INTO R3
6228  025244  112237  010432                    MOVB    (R2)+,EVTTCK
6229  025250  112237  010426                    MOVB    (R2)+,EVTSEC    ;PUT EVENT TIME (TICKS,SECS,MINS IN TEMP LOC.S)
6230  025254  112237  010430                    MOVB    (R2)+,EVTMIN
6231  025260                                    PRINTS  #EVTF1,EVTMIN,EVTSEC,EVTTCK,EVTLST(R3)  ;PRINT EVENT TIME AND DESCRIPT.
 (11) 025260  016346  010332                                                    MOV     EVTLST(R3),-(SP)
 (10) 025264  013746  010432                                                    MOV     EVTTCK,-(SP)
  (9) 025270  013746  010426                                                    MOV     EVTSEC,-(SP)
  (8) 025274  013746  010430                                                    MOV     EVTMIN,-(SP)
  (7) 025300  012746  016226                                                    MOV     #EVTF1,-(SP)
  (6) 025304  012746  000005                                                    MOV     #5,-(SP)
  (3) 025310  010600                                                            MOV     SP,R0
  (4) 025312  104416                                                            TRAP    C$PNTS
  (4) 025314  062706  000014                                                    ADD     #14,SP
6232  025320  000173  010442                    JMP     @RPTDSP(R3)     ;DISPATCH TO DECODING SECTION FOR SPECIFIC TYPE
6233
6234  025324  012237  010434          RPTTXa:   MOV     (R2)+,EVTADD    ;STORE MESSAGE ADDRESS FOR PRINTING
6235  025330  012237  010436                    MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT FOR PRINTING
6236  025334  012203                            MOV     (R2)+,R3        ;STORE MODEM STATUS FOR PRINTING
```

```
 6237  025336                                  PRINTS  #EVTF2,EVTADD,EVTBCT    ;PRINT ADDR,BYTE CNT
                                                                                        MOV     EVTBCT,-(SP)
                                                                                        MOV     EVTADD,-(SP)
                                                                                        MOV     #EVTF2,-(SP)
                                                                                        MOV     #3,-(SP)
                                                                                        MOV     SP,R0
                                                                                        TRAP    C$PNTS
                                                                                        ADD     #10,SP
 6238  025366  004737  025752                  JSR     PC,RPTMSB       ;GO PRINT MODEM STATUS
 6239  025372  000137  025152                  JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
 6240
 6241  025376  012237  010440          RPTDER: MOV     (R2)+,EVTTMP    ;GET ADDRESS OF DEVICE INFO MESSAGE
 6242  025402  012237  010472                  MOV     (R2)+,DEV1      ;STORE DEVICE REG CONTENTS FOR PRINTING
 6243  025406  012237  010474                  MOV     (R2)+,DEV2
 6244  025412                                  PRINTS  #EVTF3,EVTTMP   ;PRINT DEVICE REG CONTENTS.
                                                                                        MOV     EVTTMP,-(SP)
                                                                                        MOV     #EVTF3,-(SP)
                                                                                        MOV     #2,-(SP)
                                                                                        MOV     SP,R0
                                                                                        TRAP    C$PNTS
                                                                                        ADD     #6,SP
 6245  025436                                  PRINTS  #EVTF3C,DEV1,DEV2
                                                                                        MOV     DEV2,-(SP)
                                                                                        MOV     DEV1,-(SP)
                                                                                        MOV     #EVTF3C,-(SP)
                                                                                        MOV     #3,-(SP)
                                                                                        MOV     SP,R0
                                                                                        TRAP    C$PNTS
                                                                                        ADD     #10,SP
 6246  025466  000137  025152                  JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
 6247
 6248  025472  005037  010472          RPTDVI: CLR     DEV1
 6249  025476  005037  010474                  CLR     DEV2            ;CLEAR UPPER BYTES OF DEV1 & DEV2 BEFORE USE
 6250  025502  112237  010472                  MOVB    (R2)+,DEV1      ;STORE SETUP OPERATION PARAMETERS FOR PRINTING
 6251  025506  112237  010474                  MOVB    (R2)+,DEV2
 6252  025512  012237  010476                  MOV     (R2)+,DEV3
 6253  025516  012237  010500                  MOV     (R2)+,DEV4
 6254  025522  010246                          MOV     R2,-(SP)        ;SAVE R2 ON THE STACK
 6255  025524  004737  026650                  JSR     PC,SHWOP        ;GO PRINT MODE, MAINT-LOOP TYPE, PARAMTERS.
 6256  025530  012602                          MOV     (SP)+,R2        ;RESTORE R2
 6257  025532  000137  025152                  JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
 6258
 6259  025536                          ;;REPORT END OF PASS OR ^C ABORT
 6260  025536  012237  010434          RPTABO:
 6261  025542  012237  010436          RPTEOP: MOV     (R2)+,EVTADD
 6262  025546  012237  010440                  MOV     (R2)+,EVTBCT
 6263  025552                                  MOV     (R2)+,EVTTMP
                                                PRINTS  #EVTF4B,EVTADD,EVTBCT,EVTTMP    ;PRINT ADDR,RXBYTES,CMPBYTES.
                                                                                        MOV     EVTTMP,-(SP)
                                                                                        MOV     EVTBCT,-(SP)
                                                                                        MOV     EVTADD,-(SP)
                                                                                        MOV     #EVTF4B,-(SP)
                                                                                        MOV     #4,-(SP)
                                                                                        MOV     SP,R0
                                                                                        TRAP    C$PNTS
                                                                                        ADD     #12,SP
 6264
```

```
 6265  02560o  000137  025152                  JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY
 6266
 6267
 6268  025612  012237  010434        RPTDDE: MOV     (R2)+,EVTADD    ;STORE MESSAGE ADDRESS FOR PRINTING
 6269  025616  012237  010436                MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT FOR PRINTING
 6270  025622  012237  010440                MOV     (R2)+,EVTTMP    ;STORE TOTAL # OF CMP ERRORS
 6271  025626                                PRINTS  #EVTF4,EVTADD,EVTBCT,EVTTMP     ;PRINT ADDR, BYTE CNT, # CMP ERRS
  (10) 025626  013746  010440                                                            MOV     EVTTMP,-(SP)
  (9)  025632  013746  010436                                                            MOV     EVTBCT,-(SP)
  (8)  025636  013746  010434                                                            MOV     EVTADD,-(SP)
  (7)  025642  012746  016400                                                            MOV     #EVTF4,-(SP)
  (6)  025646  012746  000004                                                            MOV     #4,-(SP)
  (3)  025652  010600                                                                    MOV     SP,R0
  (4)  025654  104416                                                                    TRAP    C$PNTS
  (4)  025656  062706  000012                                                            ADD     #12,SP
 6272  025662  000137  025152                  JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY
 6273
 6274  025666                        RPTDLE:
 6275  025666  012237  010434        RPTDCK: MOV     (R2)+,EVTADD    ;STORE MSG ADDR FOR PRINT
 6276  025672  012237  010436                MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT
 6277  025676  012237  010440                MOV     (R2)+,EVTTMP    ;STORE BYTE COUNT COMP
 6278  025702                                PRINTS  #EVTF4A,EVTADD,EVTBCT,EVTTMP    ;PRINT ADDR,RXBYTES,CMPBYTES.
  (10) 025702  013746  010440                                                            MOV     EVTTMP,-(SP)
  (9)  025706  013746  010436                                                            MOV     EVTBCT,-(SP)
  (8)  025712  013746  010434                                                            MOV     EVTADD,-(SP)
  (7)  025716  012746  016502                                                            MOV     #EVTF4A,-(SP)
  (6)  025722  012746  000004                                                            MOV     #4,-(SP)
  (3)  025726  010600                                                                    MOV     SP,R0
  (4)  025730  104416                                                                    TRAP    C$PNTS
  (4)  025732  062706  000012                                                            ADD     #12,SP
 6279
 6280  025736  000137  025152                  JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY
 6281
 6282  025742  012604                ENDEVT: MOV     (SP)+,R4        ;RESTORE R4,R3,R2
 6283  025744  012603                        MOV     (SP)+,R3
 6284  025746  012602                        MOV     (SP)+,R2
 6285  025750  000207                        RTS     PC              ;RETURN TO CALLING ROUTINE
 6286
 6287
 6288                                ;REPORT MODEM STATUS SUBROUTINE
 6289                                ;       PART OF STATISICAL REPORTING (DUMPING EVENT LOG)
 6290
 6291  025752                        RPTMSB: PRINTS  #EVMOHD                 ;PRINT MODEM STATUS HEADER
  (7)  025752  012746  016766                                                            MOV     #EVMOHD,-(SP)
  (6)  025756  012746  000001                                                            MOV     #1,-(SP)
  (3)  025762  010600                                                                    MOV     SP,R0
  (4)  025764  104416                                                                    TRAP    C$PNTS
  (4)  025766  062706  000004                                                            ADD     #4,SP
 6292  025772  012704  010276                MOV     #MOBITS,R4      ;MAKE R4 A POINTER TO MODEM SIG. BIT DEF. TABLE
 6293  025776  012705  010314                MOV     #MOMSGS,R5      ;MAKE R5 A POINTER TO MODEM MSG. POSITION TABLE
 6294  026002  005714                6$:     TST     (R4)            ;SEE IF BIT AVAIABLE FROM DEVICE
 6295  026004  001004                        BNE     7$              ;BR IF THAT MODEM SIG. AVAIABLE
 6296  026006  112735  000130                MOVB    #'X,@(R5)+      ;ELSE PUT "X" IN REPORT IF SIGNAL NOT AVAILABLE
 6297  026012  005724                        TST     (R4)+           ;BUMP R4 TO POINT TO NEXT BIT DEFINITION
 6298  026014  000407                        BR      9$              ;GO SEE IF CHECKED ALL MODEM SIGNALS
 6299  026016  032403                7$:     BIT     (R4)+,R3        ;IF THERE, SEE IF THAT BIT IN DEVICE'S ENTRY=1
```

```
 6300  026020  001403                        BEQ     8$              ;BR IF BIT (SIGNAL) VALUE =0
 6301  026022  112735  000061                MOVB    #'1,@(R5)+      ;IF=1, PUT "1" IN REPORT MESSAGE
 6302  026026  000402                        BR      9$              ;GO SEE IF ALL MODEM SIGNALS CHECKED
 6303  026030  112735  000060        8$:     MOVB    #'0,@(R5)+      ;IF BIT(SIGNAL)=0, PUT "0" IN REPORT MESSAGE
 6304  026034  020427  010314        9$:     CMP     R4,#MOBITE      ;SEE IF ALL BITS(SIGNALS) CHECKED
 6305  026040  002760                        BLT     6$              ;LOOP UNTIL ALL SIGNALS(BITS) CHECKED
 6306  026042                                PRINTS  #EVMOST         ;THEN PRINT MODEM SIGNAL VALUE MESSAGE
   (7) 026042  012746  017046                                                    MOV     #EVMOST,-(SP)
   (6) 026046  012746  000001                                                    MOV     #1,-(SP)
   (3) 026052  010600                                                            MOV     SP,R0
   (4) 026054  104416                                                            TRAP    C$PNTS
   (4) 026056  062706  000004                                                    ADD     #4,SP
 6307  026062  000207                        RTS     PC              ;RETURN TO EVENT DECODING
 6308
 6309
```

```
                                .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
                                ;
                                ;--
026064  013702  007220          DUMPSR: MOV     STADD,R2        ;SET R2 UP TO STARTING ADDR.
026070  005003                  DUM4:   CLR     R3              ;CLEAR R3
026072                                  PRINTF  #BASM1,R2       ;PRINT ADDRESS
 (8)    026072  010246                                                  MOV     R2,-(SP)
 (7)    026074  012746  016065                                          MOV     #BASM1,-(SP)
 (6)    026100  012746  000002                                          MOV     #2,-(SP)
 (3)    026104  010600                                                  MOV     SP,R0
 (4)    026106  104417                                                  TRAP    C$PNTF
 (4)    026110  062706  000006                                          ADD     #6,SP
026114  005737  007224          DUM3:   TST     BYTBIT          ;IS THIS BYTE OR WORD
026120  001416                          BEQ     DUM1            ;BR IF WORD
026122  112237  007244                  MOVB    (R2)+,TEMP      ;MOV BYTE TO TEMP
026126                                  PRINTF  #BASM3,<B,TEMP> ;PRINT BYTE
 (8)    026126  005046                                                  CLR     -(SP)
 (8)    026130  153716  007244                                          BISB    TEMP,(SP)
 (7)    026134  012746  016047                                          MOV     #BASM3,-(SP)
 (6)    026140  012746  000002                                          MOV     #2,-(SP)
 (3)    026144  010600                                                  MOV     SP,R0
 (4)    026146  104417                                                  TRAP    C$PNTF
 (4)    026150  062706  000006                                          ADD     #6,SP
026154  000411                          BR      DUM2
026156                          DUM1:   PRINTF  #BASM2,(R2)+    ;PRINT WORD
 (8)    026156  012246                                                  MOV     (R2)+,-(SP)
 (7)    026160  012746  016056                                          MOV     #BASM2,-(SP)
 (6)    026164  012746  000002                                          MOV     #2,-(SP)
 (3)    026170  010600                                                  MOV     SP,R0
 (4)    026172  104417                                                  TRAP    C$PNTF
 (4)    026174  062706  000006                                          ADD     #6,SP
026200  020237  007222          DUM2:   CMP     R2,ENADD        ;COMPARE FOR LAST ADD
```

```
 6348  026204  003005                      BGT     DUMEX           ;IF DONE EXIT
 6349  026206  005203                      INC     R3              ;ELSE BUMP R3
 6350  026210  022703  000010              CMP     #8.,R3          ;HAVE WE PRINTED 8 ACCROSS
 6351  026214  001725                      BEQ     DUM4            ;IF SO GO BACK TO 4
 6352  026216  000736                      BR      DUM3            ;ELSE GO BACK AND PRINT ANOTHER
 6353                                                              ;BYTE OR WORD
 6354  026220  000207              DUMEX:  RTS     PC              ;RETURN TO CALLER
 6355
```

```
                                .SBTTL          UPDATE TOTAL CHAR. COUNT SUBROUTINE

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

026222  063737  007230  007240 ADDCC:  ADD     CURCC,TOTCC     ;ADD CURRENT TO TOTAL
026230  022737  001000  007240         CMP     #BUFLIM,TOTCC   ; COMPARE TO 'BUFLIM''
026236  103027                        BHIS    ADDC1           ;IF NOT MORE THEN 'BUFLIM'' EXIT

                                ; PRINT MESSAGE AND TRUNCATE COUNT

026240                          PRINTF  #MSGTRU
(7)  026240  012746  015104                                                     MOV     #MSGTRU,-(SP)
(6)  026244  012746  000001                                                     MOV     #1,-(SP)
(3)  026250  010600                                                             MOV     SP,R0
(4)  026252  104417                                                             TRAP    C$PNTF
(4)  026254  062706  000004                                                     ADD     #4,SP
026260  163737  007230  007240         SUB     CURCC,TOTCC     ;SUB CURRENT FROM TOTAL
026266  012737  001000  007244         MOV     #BUFLIM,TEMP    ;MOV 'BUFLIM'' TO TEMP
026274  163737  007240  007244         SUB     TOTCC,TEMP      ;SUB TOTAL FROM 'BUFLIM''
026302  013737  007244  007230         MOV     TEMP,CURCC      ;AND ESTABLISH NEW CURRENT
026310  063737  007230  007240         ADD     CURCC,TOTCC     ;ADD ''ADJUSTED CURRENT'' TO TOTAL CHAR. CNT.
026316  000207                 ADDC1:  RTS     PC              ;RETURN TO CALLER
```

```
                                .SBTTL          BUILD MESSAGE BUFFERS SUBROUTINE

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       BLDBUF--  BUILD POINTER TABLE AND BUFFERS
                                ;
                                ;       THIS SUBROUTINE ADDS A MESSAGE TC 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 TXB F OR CMPBUF
                                ;       APPROPRIATE POINTERS IN PTRTAB POINTER TABLE
                                ;
                                ; CALLING SEQUENCE:
                                ;       JSR PC,BLDBUF           ;BUILD MESSAGE IN BUFFER AND ADD PTRS.
                                ;--

026320                          BLDBUF:
026320  010246                         MOV     R2,-(SP)        ;SAVE R2 AND R3 ON THE STACK
026322  010346                         MOV     R3,-(SP)
026324  013702  007234                 MOV     CPTR,R2

026330  013722  007236         BLDB1:  MOV     CURADD,(R2)+    ;PUT CURRENT ADD ON POINTER TAB
026334  013722  007230                 MOV     CURCC,(R2)+     ;PUT CURRENT CC ON POINTER TAB
026340  010237  007234                 MOV     R2,CPTR         ;PUT UPDATED R2 BACK TO CURRENT POINT
026344  013702  007226                 MOV     MSGTYP,R2       ;GET MESSAGE TYPE TO USE AS INDEX
026350  006302                         ASL     R2              ;DOUBLE FOR WORD INDEX
026352  013737  007236  007244         MOV     CURADD,TEMP     ;MOVE CURRENT ADD TO TEMP
026360  063737  007230  007244         ADD     CURCC,TEMP      ;ADD CHAR COUNT TO IT TO GET END
026366  013703  007236                 MOV     CURADD,R3       ;SET R3 TO CURRENT START ADD
026372  016237  002150  007250 BLDB2:  MOV     DMSGCT(R2),TEMP2        ;GET BYTE COUNT
026400  016204  002176                 MOV     DMSGAD(R2),R4   ;PUT STARTING FROM ADD IN R4
026404  060437  007250                 ADD     R4,TEMP2        ;ADD IT TO TEMP2 TO GET END OF FROM
026410  112423                 BLDB3:  MOVB    (R4)+,(R3)+     ;MOV BYTE FROM PATTERN TO BUFFER
026412  020337  007244                 CMP     R3,TEMP         ;ALL DONE?
026416  001404                         BEQ     BLDBEX          ;IF SO EXIT
026420  020437  007250                 CMP     R4,TEMP2        ;IS PATTERN COUNT EXPIRED
026424  001762                         BEQ     BLDB2           ;IF SO GO START AGAIN
026426  000770                         BR      BLDB3           ;IF NOT GET ANOTHER BYTE
026430  063737  007230  007236 BLDBEX: ADD     CURCC,CURADD    ;BUMP CURADD
026436  012603                         MOV     (SP)+,R3        ;RESTORE R3 AND R2
026440  012602                         MOV     (SP)+,R2
026442  000207                         RTS     PC              ;RETURN TO CALLER
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 23-59
CZKMUA.P11    30-MAR-82 09:13          CREATE FACSIMILE OF TX BUFFER AND MESSAGE LIST

                                       .SBTTL  CREATE FACSIMILE OF TX BUFFER AND MESSAGE LIST

                                       ;++
                                       ;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 = STORACE 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
```

```
 6503  026444                                  FACSIMILE:
 6504  026444  010146                                MOV     R1,-(SP)                ;SAVE R1
 6505  026446  005001                                CLR     R1                      ;INIT R1
 6506  026450  116161  003400  005400  10$:         MOVB    TXBUF(R1),CMPBUF(R1)    ;COPY TX BUFFER TO EXPECTED BUFFER
 6507  026456  005201                                INC     R1                      ;BUMP INDEX
 6508  026460  020127  001000                        CMP     R1,#BUFLIM              ;ALL DATA COPIED ?
 6509  026464  001371                                BNE     10$                     ;NO,BRANCH
 6510
 6511  026466  012701  000017          20$:         MOV     #MSGLIM,R1              ;MESSAGE LIMIT
 6512  026472  006301                                ASL     R1                      ;MULTIPLY BY 2
 6513  026474  006301                                ASL     R1                      ;MULTIPLY BY 2
 6514  026476  012737  006400  007154                MOV     #PTRTAB,CMPPTR          ;TOP OF POINTER TABLE
 6515  026504  060137  007154                        ADD     R1,CMPPTR               ;START OF EXPECTED POINTER TABLE
 6516  026510  005001                                CLR     R1                      ;INIT R1
 6517
 6518                                                ;SET UP WHILE - DO LOOP
 6519  026512  012737  006400  007152                MOV     #PTRTAB,TXPTR           ;TX POINTER NOW AT TOP OF TABLE
 6520  026520  012737  005400  007162                MOV     #CMPBUF,CCURAD          ;TRANSFER ADDRESS OF 1ST MESSAGE
 6521  026526  062737  000002  007152                ADD     #2,TXPTR                ;BUMP POINTER
 6522  026534  017737  160412  007160                MOV     @TXPTR,CTOTCC           ;BYTE COUNTER 1ST MESSAGE
 6523  026542  005037  007156                        CLR     CMPTOT                  ;INIT EXPECTED MESSAGE COUNT
 6524
 6525                                                ;WHILE TX MESSAGE TOTAL <> EXPECTED MESSAGE TOTAL DO
 6526  026546  023737  007172  007156  30$:         CMP     TXMTOT,CMPTOT           ;ALL MESSAGES COPIED ?
 6527  026554  001430                                BEQ     40$                     ;YES,BRANCH
 6528  026556  013777  007162  160370                MOV     CCURAD,@CMPPTR          ;TRANSFER ADDRESS OF MESSAGE
 6529  026564  062737  000002  007154                ADD     #2,CMPPTR               ;BUMP POINTER
 6530  026572  013777  007160  160354                MOV     CTOTCC,@CMPPTR          ;BYTE COUNT OF MESSAGE
 6531  026600  062737  000004  007152                ADD     #4,TXPTR                ;BUMP TX MESSAGE POINTER
 6532  026606  063737  007160  007162                ADD     CTOTCC,CCURAD           ;CALC. TRANSFER ADDRESS
 6533  026614  017737  160332  007160                MOV     @TXPTR,CTOTCC           ;BYTE COUNT NEXT MESSAGE
 6534  026622  062737  000002  007154                ADD     #2,CMPPTR               ;BUMP POINTER
 6535  026630  005237  007156                        INC     CMPTOT                  ;INCREMENT MESSAGE COUNT
 6536  026634  000744                                BR      30$                     ;DO IT AGAIN
 6537                                                ;END WHILE - DO
 6538  026636  013737  007174  007160  40$:         MOV     TTOTCC,CTOTCC           ;COPY TOTAL CHARACTER COUNT
 6539
 6540                                                ;END ROUTINE
 6541  026644  012601                                MOV     (SP)+,R1                ;RESTORE R1
 6542  026646  000207                                RTS     PC                      ;RETURN
 6543
 6544
```

```
                                        .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
                                        ;--
026650  013702  010472          SHWOP:  MOV     DEV1,R2         ;GET THE MODE TYPE IN R2
026654  006302                          ASL     R2              ;MAKE IT A WORD TABLE OFFSET
026656  016237  003332  007244          MOV     MODES(R2),TEMP  ;GET ADDRESS OF MODE-IN-ASCII
026664  013702  010474                  MOV     DEV2,R2         ;GET MAINTENANCE LOOP TYPE
026670  006302                          ASL     R2
026672  012737  014450  007252          MOV     #LP00,TEMP3     ;LOAD TEMP3 TO POINT TO "/LOOP="
026700  005702                          TST     R2              ;SEE IF /LOOP=XXXXX OR NONE
026702  001003                          BNE     10$             ; BR IF /LOOP= OF SOME KIND
026704  012737  014447  007252          MOV     #LP0,TEMP3      ;IF NO LOOP THEN DON'T PRINT "/LOOP="
026712  016237  003350  007246  10$:    MOV     LOOPS(R2),TEMP1 ;GET ADDRESS OF LOOP-IN-ASCII
026720  013737  010476  007250          MOV     DEV3,TEMP2      ;GET NUMBER OF PASSES
026726                                  PRINTS  #SHF0,TEMP,TEMF3,TEMP1,TEMP2
 (11)   026726  013746  007250                                                  MOV     TEMP2,-(SP)
 (10)   026732  013746  007246                                                  MOV     TEMP1,-(SP)
 (9)    026736  013746  007252                                                  MOV     TEMP3,-(SP)
 (8)    026742  013746  007244                                                  MOV     TEMP,-(SP)
 (7)    026746  012746  015167                                                  MOV     #SHF0,-(SP)
 (6)    026752  012746  000005                                                  MOV     #5,-(SP)
 (3)    026756  010600                                                          MOV     SP,R0
 (4)    026760  104416                                                          TRAP    C$PNTS
 (4)    026762  062706  000014                                                  ADD     #14,SP
026766  005002                          CLR     R2              ;NOW SET UP FOR QUALIFIERS IN ASCII
026770  012737  014527  007244          MOV     #PST,TEMP
026776  032737  000001  010500          BIT     #STATB,DEV4     ;SEE IF /STATUS OR /NOSTATUS
027004  001003                          BNE     1$              ;BR IF /STATUS
027006  012737  014525  007244          MOV     #PNST,TEMP
027014  012737  014540  007246  1$:     MOV     #PCK,TEMP1
027022  032737  000002  010500          BIT     #DATCKB,DEV4    ;SEE IF /CHECK OR /NOCHECK
027030  001003                          BNE     2$              ;BR IF /CHECK
027032  012737  014536  007246          MOV     #PNCK,TEMP1
027040  012737  014550  007250  2$:     MOV     #PEC,TEMP2
027046  032737  000004  010500          BIT     #ECHOB,DEV4     ;SEE IF /ECHO OR /NOECHO
027054  001003                          BNE     3$              ;BR IF /ECHO
027056  012737  014546  007250          MOV     #PNEC,TEMP2
```

```
 6593  027064  012737  014557  007256  3$:     MOV     #PMS,TEMP5      ;ASSUME /MODEM ;REV B EC
 6594  027072  032737  000010  010500          BIT     #MOCHK,DEV4     ;MODEM CHECK ? ;REV B EC
 6595  027100  001003                          BNE     5$              ;YES,BRANCH ;REV B EC
 6596  027102  012737  014555  007256          MOV     #PNMS,TEMP5     ;"/NOMODEM" MESSAGE ;REV B EC
 6597
 6615
 6616  027110                          5$:     PRINTS  #SHF1,TEMP,TEMP1,TEMP2,TEMP5    ;,TEMP3,TEMP4 **;SEE NOTE ABOVE
   (11) 027110  013746  007256                                                                   MOV     TEMP5,-(SP)
   (10) 027114  013746  007250                                                                   MOV     TEMP2,-(SP)
   (9)  027120  013746  007246                                                                   MOV     TEMP1,-(SP)
   (8)  027124  013746  007244                                                                   MOV     TEMP,-(SP)
   (7)  027130  012746  015225                                                                   MOV     #SHF1,-(SP)
   (6)  027134  012746  000005                                                                   MOV     #5,-(SP)
   (3)  027140  010600                                                                           MOV     SP,R0
   (4)  027142  104416                                                                           TRAP    C$PNTS
   (4)  027144  062706  000014                                                                   ADD     #14,SP
 6617  027150  000207                          RTS     PC              ;RETURN
 6618
 6619
```

```
                                .SBTTL          TRAVERSE COMMAND LINE SUBROUTINES

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

027152                          P$TRV:
027152  013704  003362                  MOV     P$BUFA,R4
027156  013703  003364                  MOV     P$TREE,R3
027162  105714                 P$TR5:   TSTB    (R4)                   ;SEE IF ANY CHARS LEFT IN INPUT STRING
027164  001441                          BEQ     P$EXIT                 ;BR IF NO
027166  121327  000013                  CMPB    (R3),#11.              ;SEE IF SPECIAL CLI CHAR CODE OR ASCII
027172  003023                          BGT     20$                    ;BR IF REGULAR ASCII CHAR.
027174  111305                          MOVB    (R3),R5                ;GET SPECIAL CHAR CODE INTO R5
027176  006305                          ASL     R5
027200  016505  027214                  MOV     10$(R5),R5             ;BUILD TRAVERSE ROUTINE ADDRESS
027204  062705  027214                  ADD     #10$,R5
027210  004715                          JSR     PC,(R5)                ;JSR TO SPECIAL CLI TRAVERSE ROUTINE
027212  000763                          BR      P$TR5                  ;GO SEE IF MORE OF STRING LEFT

                                        ;TRAVERSE TABLE FOR 'CLI FUNTIONS'
027214  000114                 10$:     .WORD   TRVERR-10$             ;TAKE ERROR ACTION
027216  000134                          .WORD   TRVEXI-10$             ;TAKE EXIT ACTION
027220  000152                          .WORD   TRVBR-10$              ;TAKE BRANCH ACTION
027222  000162                          .WORD   TRVBIF-10$             ;TEST P$GDBD & TAKE BRANCH
027224  000204                          .WORD   TRVSPA-10$             ;SKIP SPACES OR TABS IN CMD LINE
027226  000270                          .WORD   TRVNUM-10$             ;TRAVERSE NUMERIC FIELD
027230  000604                          .WORD   TRVALP-10$             ;TRAVERSE ALPHABETICS
027232  000650                          .WORD   TRVALN-10$             ;TRAVERSE ALPHANUMERICS
027234  000270                          .WORD   TRVOCT-10$             ;SAME AS TRVNUM
027236  000256                          .WORD   TRVDEC-10$             ;SAME AS CLINUM BUT DECIMAL
027240  000736                          .WORD   TRVSTR-10$             ;FIND ASCIZ MATCH IN CMD LINE

                                ;NOT A SPECIAL CODE

027242  121314                 20$:     CMPB    (R3),(R4)              ;SEE IF FIRST CHAR OF STRING IS A MATCH
027244  001403                          BEQ     22$                    ;BR IF A MATCH
027246  004737  027312                  JSR     PC,TRVBRC              ;IF NOT A MATCH, GO TAKE MISS BRANCH
027252  000743                          BR      P$TR5                  ; THEN GO BACK PT'G TO MISS NODE
027254  004737  027272         22$:     JSR     PC,TRVACT              ;IF A MATCH, GO DO ACTION DEFINED BY
027260  062703  000004                  ADD     #4,R3                  ; ACTION CODE IN CLI NODE, THEN
                                                                       ; ADJUST PTR TO NEXT CLI NODE
027264  005204                          INC     R4                     ;ADJUST BUF PTR TO NEXT CHAR IF MATCH
027266  000735                          BR      P$TR5
```

```
6677
6678  027270  000207                    P$EXIT: RTS     PC                      ;RETURN FROM PARSER
6679
6680                                    ;-----------------------------------------------------------------------
6681
6682                                    ;GOTO USER ACTION ROUTINE
6683  027272  116302  000001            TRVACT: MOVB    1(R3),R2                ;GET ACTION CODE FROM CLI NODE
6684  027276  042702  177400                    BIC     #177400,R2              ;CLEAR ANY SIGN EXTENSION
6685  027302  013705  003366                    MOV     P$ACT,R5                ;GET ADDRESS OF CLI ACTION ROUTINE
6686  027306  004715                            JSR     PC,(R5)                 ;GO DO ACTION DEFINED BY CODE
6687  027310  000207                            RTS     PC                      ;RETURN TO CALLING CODE
6688
6689                                    ;TAKE BRANCH IN TREE
6690  027312  016305  000002            TRVBRC: MOV     2(R3),R5                ;GET BRANCH DISPLACEMENT FROM TREE
6691  027316  060503                            ADD     R5,R3                   ; AND POINT R3 TO THE "MISS" NODE
6692  027320  000207                            RTS     PC                      ; RETURN TO P$TRV
6693
6694                                    ;NO BRANCH TAKEN
6695  027322  062703  000004            TRVNOB: ADD     #4,R3                   ;THINGS OK, UPDATE R3 TO POINT TO NEXT
6696  027326  000207                            RTS     PC                      , NODE AND RETURN TO P$TRV
6697
6698                                    ;-----------------------------------------------------------------------
6699  027330  004737  027272            TRVERR: JSR     PC,TRVACT               ;TAKE ERROR ACTION
6700  027334  112737  177777  003377            MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
6701  027342  005726                            TST     (SP)+                   ;GET RID OF "JSR PUSH TO TRVERR"
6702  027344  000137  027270                    JMP     P$EXIT                  ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
6703
6704  027350  004737  027272            TRVEXI: JSR     PC,TRVACT               ;TAKE EXIT ACTION
6705  027354  105037  003377                    CLRB    P$GDBD                  ;SET GOOD/BAD FLAG TO "SUCCESS (0)"
6706  027360  005726                            TST     (SP)+                   ;GET RID OF "JSR PUSH TO TRVEXI"
6707  027362  000137  027270                    JMP     P$EXIT                  ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
6708
6709  027366  004737  027272            TRVBR:  JSR     PC,TRVACT               ;GO TAKE BRANCH ACTION
6710  027372  000137  027312                    JMP     TRVBRC
6711
6712  027376  004737  027272            TRVBIF: JSR     PC,TRVACT
6713  027402  105737  003377                    TSTB    P$GDBD                  ;SEE IF P$GDBD SET OR CLEARED BY ACTION
6714  027406  001402                            BEQ     1$                      ;IF CLEAR FALL THRU TO NEXT NODE
6715  027410  000137  027312                    JMP     TRVBRC                  ;ELSE TAKE THE "MISS" BRANCH
6716  027414  000137  027322            1$:     JMP     TRVNOB                  ;JUST UPDATE TO NEXT NODE IF THINGS OK
6717
6718  027420  005005                    TRVSPA: CLR     R5                      ;CLEAR "SPACE OR TAB FOUND" FLAG
6719  027422  121427  000011            1$:     CMPB    (R4),#11                ;SEE IF CHAR. IN CMD LINE= TAB
6720  027426  001003                            BNE     2$                      ;BR IF NO, NOT A TAB
6721  027430  005204                            INC     R4                      ;INC INPUT STRING POINTER
6722  027432  005205                            INC     R5                      ;INDICATE A TAB FOUND
6723  027434  000772                            BR      1$                      ;GO CHECK NEXT CHAR
6724
6725  027436  121427  000040            2$:     CMPB    (R4),#40                ;SEE IF CHAR. IN CMD LINE= SPACE
6726  027442  001003                            BNE     10$                     ;BR IF NO, NON-SPACE OR NON-TAB CHAR.
6727  027444  005204                            INC     R4                      ;INC INPUT STRING POINTER
6728  027446  005205                            INC     R5                      ;INDICATE A SPACE FOUND
6729  027450  000764                            BR      1$                      ;GO CHECK NEXT CHAR
6730  027452  005705                    10$:    TST     R5                      ;SEE IF ANY SPACES OR TABS FOUND
6731  027454  001404                            BEQ     15$                     ;BR IF NO, TAKE NO ACTION
6732  027456  004737  027272                    JSR     PC,TRVACT               ;GO TAKE ACTION IF ANY FOUND
```

```
                                        JMP     TRVNOB                  ;JUST GO UPDATE R3 TO NEXT NODE IF OK
                                15$:    JMP     TRVBRC                  ;TAKE BRANCH (MISS) IF NONE FOUND

                                TRVDEC: MOV     #10.,P$RADX             ;USE DECIMAL AS RADIX AND ASSUME +
                                        JMP     TRVNMA
                                TRVOCT: ;(SAME AS TRVNUM SINCE DEFAULT RADIX IS OCTAL)
                                TRVNUM: MOV     #8.,P$RADX              ;USE OCTAL AS RADIX AND ASSUME +
                                TRVNMA: CLR     R5                      ;CLEAR DIGIT COUNTER
                                        CMPB    (R4),#'+                ;SEE IF THERE'S A + SIGN THERE
                                        BNE     10$                     ; BR IF NO
                                        BR      11$                     ; ELSE P$RADX ALREADY SAYS +, JUST BR
                                10$:    CMPB    (R4),#'-                ;SEE IF THERE'S A - SIGN THERE
                                        BNE     1$                      ; BR IF NO
                                        MOVB    #-1,P$RADX+1            ;SET 'MINUS FLAG' (HI BYTE OF P$RADx)
                                11$:    INC     R4                      ;BUMP R4 TO POINT TO FIRST CHAR

                                1$:     CMPB    (R4),#60                ;SEE IF CHAR. LESS THAN A '0'
                                        BLT     2$                      ;BR IF YES (NOT NUMERIC)
                                        CMPB    (R4),#67                ;SEE IF CHAR. GREATER THAN A '7'
                                        BLE     13$                     ; BR IF YES
                                        CMPB    P$RADX,#10.             ;SEE IF IN DECIMAL MODE
                                        BEQ     12$                     ; BR IF YES (CAN USE HIGHER LIMIT)
                                        CMPB    (R4),#71                ;SEE IF DIGIT WAS A 8 OR 9
                                        BGT     2$                      ;BR IF NON-NUMERIC
                                        PRINTF  #CLIBRX                 ;ELSE WAS A 8 OR 9 WHEN IN OCTAL RADIX
                                                                                        MOV     #CLIBRX,-(SP)
                                                                                        MOV     #1,-(SP)
                                                                                        MOV     SP,R0
                                                                                        TRAP    C$PNTF
                                                                                        ADD     #4,SP
                                        MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
                                        BR      5$                      ; PRINT ERROR AND TAKE MISS

                                12$:    CMPB    (R4),#71                ;SEE IF CHAR. GREATER THAN A '9'
                                        BGT     2$                      ;BR IF YES (NOT NUMERIC)
                                13$:    INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR.
                                        INC     R5                      ;INDICATE A NUMERIC FOUND
                                        BR      1$                      ;GO LOOK AT NEXT CHAR.

                                2$:     TST     R5                      ;SEE IF FOUND ANY NUMERICS
                                        BEQ     5$                      ;BR IF NO, TAKE 'MISS' BRANCH
                                        MOV     R4,R1                   ;GET POINTER TO START OF NUMERIC STRING
                                        SUB     R5,R1
                                        CLR     P$NUM                   ;CLEAR LOC. WHERE VALUE WILL BE STORED
                                3$:     MOVB    (R1)+,R2                ;GET ASCII CHAR AND CONVERT IT TO A #
                                        SUB     #60,R2
                                        ASL     P$NUM                   ;SHIFT CURRENT VALUE TO MAKE ROOM
                                        BCS     7$                      ;ERROR IF NUMBER TOO BIG
                                        MOV     P$NUM,P$CNT             ;SAVE FOR LATER IN CASE DECIMAL RADIX
                                        ASL     P$NUM
                                        BCS     7$                      ;ERROR IF NUMBER TOO BIG
                                        ASL     P$NUM
                                        BCS     7$                      ;ERROR IF NUMBER TOO BIG
                                        CMPB    P$RADX,#10.             ;SEE IF DECIMAL RADIX
                                        BNE     4$                      ;BR IF NOT EQUAL
```

```
 6784  027722  063737  003370  003372          ADD     P$CNT,P$NUM
 6785  027730  103416                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
 6786  027732  060237  003372          4$:     ADD     R2,P$NUM
 6787  027736  103413                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
 6788  027740  005305                          DEC     R5
 6789  027742  001344                          BNE     3$
 6790  027744  105737  003375                  TSTB    P$RADX+1                ;SEE IF NUM WAS PRECEDED BY A - SIGN
 6791  027750  001402                          BEQ     15$                     ; BR IF NO
 6792  027752  005437  003372                  NEG     P$NUM                   ; ELSE NEGATE THE NUMBER BEFORE LEAVING
 6793  027756  004737  027272          15$:    JSR     PC,TRVACT               ;SINCE NUMERIC FOUND, GO TAKE ACTION
 6794  027762  000137  027322                  JMP     TRVNOB                  ;GO POINT R3 TO NEXT NODE
 6795
 6796  027766                          7$:     PRINTF  #CLINBG                 ;PRINT NUMBER TOO BIG ERROR
   (7) 027766  012746  012377                                                          MOV     #CLINBG,-(SP)
   (6) 027772  012746  000001                                                          MOV     #1,-(SP)
   (3) 027776  010600                                                                  MOV     SP,R0
   (4) 030000  104417                                                                  TRAP    C$PNTF
   (4) 030002  062706  000004                                                          ADD     #4,SP
 6797  030006  112737  177777  003377          MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
 6798  030014  000137  027312          5$:     JMP     TRVBRC                  ;TAKE "MISS" BRANCH
 6799
 6800
 6801  030020  005005                  TRVALP: CLR     R5                      ;CLEAR ALPHA FOUND FLAG
 6802  030022  121427  000101          1$:     CMPB    (R4),#101               ;SEE IF CHAR. LESS THAN A "A"
 6803  030026  002406                          BLT     2$                      ;BR IF YES (NOT ALPHA)
 6804  030030  121427  000132                  CMPB    (R4),#132               ;SEE IF CHAR. GREATER THAN A "Z"
 6805  030034  003003                          BGT     2$                      ;BR IF YES (NOT ALPHA)
 6806  030036  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR
 6807  030040  005205                          INC     R5                      ;INDICATE AN ALPHA WAS FOUND
 6808  030042  000767                          BR      1$                      ;GO LOOK AT NEXT CHAR.
 6809  030044  005705                  2$:     TST     R5                      ;SEE IF ANY ALPHA'S WERE FOUND
 6810  030046  001404                          BEQ     3$                      ;BR IF NO
 6811  030050  004737  027272                  JSR     PC,TRVACT               ;IF ANY FOUND TAKE ACTION
 6812  030054  000137  027322                  JMP     TRVNOB                  ;THEN UPDATE R3 TO NEXT NODE -NO BRANCH
 6813  030060  000137  027312          3$:     JMP     TRVBRC                  ;NONE FOUND, TAKE MISS BRANCH
 6814
 6815  030064  005005                  TRVALN: CLR     R5                      ;CLEAR ALPHANUM FOUND FLAG
 6816  030066  121427  000060          10$:    CMPB    (R4),#60                ;SEE IF CHAR. LESS THAN A "0"
 6817  030072  002417                          BLT     2$                      ;BR IF YES (NOT NUMERIC OR ALPHA)
 6818  030074  121427  000072                  CMPB    (R4),#72                ;SEE IF CHAR. GREATER THAN A "9"
 6819  030100  003003                          BGT     1$                      ;BR IF YES (NOT NUMERIC)
 6820  030102  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR.
 6821  030104  005205                          INC     R5                      ;INDICATE A NUMERIC FOUND
 6822  030106  000767                          BR      10$                     ;GO LOOK AT NEXT CHAR.
 6823  030110  121427  000101          1$:     CMPB    (R4),#101               ;SEE IF CHAR. LESS THAN A "A"
 6824  030114  002406                          BLT     2$                      ;BR IF YES (NOT ALPHA)
 6825  030116  121427  000132                  CMPB    (R4),#132               ;SEE IF CHAR. GREATER THAN A "Z"
 6826  030122  003003                          BGT     2$                      ;BR IF YES (NOT ALPHA)
 6827  030124  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR
 6828  030126  005205                          INC     R5                      ;INDICATE AN ALPHA FOUND
 6829  030130  000756                          BR      10$                     ;GO LOOK AT NEXT CHAR.
 6830  030132  005705                  2$:     TST     R5                      ;SEE IF ANY ALPHANUM'S WERE FOUND
 6831  030134  001404                          BEQ     3$                      ;BR IF NO
 6832  030136  004737  027272                  JSR     PC,TRVACT               ;IF ANY FOUND TAKE ACTION
 6833  030142  000137  027322                  JMP     TRVNOB                  ;THEN UPDATE R3 TO NEXT NODE -NO BRANCH
 6834  030146  000137  027312          3$:     JMP     TRVBRC                  ;NONE FOUND, TAKE MISS BRANCH
```

```
 6835
 6836
 6837
 6838  030152  010401                TRVSTR: MOV     R4,R1                   ;POINT R1 TO CMD STRING
 6839  030154  010305                        MOV     R3,R5
 6840  030156  062705  000006                ADD     #6,R5                   ;POUNT R5 TO MATCH STRING FROM CLI NODE
 6841  030162  005037  003370                CLR     P$CNT                   ;CLEAR CHAR MATCH COUNT
 6842  030166  105715                2$:     TSTB    (R5)                    ;SEE IF END OF MATCH STRING YET
 6843  030170  001411                        BEQ     10$                     ;BR IF YES
 6844  030172  105711                        TSTB    (R1)                    ;SEE IF END OF CMD LINE YET
 6845  030174  001407                        BEQ     10$                     ;BR IF YES
 6846  030176  121115                        CMPB    (R1),(R5)               ;SEE IF CHARACTERS MATCH
 6847  030200  001005                        BNE     10$                     ;BR IF NO
 6848  030202  005237  003370                INC     P$CNT                   ;MATCH -INCREMENT MATCH COUNT
 6849  030206  005201                        INC     R1                      ;UPDATE STRING POINTERS
 6850  030210  005205                        INC     R5
 6851  030212  000765                        BR      2$                      ;BR TO CONTINUE CHECKING CHARS.
 6852
 6853  030214  005737  003370        10$:    TST     P$CNT                   ;WHEN DONE SEE IF ANY MATCHES FOUND
 6854  030220  001406                        BEQ     15$                     ;BR IF NO, GO TAKE THE MISS BRANCH
 6855  030222  010104                        MOV     R1,R4                   ;POINT CMD POINTER TO END OF STRING &
 6856  030224  004737  027272                JSR     PC,TRVACT               ;IF A MATCH FOUND, GO DO MATCH ACTION
 6857  030230  066303  000004                ADD     4(R3),R3                ;UPDATE R3 TO NEXT NODE (NO BRANCH)
 6858  030234  000207                        RTS     PC                      ;  (NO RETURN THRU TRVNOB SINCE DIFFERNT
 6859                                                                        ;   DISPLACEMENT DUE TO MATCH STRING)
 6860  030236  000137  027312        15$:    JMP     TRVBRC                  ; GO TAKE BRANCH
 6861
 6862                                                                        ; (PARSED OK), -1 IF ILL CMD.....
 6863                                ;-----------------------------------------------------------------------
 6864
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 23-68
CZKMUA.P11    30-MAR-82 09:13          REPORT CODING SECTION

  6866                                .SBTTL  REPORT CODING SECTION
  6867
  6868
  6869                                ;++
  6870                                ; THE REPORT CODING SECTION CONTAINS THE
  6871                                ; 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
  6872                                ;--
  6873
  6874  030242                                BGNRPT
   (3)  030242                                                                  L$RPT::
  6875
  6887
  6888  030242  004737  023604                JSR     PC,REPORT             ;CALL SUBROUTINE TO DUMP EVENT LOG
  6889                                                                        ; AND BASE TABLE
  6890
  6897
  6898
  6905
  6906  030246                                ENDRPT
   (3)  030246                                                                  L10011:
   (3)  030246  104425                                                                  TRAP    C$RPT
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 23-69
CZKMUA.P11    30-MAR-82 09:13            PROTECTION TABLE

  6908                                    .SBTTL  PROTECTION TABLE
  6909
  6910                                    ;++
  6911                                    ; THIS TABLE IS USED BY THE RUNTIME SERVICES
  6912                                    ; TO PROTECT THE LOAD MEDIA.
  6913                                    ;--
  6914
  6915  030250                                    BGNPROT
    (3) 030250                                                                  L$PROT::
  6916
  6917  030250  177777                           -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
  6918  030252  177777                           -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
  6919  030254  177777                           -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER
  6920
  6921  030256                                    ENDPROT
  6922
```

```
                                .SBTTL  INITIALIZE SECTION

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

030256                                  BGNINIT
030256                                                                  L$INIT::

030256  005037  003202                  CLR     KEYWD1                  ;INIT USER COMMAND VARIABLE
030262  005737  007302                  TST     DCLFLG                  ;CLEANUP & EXIT ?
030266  001403                          BEQ     INIT1                   ;NO,BRANCH
030270  005037  007302                  CLR     DCLFLG                  ;CLEAR FLAG
030274                                  DOCLN                           ;GO CLEANUP
030274  104444                                                                  TRAP    C$DCLN

030276  012737  177777  007304  INIT1:  MOV     #-1,RESFLG              ;SET RESTART FLAG
030304                                  READEF  #EF.START               ;IF HERE CAUSE OF START,DO SOME INIT
030304  012700  000040                                                          MOV     #EF.START,R0
030310  104447                                                                  TRAP    C$REFG
030312                                  BCOMPLETE       START
030312  103417                                                                  BCS     START
030314                                  READEF  #EF.RESTART             ;IF HERE CAUSE OF RESTART, DO SOME INIT
030314  012700  000037                                                          MOV     #EF.RESTART,R0
030320  104447                                                                  TRAP    C$REFG
030322                                  BCOMPLETE       RESTRT
030322  103513                                                                  BCS     RESTRT
030324                                  READEF  #EF.CONTINUE            ;SEE IF WE'RE HERE CAUSE OF A CONTINUE
030324  012700  000036                                                          MOV     #EF.CONTINUE,R0
030330  104447                                                                  TRAP    C$REFG
030332                                  BNCOMPLETE      S1              ;BR IF NOT HERE CAUSE OF CONITNUE
030332  103002                                                                  BCC     S1
030334  000137  031314                  JMP     ENDIT                   ;JMP IF HERE CAUSE OF A CONTINUE
030340                          S1:     READEF  #EF.NEW                 ;SEE IF THIS IS A 'NEW PASS"
030340  012700  000035                                                          MOV     #EF.NEW,R0
030344  104447                                                                  TRAP    C$REFG
030346                                  BCOMPLETE       NEW             ;IF YES, BR AROUND LOGUNIT # SETUP
030346  103521                                                                  BCS     NEW
030350  000523                          BR      GETPRM

030352  005037  007304          START:  CLR     RESFLG                  ;CLEAR RESTART FLAG SINCE HERE ON START
030356  005037  007344                  CLR     CLKVEC                  ;CLEAR CLK VECTOR PTR. AS A FLAG IN
                                                                        ; NO CLOCK IS FOUND.
030362  012702  007340                  MOV     #CLKCSR,R2              ;SETUP R2 AS A PTR. TO CLOCK INFO BLOCK
030366                                  CLOCK   L,R1                    ;LOOK FOR A LINE CLOCK
030366  012700  000114                                                          MOV     #'L,R0
030372  104462                                                                  TRAP    C$CLCK
030374  010001                                                                  MOV     R0,R1
030376                                  BNCOMPLETE      S2              ; IF NONE THERE GO LOOK FOR A P-CLOCK
030376  103006                                                                  BCC     S2
030400  004737  022742                  JSR     PC,CLKSET               ; GO SET UP CLOCK INFO TABLE & CLK VEC.
030404  012737  000100  007350          MOV     #LCLKEN,CLKEN           ;SETUP THE ENABLE LINE CLOCK DATA
030412  000457                          BR      RESTRT

030414                          S2:     CLOCK   P,R1                    ;LOOK FOR A P-CLOCK SINCE NO LINE CLOCK
```

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

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

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

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

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

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

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

```
  (3)  030642  010001                                                                        MOV     R0,R1
 7032  030644                                   BNCOMPLETE      GETPRM          ;IF NO P-TABLE AVAIL., GO GET NEXT ONE
  (2)  030644  103365                                                                        BCC     GETPRM
 7033
 7034  030646  011137  007312                   MOV     (R1),FHDPLX             ;PUT FULL OR HALF DUPLEX ANSWER IN LOC.
 7035
 7046
 7047                                           ;DEVICE DEPENDENT PART OF GETTING INFO FROM P-TABLE
 7048
 7049  030652  016137  000002  012202           MOV     2(R1),SEL0              ;STORE AWAY CSR ADDRESSES
 7050  030660  016137  000002  012204           MOV     2(R1),BSEL1
 7051  030666  005237  012204                   INC     BSEL1
 7052  030672  016137  000002  012206           MOV     2(R1),SEL2
 7053  030700  062737  000002  012206           ADD     #2,SEL2
 7054  030706  016137  000002  012210           MOV     2(R1),BSEL3
 7055  030714  062737  000003  012210           ADD     #3,BSEL3
 7056  030722  016137  000002  012212           MOV     2(R1),SEL4
 7057  030730  062737  000004  012212           ADD     #4,SEL4
 7058  030736  016137  000002  012214           MOV     2(R1),BSEL5
 7059  030744  062737  000005  012214           ADD     #5,BSEL5
 7060  030752  016137  000002  012216           MOV     2(R1),SEL6
 7061  030760  062737  000006  012216           ADD     #6,SEL6
 7062  030766  016137  000002  012220           MOV     2(R1),BSEL7
 7063  030774  062737  000007  012220           ADD     #7,BSEL7
 7064
 7065  031002  016137  000004  012222           MOV     4(R1),INVEC             ;STORE AWAY INPUT INTERRUPT VECTOR
 7066  031010  016137  000004  012224           MOV     4(R1),OUTVEC
 7067  031016  062737  000004  012224           ADD     #4,OUTVEC               ;BUILD OUTPUT INTERRUPT VECTOR
 7068  031024  016137  000006  012226           MOV     6(R1),INTPRI            ;STORE AWAY INTERRUPT PRIORITY
 7069
 7070                                   ;INITIALIZATION CODE - LOAD FIRMWARE INTO KMC-11B
 7071
 7072  031032  005005                   LDFIRM: CLR     R5                      ;SPECIFY INITIAL CRAM ADDRESS
 7073  031034  012704  031326                   MOV     #MCBEGL,R4              ;POINT TO KMS11 FIRMWARE IMAGE
 7074  031040  012777  002000  161134           MOV     #RAM0,@SEL0             ;SPECIFY LOAD CRAM
 7075  031046  010577  161140           2$:     MOV     R5,@SEL4                ;WRITE CRAM ADDRESS
 7076  031052  011477  161140                   MOV     (R4),@SEL6              ;WRITE IMAGE WORD TO CRAM
 7077  031056  017702  161134                   MOV     @SEL6,R2                ;READBACK WORD JUST WRITTEN
 7078  031062  022402                           CMP     (R4)+,R2                ;IF WORD READ BACK IS WORD WRITTEN
 7079  031064  001446                           BEQ     10$                     ; THEN CONTINUE LOADING
 7080                                                                   ;FAILURE LOADING FIRMWARE
 7081  031066  010537  007250                   MOV     R5,TEMP2                ;SAVE CRAM ADDRESS
 7082  031072  016437  177776  007252           MOV     -2(R4),TEMP3            ;SAVE WORD LOADED
 7083  031100  010237  007254                   MOV     R2,TEMP4                ;SAVE WORD READ BACK
 7084  031104                                   PRINTF  #FIRMLD                 ;PRINT FAILURE MESSAGE
  (7)  031104  012746  035326                                                                MOV     #FIRMLD,-(SP)
  (6)  031110  012746  000001                                                                MOV     #1,-(SP)
  (3)  031114  010600                                                                        MOV     SP,R0
  (4)  031116  104417                                                                        TRAP    C$PNTF
  (4)  031120  062706  000004                                                                ADD     #4,SP
 7085  031124                                   PRINTF  #DATAHD                 ;PRINT HEADER
  (7)  031124  012746  035366                                                                MOV     #DATAHD,-(SP)
  (6)  031130  012746  000001                                                                MOV     #1,-(SP)
  (3)  031134  010600                                                                        MOV     SP,R0
  (4)  031136  104417                                                                        TRAP    C$PNTF
  (4)  031140  062706  000004                                                                ADD     #4,SP
```

```
7086  031144                                  PRINTF  #DATALD,TEMP2,TEMP3,TEMP4       ;PRINT DATA
 (10) 031144  013746  007254                                                          MOV     TEMP4,-(SP)
 (9)  031150  013746  007252                                                          MOV     TEMP3,-(SP)
 (8)  031154  013746  007250                                                          MOV     TEMP2,-(SP)
 (7)  031160  012746  035437                                                          MOV     #DATALD,-(SP)
 (6)  031164  012746  000004                                                          MOV     #4,-(SP)
 (3)  031170  010600                                                                  MOV     SP,R0
 (4)  031172  104417                                                                  TRAP    C$PNTF
 (4)  031174  062706  000012                                                          ADD     #12,SP
7087  031200                                  DOCLN                                ;CLEAN UP AND EXIT
 (3)  031200  104444                                                                  TRAP    C$DCLN
7088
7089                                                    ;CONTINUE LOADING FIRMWARE
7090  031202  005205                    10$:  INC     R5                           ;UPDATE THE CRAM ADDRESS
7091  031204  022704  035326                CMP     #MCENDL,R4                   ;TEST FOR DONE LOADING KMS11 FIRMWARE
7092  031210  001316                        BNE     2$                           ;LOOP TIL FIRMWARE LOADED
7093
7094                                                    ;FIRMWARE LOADED
7095
7096  031212                                  SETVEC  CLKVEC,#CLKINT,#340          ;SETUP CLOCK VECTOR
 (7)  031212  012746  000340                                                          MOV     #340,-(SP)
 (6)  031216  012746  022766                                                          MOV     #CLKINT,-(SP)
 (5)  031222  013746  007344                                                          MOV     CLKVEC,-(SP)
 (4)  031226  012746  000003                                                          MOV     #3,-(SP)
 (3)  031232  104437                                                                  TRAP    C$SVEC
 (2)  031234  062706  000010                                                          ADD     #10,SP
7097
7098                                                    ;DEVICE DEPENDENT VECTOR SETUP
7099
7108  031240                                  SETVEC  INVEC,#DVINS,INTPRI          ;SETUP INPUT INTERRUPT VECTOR
 (7)  031240  013746  012226                                                          MOV     INTPRI,-(SP)
 (6)  031244  012746  045310                                                          MOV     #DVINS,-(SP)
 (5)  031250  013746  012222                                                          MOV     INVEC,-(SP)
 (4)  031254  012746  000003                                                          MOV     #3,-(SP)
 (3)  031260  104437                                                                  TRAP    C$SVEC
 (2)  031262  062706  000010                                                          ADD     #10,SP
7109  031266                                  SETVEC  OUTVEC,#DVOUTS,INTPRI        ;SETUP OUTPUT INTERRUPT VECTOR
 (7)  031266  013746  012226                                                          MOV     INTPRI,-(SP)
 (6)  031272  012746  045320                                                          MOV     #DVOUTS,-(SP)
 (5)  031276  013746  012224                                                          MOV     OUTVEC,-(SP)
 (4)  031302  012746  000003                                                          MOV     #3,-(SP)
 (3)  031306  104437                                                                  TRAP    C$SVEC
 (2)  031310  062706  000010                                                          ADD     #10,SP
7110
7111  031314                    ENDIT:
7112  031314                                  SETPRI  #PRI00                       ;SET THE 'RUN' PRIORITY TO 0
 (3)  031314  012700  000000                                                          MOV     #PRI00,R0
 (3)  031320  104441                                                                  TRAP    C$SPRI
7113
7114  031322                    EXITIT: EXIT    INIT
 (3)  031322  104432                                                                  TRAP    C$EXIT
 (3)  031324  004140                                                                  .WORD   L10013-.
7115
7116
7117  031326                    MCBEGL:                       ;START OF KMS11 FIRMWARE IMAGE
7118  031326  063220  063223  063237          .WORD   ^0063220,^0063223,^0063237,^0063232,^0061200,^0061202,^0003370,^0063130
```

```
                031334  063232  061200  061202
                031342  003370  063130
7119            031346  076423  063060  101414          .WORD   ^0076423,^0063060,^0101414,^0100407,^0003401,^0063231,^0010162,^0000626
                031354  100407  003401  063231
                031362  010162  000626
7120            031366  062234  016403  016402          .WORD   ^0062234,^0016403,^0016402,^0016406,^0016407,^0016401,^0010210,^0016455
                031374  016406  016407  016401
                031402  010210  016455
7121            031406  016461  010067  016471          .WORD   ^0016461,^0010067,^0016471,^0002471,^0043236,^0010022,^0016424,^0002424
                031414  002471  043236  010022
                031422  016424  002424
7122            031426  043235  010240  016642          .WORD   ^0043235,^0010240,^0016642,^0002642,^0000600,^0061221,^0110642,^0020620
                031434  002642  000600  061221
                031442  110642  020620
7123            031446  173202  020640  167203          .WORD   ^0173202,^C020640,^0167203,^0010210,^0140620,^0123620,^0113246,^0010211
                031454  010210  140620  123620
                031462  113246  010211
7124            031466  140620  060601  103214          .WORD   ^0140620,^0060601,^0103214,^0123400,^0001620,^0117044,^0023660,^0060520
                031474  123400  001620  117044
                031502  023660  060520
7125            031506  103105  060610  001620          .WORD   ^0103105,^0060610,^0001620,^0103047,^0060521,^0102447,^0103447,^0123420
                031514  103047  060521  102447
                031522  103447  123420
7126            031526  060400  103047  000500          .WORD   ^0060400,^0103047,^0000500,^0063301,^0114657,^0000677,^0100646,^0123400
                031534  063301  114657  000677
                031542  100646  123400
7127            031546  103512  114444  060520          .WORD   ^0103512,^0114444,^0060520,^0103606,^0120400,^0001620,^0103066,^0123400
                031554  103606  120400  001620
                031562  103066  123400
7128            031566  102532  002655  102127          .WORD   ^0102532,^0002655,^0102127,^0001620,^0002723,^0102527,^0002702,^0060601
                031574  001620  002723  102527
                031602  002702  060601
7129            031606  102133  100447  102143          .WORD   ^0102133,^0100447,^0102143,^0002461,^0000500,^0061260,^0010177,^0016402
                031614  002461  000500  061260
                031622  010177  016402
7130            031626  002400  042233  114511          .WORD   ^0002400,^0042233,^0114511,^0060721,^0102133,^0002461,^0010017,^0136500
                031634  060721  102133  002461
                031642  010017  136500
7131            031646  136520  122560  123000          .WORD   ^0136520,^0122560,^0123000,^0000500,^0061260,^0002133,^0040620,^0103167
                031654  000500  061260  002133
                031662  040620  103167
7132            031666  010151  016406  002700          .WORD   ^0010151,^0016406,^0002700,^0063161,^0000641,^0003374,^0110743,^0003004
                031674  063161  000641  003374
                031702  110743  003004
7133            031706  063070  010017  000745          .WORD   ^0063070,^0010017,^0000745,^0110463,^0010154,^0057310,^0057231,^0057235
                031714  110463  010154  057310
                031722  057231  057235
7134            031726  043237  043232  063170          .WORD   ^0043237,^0043232,^0063170,^0063161,^0000606,^0114434,^0002514,^0000415
                031734  063161  000606  114434
                031742  002514  000415
7135            031746  123220  063260  000600          .WORD   ^0123220,^0063260,^0000600,^0104457,^0002650,^0010240,^0050220,^0123040
                031754  104457  002650  010240
                031762  050220  123040
7136            031766  055302  050220  074520          .WORD   ^0055302,^0050220,^0074520,^0055224,^0055225,^0055227,^0055226,^0103752
                031774  055224  055225  055227
                032002  055226  103752
```

```
 7137  032006  010240  043220  002642          .WORD   ^0010240,^0043220,^0002642,^0000776,^0060360,^0101640,^0000402,^0062400
       032014  000776  060360  101640
       032022  000402  062400
 7138  032026  043220  010241  040360          .WORD   ^0043220,^0010241,^0040360,^0101645,^0100447,^0000757,^0063261,^0100447
       032034  101645  100447  000757
       032042  063261  100447
 7139  032046  123440  103466  000500          .WORD   ^0123440,^0103466,^0000500,^0061262,^0100667,^0123560,^0001620,^0102677
       032054  061262  100667  123560
       032062  001620  102677
 7140  032066  002113  060600  116351          .WORD   ^0002113,^0060600,^0116351,^0123000,^0000500,^0061260,^0010211,^0002461
       032074  123000  000500  061260
       032102  010211  002461
 7141  032106  100447  123220  000515          .WORD   ^0100447,^0123220,^0000515,^0063260,^0000404,^0061011,^0114734,^0000500
       032114  063260  000404  061011
       032122  114734  000500
 7142  032126  063310  100661  010070          .WORD   ^0063310,^0100661,^0010070,^0053220,^0016401,^0014543,^0136500,^0136520
       032134  053220  016401  014543
       032142  136500  136520
 7143  032146  136560  136540  010070          .WORD   ^0136560,^0136540,^0010070,^0002471,^0060360,^0101720,^0000406,^0062400
       032154  002471  060360  101720
       032162  000406  062400
 7144  032166  000402  063310  100663          .WORD   ^0000402,^0063310,^0100663,^0010023,^0053220,^0016401,^0136500,^0136520
       032174  010023  053220  016401
       032202  136500  136520
 7145  032206  136560  136540  010023          .WORD   ^0136560,^0136540,^0010023,^0002424,^0000462,^0060360,^0101663,^0000405
       032214  002424  000462  060360
       032222  101663  000405
 7146  032226  062400  100663  000757          .WORD   ^0062400,^0100663,^0000757,^0063670,^0001620,^0107015,^0000400,^0063233
       032234  063670  001620  107015
       032242  000400  063233
 7147  032246  000427  104425  000700          .WORD   ^0000427,^0104425,^0000700,^0123220,^0061311,^0100630,^0120600,^0102047
       032254  123220  061311  100630
       032262  120600  102047
 7148  032266  114725  000404  114657          .WORD   ^0114725,^0000404,^0114657,^0000720,^0110554,^0000727,^0063270,^0104504
       032274  000720  110554  000727
       032302  063270  104504
 7149  032306  000000  000000  000000          .WORD   ^0000000,^0000000,^0000000,^0000000,^0000000,^0000000,^0000000,^0000000
       032314  000000  000000  000000
       032322  000000  000000
 7150  032326  023200  060601  106012          .WORD   ^0023200,^0060601,^0106012,^0107412,^0000601,^0060360,^0101742,^0000405
       032334  107412  000601  060360
       032342  101742  000405
 7151  032346  060360  105424  000620          .WORD   ^0060360,^0105424,^0000620,^0060360,^0115760,^0002212,^0000400,^0063223
       032354  060360  115760  002212
       032362  000400  063223
 7152  032366  000757  063270  060070          .WORD   ^0000757,^0063270,^0060070,^0110665,^0000703,^0063223,^0100447,^0023204
       032374  110665  000703  063223
       032402  100447  023204
 7153  032406  070215  054620  106042          .WORD   ^0070215,^0054620,^0106042,^0060601,^0107440,^0000710,^0010012,^0104560
       032414  060601  107440  000710
       032422  010012  104560
 7154  032426  000402  063301  000462          .WORD   ^0000402,^0063301,^0000462,^0063223,^0056226,^0056227,^0123220,^0000501
       032434  063223  056226  056227
       032442  123220  000501
 7155  032446  063260  003305  040665          .WORD   ^0063260,^0003305,^0040665,^0001620,^0001620,^0001620,^0001620,^0061311
       032454  001620  001620  001620
```

```
                032462  001620  061311
7156            032466  100447  100453  104756          .WORD   ^0100447,^0100453,^0104756,^0010167,^0076604,^0076605,^0000470,^0104425
                032474  010167  076604  076605
                032502  000470  104425
7157            032506  000522  063223  023600          .WORD   ^0000522,^0063223,^0023600,^0060757,^0060641,^0107514,^0060610,^0001620
                032514  060757  060641  107514
                03252.  060610  001620
7158            03252?  103047  106110  010153          .WORD   ^0103047,^0106110,^0010153,^0062600,^0010003,^0002401,^0003374,^0100447
                032534  062600  010003  002401
                032542  003374  100447
7159            032546  060600  060772  103447          .WORD   ^0060600,^0060772,^0103447,^0104502,^0060600,^0060372,^0105506,^0060530
                032554  104502  060600  060372
                032562  105506  060530
7160            032566  113705  100447  060601          .WORD   ^0113705,^0100447,^0060601,^0107716,^0020600,^0060371,^0105531,^0063173
                032574  107716  020600  060371
                032602  105531  063173
7161            032606  104532  063071  000534          .WORD   ^0104532,^0063071,^0000534,^0104425,^0063164,^0105137,^0063165,^0000542
                032614  104425  063164  105137
                032622  063165  000542
7162            032626  020200  104425  000544          .WORD   ^0020200,^0104425,^0000544,^0104540,^0023200,^0020640,^0116157,^0060601
                032634  104540  023200  020640
                032642  116157  060601
7163            032646  107746  060610  001620          .WORD   ^0107746,^0060610,^0001620,^0117303,^0010153,^0062617,^0000701,^0010013
                032654  117303  010153  062617
                032662  000701  010013
7164            032666  043220  062460  010151          .WORD   ^0043220,^0062460,^0010151,^0016402,^0062620,^0010001,^0040620,^0061620
                032674  016402  062620  010001
                032702  040620  061620
7165            032706  062620  116252  114700          .WORD   ^0062620,^0116252,^0114700,^0060573,^0115456,^0000400,^0063223,^0060610
                032714  060573  115456  000400
                032722  063223  060610
7166            032726  117024  001620  107210          .WORD   ^0117024,^0001620,^0107210,^0010162,^0054373,^0115402,^0054373,^0115426
                032734  010162  054373  115402
                032742  054373  115426
7167            032746  010164  054373  115516          .WORD   ^0010164,^0054373,^0115516,^0054373,^0115411,^0054373,^0105773,^0100447
                032754  054373  115411  054373
                032762  105773  100447
7168            032766  023640  060400  103451          .WORD   ^0023640,^0060400,^0103451,^0110402,^0123600,^0102047,^0022203,^0000672
                032774  110402  123600  102047
                033002  022203  000672
7169            033006  063223  000621  104641          .WORD   ^0063223,^0000621,^0104641,^0123600,^0102047,^0000646,^0063223,^0022203
                033014  123600  102047  000646
                033022  063223  022203
7170            033026  000421  061310  100447          .WORD   ^0000421,^0061310,^0100447,^0123600,^0000720,^0104630,^0120600,^0102047
                033034  123600  000720  104630
                033042  120600  102047
7171            033046  023140  062066  063164          .WORD   ^0023140,^0062066,^0063164,^0105256,^0063165,^0105724,^0022202,^0023140
                033054  105256  063165  105724
                033062  022202  023140
7172            033066  062066  023160  062107          .WORD   ^0062066,^0023160,^0062107,^0115016,^0063164,^0105270,^0063165,^0105643
                033074  115016  063164  105270
                033102  063165  105643
7173            033106  000633  104425  123200          .WORD   ^0000633,^0104425,^0123200,^0000577,^0061270,^0104652,^0023200,^0062202
                033114  000577  061270  104652
                033122  023200  062202
7174            033126  060601  117571  104664          .WORD   ^0060601,^0117571,^0104664,^0023213,^0000706,^0104425,^0104756,^0000711
```

```
                033134  023213  000706  104425
                033142  104756  000711
7175            033146  104425  000403  060353          .WORD   ^0104425,^0000403,^0060353,^0000716,^0105140,^0104471,^0000534,^0104540
                033154  000716  105140  104471
                033162  000534  104540
7176            033166  123600  102047  000576          .WORD   ^0123600,^0102047,^0000576,^0061270,^0000726,^0104540,^0120620,^0116051
                033174  061270  000726  104540
                033202  120620  116051
7177            033206  023200  020640  116206          .WORD   ^0023200,^0020640,^0116206,^0060601,^0107746,^0000702,^0010014,^0104560
                033214  060601  107746  000702
                033222  010014  104560
7178            033226  000410  010177  114507          .WORD   ^0000410,^0010177,^0114507,^0020600,^0060373,^0105754,^0060521,^0107351
                033234  020600  060373  105754
                033242  060521  107351
7179            033246  104415  000565  063261          .WORD   ^0104415,^0000565,^0063261,^0104415,^0063164,^0100447,^0023605,^0107763
                033254  104415  063164  100447
                033262  023605  107763
7180            033266  060525  107766  164463          .WORD   ^0060525,^0107766,^0164463,^0000600,^0063310,^0104760,^0000420,^0063310
                033274  000600  063310  104760
                033302  000420  063310
7181            033306  164463  000402  114657          .WORD   ^0164463,^0000402,^0114657,^0060530,^0103365,^0107422,^0100447,^0000000
                033314  060530  103365  107422
                033322  100447  000000
7182            033326  060530  107620  060610          .WORD   ^0060530,^0107620,^0060610,^0113405,^0100451,^0112032,^0112410,^0100451
                033334  113405  100451  112032
                033342  112410  100451
7183            033346  001620  103051  112432          .WORD   ^0001620,^0103051,^0112432,^0000450,^0063222,^0060521,^0113021,^0000620
                033354  000450  063222  060521
                033362  113021  000620
7184            033366  111024  060610  112026          .WORD   ^0111024,^0060610,^0112026,^0000601,^0062230,^0100451,^0000601,^0063222
                033374  000601  062230  100451
                033402  000601  063222
7185            033406  000405  110424  020660          .WORD   ^0000405,^0110424,^0020660,^0001620,^0103051,^0000773,^0063270,^0000443
                033414  001620  103051  000773
                033422  063270  000443
7186            033426  063222  000410  063226          .WORD   ^0063222,^0000410,^0063226,^0063166,^0111413,^0002011,^0000626,^0110424
                033434  063166  111413  002011
                033442  000626  110424
7187            033446  063472  062220  070216          .WORD   ^0063472,^0062220,^0070216,^0016403,^0076612,^0000543,^0060376,^0111504
                033454  016403  076612  000543
                033462  060376  111504
7188            033466  000406  063016  000506          .WORD   ^0000406,^0063016,^0000506,^0063222,^0056224,^0056225,^0043227,^0123200
                033474  063222  056224  056225
                033502  043227  123200
7189            033506  000620  063260  003306          .WORD   ^0000620,^0063260,^0003306,^0054666,^0042230,^0001620,^0001620,^0001620
                033514  054666  042230  001620
                033522  001620  001620
7190            033526  001620  061310  043626          .WORD   ^0001620,^0061310,^0043626,^0100447,^0000726,^0110461,^0000477,^0063667
                033534  100447  000726  110461
                033542  000477  063667
7191            033546  062230  000513  110554          .WORD   ^0062230,^0000513,^0110554,^0000522,^0063166,^0111117,^0063167,^0010171
                033554  000522  063166  111117
                033562  063167  010171
7192            033566  042230  110554  123600          .WORD   ^0042230,^0110554,^0123600,^0102051,^0022010,^0000527,^0110574,^0000533
                033574  102051  022010  000527
                033602  110574  000533
```

```
7193  033606  063222  000401  110424          .WORD   ^0063222,^0000401,^0110424,^0000402,^0062231,^0062230,^0020500,^0112153
      033614  000402  062231  062230
      033622  020500  112153
7194  033626  000542  110554  123600          .WORD   ^0000542,^0110554,^0123600,^0102051,^0022010,^0023100,^0062064,^0063166
      033634  102051  022010  023100
      033642  062064  063166
7195  033646  111153  063167  101763          .WORD   ^0111153,^0063167,^0101763,^0000556,^0063222,^0100451,^0123600,^0102051
      033654  000556  063222  100451
      033662  123600  102051
7196  033666  022030  023100  062064          .WORD   ^0022030,^0023100,^0062064,^0023120,^0062105,^0111376,^0063166,^0111172
      033674  023120  062105  111376
      033702  063166  111172
7197  033706  063167  101763  123600          .WORD   ^0063167,^0101763,^0123600,^0000542,^0063222,^0000556,^0063260,^0061070
      033714  000542  063222  000556
      033722  063260  061070
7198  033726  104506  010151  043226          .WORD   ^0104506,^0010151,^0043226,^0000605,^0110520,^0010152,^0000610,^0110520
      033734  000605  110520  010152
      033742  000610  110520
7199  033746  000403  060346  000616          .WORD   ^0000403,^0060346,^0000616,^0063222,^0111223,^0110517,^0000627,^0063222
      033754  063222  111223  110517
      033762  000627  063222
7200  033766  000403  060366  111625          .WORD   ^0000403,^0060366,^0111625,^0000400,^0110424,^0060612,^0110424,^0000631
      033774  000400  110424  060612
      034002  110424  000631
7201  034006  110530  000402  062231          .WORD   ^0110530,^0000402,^0062231,^0062230,^0000404,^0063710,^0060530,^0113644
      034014  062230  000404  063710
      034022  060530  113644
7202  034026  000776  063270  000400          .WORD   ^0000776,^0063270,^0000400,^0110554,^0000576,^0110641,^0000577,^0061271
      034034  110554  000576  110641
      034042  000577  061271
7203  034046  010236  043220  063460          .WORD   ^0010236,^0043220,^0063460,^0076620,^0111256,^0100457,^0043220,^0063460
      034054  076620  111256  100457
      034062  043220  063460
7204  034066  062620  111263  100457          .WORD   ^0062620,^0111263,^0100457,^0002650,^0063074,^0060601,^0102047,^0103447
      034074  002650  063074  060601
      034102  102047  103447
7205  034106  060610  061620  103164          .WORD   ^0060610,^0061620,^0103164,^0111277,^0116306,^0117706,^0110737,^0000605
      034114  111277  116306  117706
      034122  110737  000605
7206  034126  063310  003374  060612          .WORD   ^0063310,^0003374,^0060612,^0060377,^0111737,^0010015,^0043220,^0062460
      034134  060377  111737  010015
      034142  043220  062460
7207  034146  010003  040620  061620          .WORD   ^0010003,^0040620,^0061620,^0062620,^0106371,^0010151,^0016403,^0110741
      034154  062620  106371  010151
      034162  016403  110741
7208  034166  120620  116051  000402          .WORD   ^0120620,^0116051,^0000402,^0062231,^0062230,^0060601,^0113764,^0070216
      034174  062231  062230  060601
      034202  113764  070216
7209  034206  040620  112242  000775          .WORD   ^0040620,^0112242,^0000775,^0063670,^0112242,^0000400,^0063222,^0010151
      034214  063670  112242  000400
      034222  063222  010151
7210  034226  016401  000405  016700          .WORD   ^0016401,^0000405,^0016700,^0063310,^0100447,^0120600,^0102047,^0070604
      034234  063310  100447  120600
      034242  102047  070604
7211  034246  103574  036400  036420          .WORD   ^0103574,^0036400,^0036420,^0000402,^0063004,^0023100,^0062004,^0023120
      034254  000402  063004  023100
```

```
                034262  062004  023120
7212            034266  062105  123200  115166          .WORD   ^0062105,^0123200,^0115166,^0110575,^0000774,^0063270,^0063233,^0010067
                034274  110575  000774  063270
                034302  063233  010067
7213            034306  043220  000404  063310          .WORD   ^0043220,^0000404,^0063310,^0000400,^0063222,^0114531,^0123200,^0000404
                034314  000400  063222  114531
                034322  123200  000404
7214            034326  061010  110566  010016          .WORD   ^0061010,^0110566,^0010016,^0043220,^0062460,^0010151,^0016402,^0016703
                034334  043220  062460  010151
                034342  016402  016703
7215            034346  114700  117013  100447          .WORD   ^0114700,^0117013,^0100447,^0000727,^0063270,^0110737,^0123220,^0000515
                034354  000727  063270  110737
                034362  123220  000515
7216            034366  063260  000404  061011          .WORD   ^0063260,^0000404,^0061011,^0104664,^0002212,^0104601,^0010011,^0043220
                034374  104664  002212  104601
                034402  010011  043220
7217            034406  042460  060617  063232          .WORD   ^0042460,^0060617,^0063232,^0000406,^0010067,^0053236,^0063310,^0003374
                034414  000406  010067  053236
                034422  063310  003374
7218            034426  113727  040620  102051          .WORD   ^0113727,^0040620,^0102051,^0110732,^0000600,^0123000,^0061300,^0002507
                034434  110732  000600  123000
                034442  061300  002507
7219            034446  100447  010177  016401          .WORD   ^0100447,^0010177,^0016401,^0002400,^0043230,^0114511,^0070075,^0060601
                034454  002400  043230  114511
                034462  070075  060601
7220            034466  116472  014477  063265          .WORD   ^0116472,^0014477,^0063265,^0077220,^0054660,^0060365,^0115476,^0115100
                034474  077220  054660  060365
                034502  115476  115100
7221
7222            034506  020540  116074  000676          .WORD   ^0020540,^0116074,^0000676,^0104425,^0000624,^0104425,^0040364,^0114467
                034514  104425  000624  104425
                034522  040364  114467
7223            034526  010151  016402  002711          .WORD   ^0010151,^0016402,^0002711,^0010175,^0036540,^0036560,^0000420,^0016400
                034534  010175  036540  036560
                034542  000420  016400
7224            034546  062620  002212  000404          .WORD   ^0062620,^0002212,^0000404,^0063222,^0003001,^0114662,^0060610,^0001620
                034554  063222  003001  114662
                034562  060610  001620
7225            034566  117123  000600  104741          .WORD   ^0117123,^0000600,^0104741,^0010151,^0016407,^0000411,^0110742,^0040757
                034574  010151  016407  000411
                034602  110742  040757
7226            034606  107573  070200  002400          .WORD   ^0107573,^0070200,^0002400,^0010067,^0002471,^0000543,^0060360,^0115542
                034614  010067  002471  000543
                034622  060360  115542
7227            034626  000406  062400  010241          .WORD   ^0000406,^0062400,^0010241,^0053223,^0016600,^0062460,^0010241,^0002642
                034634  053223  016600  062460
                034642  010241  002642
7228            034646  000776  060363  115555          .WORD   ^0000776,^0060363,^0115555,^0000402,^0062403,^0000420,^0063301,^0010153
                034654  000402  062403  000420
                034662  063301  010153
7229            034666  043237  010067  053620          .WORD   ^0043237,^0010067,^0053620,^0054620,^0116527,^0104573,^0000404,^0063000
                034674  054620  116527  104573
                034702  000404  063000
7230            034706  110575  023333  115576          .WORD   ^0110575,^0023333,^0115576,^0000406,^0060360,^0115603,^0060601,^0106746
                034714  000406  060360  115603
                034722  060601  106746
```

```
7231  034726 060521 107340 104664          .WORD   ^0060521,^0107340,^0104664,^0063164,^0000743,^0104425,^0010001,^0002401
      034734 063164 000743 104425
      034742 010001 002401
7232  034746 010167 057224 057225          .WORD   ^0010167,^0057224,^0057225,^0062571,^0010241,^0053223,^0016604,^0072615
      034754 062571 010241 053223
      034762 016604 072615
7233  034766 016400 017300 014402          .WORD   ^0016400,^0017300,^0014402,^0042660,^0056705,^0076604,^0010241,^0062403
      034774 042660 056705 076604
      035002 010241 062403
7234  035006 000776 060363 115646          .WORD   ^0000776,^0060363,^0115646,^0000462,^0060375,^0115650,^0000405,^0063015
      035014 000462 060375 115650
      035022 000405 063015
7235  035026 000420 063301 060610          .WORD   ^0000420,^0063301,^0060610,^0107015,^0000400,^0104425,^0002642,^0114633
      035034 107015 000400 104425
      035042 002642 114633
7236  035046 000742 114637 010152          .WORD   ^0000742,^0114637,^0010152,^0043620,^0060400,^0103361,^0000401,^0010177
      035054 043620 060400 103361
      035062 000401 010177
7237  035066 016400 062620 010241          .WORD   ^0016400,^0062620,^0010241,^0053220,^0016601,^0002574,^0010241,^0002642
      035074 053220 016601 002574
      035102 010241 002642
7238  035106 000776 060360 115675          .WORD   ^0000776,^0060360,^0115675,^0000402,^0062400,^0000420,^0063701,^0116311
      035114 000402 062400 000420
      035122 063701 116311
7239  035126 010171 043231 063071          .WORD   ^0010171,^0043231,^0063071,^0000401,^0063310,^0104415,^0023640,^0060400
      035134 000401 063310 104415
      035142 023640 060400
7240  035146 103447 010154 000442          .WORD   ^0103447,^0010154,^0000442,^0076670,^0076611,^0076615,^0076617,^0000461
      035154 076670 076611 076615
      035162 076617 000461
7241  035166 010210 002756 003004          .WORD   ^0010210,^0002756,^0003004,^0010017,^0104443,^0000402,^0063004,^0023140
      035174 010017 104443 000402
      035202 063004 023140
7242  035206 062006 023160 062107          .WORD   ^0062006,^0023160,^0062107,^0101271,^0060601,^0116340,^0070604,^0117346
      035214 101271 060601 116340
      035222 070604 117346
7243  035226 070604 117746 056222          .WORD   ^0070604,^0117746,^0056222,^0056223,^0123200,^0104640,^0010210,^0002455
      035234 056223 123200 104640
      035242 010210 002455
7244  035246 114644 000600 063301          .WORD   ^0114644,^0000600,^0063301,^0000604,^0063230,^0000400,^0063222,^0100663
      035254 000604 063230 000400
      035262 063222 100663
7245  035266 060601 103746 000610          .WORD   ^0060601,^0103746,^0000610,^0063301,^0100746,^0000000,^0000000,^0000000
      035274 063301 100746 000000
      035302 000000 000000
7246  035306 000000 000000 000000          .WORD   ^0000000,^0000000,^0000000,^0000000,^0000000,^0000000,^0000000,^0000000
      035314 000000 000000 000000
      035322 000000 000000
7247  035326                         MCENDL:                      ;END OF KMS11 FIRMWARE IMAGE
7248        004000                   MCSIZL==.-MCBEGL             ;SIZE OF KMS11 FIRMWARE IMAGE
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 24
CZKMUA.P11    30-MAR-82 09:13          INITIALIZE SECTION

 7250
 7251
 7263
 7264  035326  047045  051445  022463  FIRMLD: .ASCIZ  /%N%S3%AFAILURE LOADING FIRMWARE/
       035334  043101  044501  052514
       035342  042522  046040  040517
       035350  044504  043516  043040
       035356  051111  053515  051101
       035364  000105
 7265  035366  047045  022462  031523  DATAHD: .ASCIZ  /%N2%S3%AADDRESS%S3%ALOADED%S3%AREAD BACK/
       035374  040445  042101  051104
       035402  051505  022523  031523
       035410  040445  047514  042101
       035416  042105  051445  022463
       035424  051101  040505  020104
       035432  040502  045503     000
 7266  035437     045  022516  031523  DATALD: .ASCIZ  /%N%S3%06%S4%06%S3%06/
       035444  047445  022466  032123
       035452  047445  022466  031523
       035460  047445  000066
 7267
 7268                                  .EVEN
 7269
 7270  035464                          ENDINIT
  (3)  035464                                                  L10013:
  (3)  035464  104411                                                  TRAP    C$INIT
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 24-1
CZKMUA.P11    30-MAR-82 09:13            AUTODROP SECTION

  7272                                   .SBTTL  AUTODROP SECTION
  7273
  7274                                   ;++
  7275                                   ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
  7276                                   ; THE "ADR" FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
  7277                                   ; SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
  7278                                   ; DROPPED FROM TESTING.
  7279                                   ;--
  7280
  7281  035466                                   BGNAUTO
   (3)  035466                                                                  L$AUTO::
  7282
  7289
  7290  035466                                   ENDAUTO
   (3)  035466                                                                  L10014:
   (3)  035466  104461                                                                  TRAP    C$AUTO
```

```
                                .SBTTL  CLEANUP CODING SECTION

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

                                        BGNCLN
                                                                                L$CLEAN::
                                        CLR     @CLKCSR         ;DISABLE CLOCK
                                        SETPRI  #PRI07          ;SET PROCESSOR PRIORITY BACK TO 7
                                                                                MOV     #PRI07,R0
                                                                                TRAP    C$SPRI
                                        CMP     #EXIT,KEYWD1    ;'EXIT" COMMAND ?
                                        BEQ     EXITCLN         ;YES,BRANCH
                                                                ;PRINT REPORT
                                ;;^C WAS ENTERED-- LOG IT
                                        MOV     #ABO,TEMP       ;EVENT TYPE
                                        MOV     NOBUF,TEMP4     ;;BUFFER NOT AVAILABLE
                                        MOV     PSCNT,TEMP2     ;;PASSES
                                        MOV     ERRCNT,TEMP3    ;;ERRORS
                                        CALL    LOGS5           ;GO LOG IT
                                EXITCLN:EXIT    CLN
                                                                                TRAP    C$EXIT
                                                                                .WORD   L10015-.


                                        .EVEN

                                        ENDCLN
                                                                        L10015:
                                                                                TRAP    C$CLEAN
```

```
 7339                                   .SBTTL  DROP UNIT SECTION
 7340
 7341                                   ;++
 7342                                   ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
 7343                                   ; TO NO LONGER BE TESTED.
 7344                                   ;--
 7345
 7346  035554                                   BGNDU                                                  L$DU::
  (3)  035554
 7347
 7356
 7357  035554                                   EXIT    DU                                                             .WORD   J$JMP
  (4)  035554  000167                                                                                                    .WORD   L10016-2-.
  (3)  035556  000000
 7358
 7370
 7371                                   .EVEN
 7372
 7373  035560                                   ENDDU                                                  L10016:
  (3)  035560                                                                                                            TRAP    C$DU
  (3)  035560  104453
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 24-4
CZKMUA.P11    30-MAR-82 09:13           ADD UNIT SECTION

  7375                                  .SBTTL  ADD UNIT SECTION
  7376
  7377                                  ;++
  7378                                  ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
  7379                                  ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
  7380                                  ;  TO THE TEST CYCLE.
  7381                                  ;--
  7382
  7383  035562                                  BGNAU
   (3)  035562                                                                  L$AU::
  7384
  7393
  7394  035562                                  EXIT    AU
   (4)  035562  000167                                                                  .WORD   J$JMP
   (3)  035564  000000                                                                  .WORD   L10017-2-.
  7395
  7407
  7408                                  .EVEN
  7409
  7410  035566                                  ENDAU
   (3)  035566                                                                  L10017:
   (3)  035566  104452                                                                  TRAP    C$AU
  7411
  7412
```

```
 7414                                   .SBTTL  TEST 1:  SETUP AND MODES OF OPERATION
 7415
 7416
 7417                                   ;++
 7418                                   ; TEST TO DETECT FAULTS IN THE DATA COMMUNICATION LINK.  THIS TEST WILL
 7419                                   ; THE PROVIDE COVERAGE NECESSARY TO ISOLATE FAILURES TO THE COMPUTER
 7420                                   ; EQUIPMENT, THE COMMUNICATION LINK, OR THE MODEM.
 7421                                   ;--
 7422
 7429
 7435
 7436  035570                                   BGNTST
   (3) 035570                                                                                    T1::
 7437
 7443
 7444                                   .SBTTL          PROGRAM SETUP SECTION
 7445
 7446  035570  013777  007350  151542           MOV     CLKEN,@CLKCSR   ;ENABLE THE CLOCK
 7447
 7448  035576                         GTXRXB:
 7449  035576  005001                 GTRA2:  CLR     R1
 7450  035600  012737  000001  007360         MOV     #1,TIMER1       ;SET TIMER TO COUNT 1 TICK
 7451  035606  005737  007360         1$:     TST     TIMER1          ;CHECK FOR IT TO BE COUNTED OFF
 7452  035612  001412                         BEQ     GTRA3           ;BRANCH IF CLOCK EXISTS (COUNTED A TICK)
 7453  035614  005301                         DEC     R1
 7454  035616  001373                         BNE     1$              ;KEEP CHECKING UNTIL R1 DOES FULL COUNTDWN
 7455  035620                               PRINTF  #NOCLK          ;PRINT BAD CLK MSG AND WARN OF HANG IF TIMEOUT
   (7) 035620  012746  014714                                                                    MOV     #NOCLK,-(SP)
   (6) 035624  012746  000001                                                                    MOV     #1,-(SP)
   (3) 035630  010600                                                                            MOV     SP,R0
   (4) 035632  104417                                                                            TRAP    C$PNTF
   (4) 035634  062706  000004                                                                    ADD     #4,SP
 7456
 7457  035640  005737  007304         GTRA3:  TST     RESFLG          ;SEE IF HERE AFTER A RESTART.
 7458  035644  001117                         BNE     GTRA5           ;BR IF HERE CAUSE OF A RESTART
 7459
 7460                                         ; CLEAR COUNTS AND SET UP DEFAULTS
 7461
 7462  035646  005037  007240         GTRA4:  CLR     TOTCC           ;CLEAR TOTAL CHAR. COUNT TEMP. LOC.
 7463  035652  005037  007174                 CLR     TTOTCC          ; CLEAR TOTAL CHAR. COUNT FOR TX BUFF
 7464  035656  005037  007160                 CLR     CTOTCC          ; CLEAR TOTAL CHAR. COUNT FOR CMP BUFF
 7465  035662  012701  006400                 MOV     #PTRTAB,R1      ;INIT TRANSMIT MESSAGE POINTER
 7466  035666  010137  007152                 MOV     R1,TXPTR
 7467  035672  005037  007150                 CLR     RXPTR           ; ZERO RX POINTER
 7468  035676  012702  000017                 MOV     #MSGLIM,R2
 7469  035702  006302                         ASL     R2
 7470  035704  006302                         ASL     R2
 7471  035706  010137  007154                 MOV     R1,CMPPTR
 7472  035712  060237  007154                 ADD     R2,CMPPTR       ;INIT COMPARE MESSAGE POINTER
 7473
 7474  035716  012737  000005  007226         MOV     #5,MSGTYP       ;SET UP DEFAULT MSG TYPE (QUICK FOX - ITEP MSG)
 7475  035724  013737  002162  007230         MOV     MSG5C,CURCC     ;SET UP DEFAULT CHAR COUNT
 7476  035732  012737  003400  007176         MOV     #TXBUF,TCURAD   ;SET UP CURRENT ADD TO START OF TX BUFFER
 7477  035740  012737  005400  007162         MOV     #CMPBUF,CCURAD  ;SET UP CURRENT ADD TO START OF CMP BUFFER
 7478
 7479  035746  013737  007176  007236         MOV     TCURAD,CURADD   ;SETUP CURRENT ADDR TO START OF TXBUF
```

```
 7480  035754  013737  007152  007234          MOV     TXPTR,CPTR      ;SETUP CURRENT POINTER TABLE POINTER FOR TXBUF
 7481  035762  004737  026320                  JSR     PC,BLDBUF       ; GO BUILD POINTER TABLE AND BUFFER
 7482  035766  012737  000001  007172          MOV     #1,TXMTOT       ;BUMP TOTAL MESSAGE COUNT
 7483
 7484  035774  013737  007154  007234          MOV     CMPPTR,CPTR     ;SET UP START OF COMPARE POINTER TABLE
 7485  036002  013737  007162  007236          MOV     CCURAD,CURADD   ;SET UP CURRENT ADDR. TO START OF CMPBUF
 7486  036010  012737  000005  007226          MOV     #5,MSGTYP
 7487  036016  013737  002162  007230          MOV     MSG5C,CURCC
 7488  036024  004737  026320                  JSR     PC,BLDBUF       ;PUT DEFAULT MESSAGE INTO CMPBUF
 7489  036030  012737  000001  007156          MOV     #1,CMPTOT       ;BUMP THE COMP MESG COUNT
 7490  036036  012737  000003  007306          MOV     #ACT,MODTYP     ;SET DEFAULT MODE= ACTIVE
 7491  036044  005037  007310                  CLR     MLTYP           ;SET DEFAULT MAINTENANCE LOOP MODE =NONE
 7492  036050  012737  000001  007316          MOV     #1,RPASS        ;SET UP DEFAULT 'RUN PASS" COUNT TO 1
 7493  036056  012737  000002  007314          MOV     #2,PARAM        ;SET UP PROG. PARAMETERS - DATACHECKING ENABLED
 7494                                                                  ;                  OPERATOR STATUS MSGS. PRINT OFF
 7495  036064                                  PRINTF  #HLP0
   (7) 036064  012746  012761                                                          MOV     #HLP0,-(SP)
   (6) 036070  012746  000001                                                          MOV     #1,-(SP)
   (3) 036074  010600                                                                  MOV     SP,R0
   (4) 036076  104417                                                                  TRAP    C$PNTF
   (4) 036100  062706  000004                                                          ADD     #4,SP
 7496  036104  013737  007306  010472  GTRA5:  MOV     MODTYP,DEV1
 7497  036112  013737  007310  010474          MOV     MLTYP,DEV2
 7498  036120  013737  007316  010476          MOV     RPASS,DEV3
 7499  036126  013737  007314  010500          MOV     PARAM,DEV4
 7500  036134  004737  026650                  JSR     PC,SHWOP        ;PRINT TO OPERATOR THE CURRENT MODE.....
 7501
 7502  036140                                  MANUAL                  ;SEE IF MANUAL INTERVENTION ALLOWED
   (3) 036140  104450                                                                  TRAP    C$MANI
 7503  036142                                  BCOMPLETE       GETCL   ;  BR IF YES (UAM=0 AND NOT CHAINED)
   (2) 036142  103412                                                                  BCS     GETCL
 7504  036144  005737  007316                  TST     RPASS           ;SEE IF THIS IS FIRST 'DCLT PASS"
 7505  036150  001002                          BNE     1$              ; BR IF NOT COMPLETED 1 PASS
 7506  036152                                  EXIT    TST             ; IF DONE 1 PASS IN UNATTENDED MODE - EXIT
   (3) 036152  104432                                                                  TRAP    C$EXIT
   (3) 036154  010106                                                                  .WORD   L10020-.
 7507  036156  012737  000001  007310  1$:     MOV     #TTL,MLTYP      ;SET UP DEFAULT FOR UNATTENDED MODE
 7508  036164  000137  041214                  JMP     GTR9            ;  'R M=ACT/LO=I/PAS=1/NOST/CH" AND RUN
 7509
 7510                                  .SBTTL          COMMAND LINE FETCH & INTERPRETATION SECTION
 7511
 7512  036170  105037  003377          GETCL:  CLRB    P$GDBD          ;CLEAR CMD LINE PARSING ERROR FLAGS
 7513  036174  105037  003376                  CLRB    P$NNUF
 7514  036200                                  GMANID  CLI$PM,CMDBUF,A,-1,1,72.,NO     ;GET A COMMAND LINE FROM OPR.
   (3) 036200  104443                                                                  TRAP    C$GMAN
   (3) 036202  000406                                                                  BR      10000$
   (4) 036204  003060                                                                  .WORD   CMDBUF
   (5) 036206  000142                                                                  .WORD   T$CODE
   (5) 036210  012310                                                                  .WORD   CLI$PM
   (5) 036212  177777                                                                  .WORD   -1
   (5) 036214  000001                                                                  .WORD   T$LOLIM
   (5) 036216  000110                                                                  .WORD   T$HILIM
   (3) 036220                                                                  10000$:
 7515  036220  012737  003060  003362          MOV     #CMDBUF,P$BUFA
 7516  036226  012737  010502  003364          MOV     #CLITRE,P$TREE
 7517  036234  012737  037162  003366          MOV     #CLIACT,P$ACT
```

```
 7518  036242  005037  003204                  CLR     QUALFG                  ;CLEAR QUALIFIER FLAG LOCATION
 7519  036246  004737  027152                  JSR     PC,P$TRV                ;GO PARSE COMMAND LINE
 7520  036252  105737  003377                  TSTB    P$GDBD                  ;SEE IF PARSED OK OR AN ERROR
 7521  036256  001412                          BEQ     1$
 7522  036260                                  PRINTF  #CLIERM
   (7) 036260  012746  012324                                                          MOV     #CLIERM,-(SP)
   (6) 036264  012746  000001                                                          MOV     #1,-(SP)
   (3) 036270  010600                                                                  MOV     SP,R0
   (4) 036272  104417                                                                  TRAP    C$PNTF
   (4) 036274  062706  000004                                                          ADD     #4,SP
 7523  036300  000137  036170                  JMP     GETCL
 7524  036304  105737  003376          1$:     TSTB    P$NNUF                  ;SEE IF INCOMPLETE COMMAND TYPED
 7525  036310  001412                          BEQ     10$
 7526  036312                                  PRINTF  #CLINUF
   (7) 036312  012746  012354                                                          MOV     #CLINUF,-(SP)
   (6) 036316  012746  000001                                                          MOV     #1,-(SP)
   (3) 036322  010600                                                                  MOV     SP,R0
   (4) 036324  104417                                                                  TRAP    C$PNTF
   (4) 036326  062706  000004                                                          ADD     #4,SP
 7527  036332  000137  036170                  JMP     GETCL
 7528
 7529  036336  023727  003202  000060  10$:     CMP     KEYWD1,#SETET           ;WAS "SET EXPECT = TRANMIT" ENTERED?
 7530  036344  001711                          BEQ     GETCL                   ;YES,BRANCH
 7531
 7532  036346  023727  003202  000005          CMP     KEYWD1,#HLP             ;SEE IF HELP WAS TYPED
 7533  036354  001705                          BEQ     GETCL                   ;GO GET CMD AGAIN IF YES
 7534  036356  023727  003202  000055          CMP     KEYWD1,#PRNT            ;SEE IF PRINT WAS TYPED
 7535  036364  001701                          BEQ     GETCL                   ; GO GET CMD AGAIN IF YES
 7536  036366  023727  003202  000004          CMP     KEYWD1,#RUN             ;SEE IF RUN WAS TYPED
 7537  036374  001002                          BNE     11$                     ; BR IF NO
 7538  036376  000137  041214                  JMP     GTR9                    ; START EXEC. IF YES
 7539  036402  023727  003202  000052  11$:     CMP     KEYWD1,#DMPS            ;SEE IF DUMP WAS TYPED
 7540  036410  001004                          BNE     12$                     ; BR IF NO
 7541  036412  004737  026064                  JSR     PC,DUMPSR               ; ELSE, DUMP PART OF MEMORY
 7542  036416  000137  036170                  JMP     GETCL                   ; THEN RETURN TO GET ANOTHER CMD.
 7543  036422  023727  003202  000057  12$:     CMP     KEYWD1,#EXIT            ;EXIT COMMAND ?
 7544  036430  001005                          BNE     13$                     ;NO,BRANCH
 7545  036432  012737  000001  007302          MOV     #1,DCLFLG               ;SET CLEANUP & EXIT FLAG
 7546  036440                                  EXIT    TST                     ;GO BACK TO INIT ROUTINE
   (3) 036440  104432                                                                  TRAP    C$EXIT
   (3) 036442  007620                                                                  .WORD   L10C20-.
 7547  036444  023727  003202  000001  13$:     CMP     KEYWD1,#CLEAR           ;SEE IF CLEAR WAS TYPED
 7548  036452  001646                          BEQ     GETCL                   ; IF YES, BACK TO GET ANOTHER CMD.
 7549  036454  023727  003202  000002          CMP     KEYWD1,#SHOW            ;SEE IF SHOW WAS TYPED
 7550  036462  001642                          BEQ     GETCL                   ; IF YES, BACK TO GET ANOTHER CMD.
 7551  036464  023727  003202  000010  4$:      CMP     KEYWD1,#SETEXP          ;SEE IF SET EXPECTED
 7552  036472  001512                          BEQ     2$                      ; BR IF YES (A SETEXP WAS TYPED)
 7553  036474  013737  007174  007240  5$:      MOV     TTOTCC,TOTCC
 7554  036502  023727  007240  001000          CMP     TOTCC,#BUFLIM           ;SEE IF BUFFER ALREADY FULL
 7555  036510  002414                          BLT     15$                     ; BR IF NOT FULL (BUFLIM # OF CHARS.)
 7556  036512                                  PRINTF  #MSGTRN,#BUFEX          ; ELSE TELL OPR. AND DON'T BUILD MSG.
   (8) 036512  012746  015035                                                          MOV     #BUFEX,-(SP)
   (7) 036516  012746  015053                                                          MOV     #MSGTRN,-(SP)
   (6) 036522  012746  000002                                                          MOV     #2,-(SP)
   (3) 036526  010600                                                                  MOV     SP,R0
   (4) 036530  104417                                                                  TRAP    C$PNTF
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 24-8
CZKMUA.P11    30-MAR-82 09:13                    COMMAND LINE FETCH & INTERPRETATION SECTION

  (4)  036532  062706  000006                                                          ADD     #6,SP
 7557  036536  000137  036170                    JMP     GETCL                 ; THEN GO GET A NEW COMMAND
 7558  036542  005737  007174          15$:      TST     TTOTCC                ;IF FIRST "SET" THEN GET RID OF DEFAULT
 7559  036546  001002                            BNE     6$
 7560  036550  005037  007172                    CLR     TXMTOT
 7561  036554  012737  006400  007152  6$:       MOV     #PTRTAB,TXPTR         ;GET POSITION OF END OF TX LIST
 7562  036562  013701  007172                    MOV     TXMTOT,R1
 7563  036566  020127  000017                    CMP     R1,#MSGLIM            ;SEE IF MSG COUNT EXCEEDED.
 7564  036572  002414                            BLT     17$                   ; BR IF NO
 7565  036574                                    PRINTF  #MSGTRN,#TABEX        ; ELSE TELL OPR. AND DON'T BUILD MSG.
  (8)  036574  012746  014775                                                          MOV     #TABEX,-(SP)
  (7)  036600  012746  015053                                                          MOV     #MSGTRN,-(SP)
  (6)  036604  012746  000002                                                          MOV     #2,-(SP)
  (3)  036610  010600                                                                  MOV     SP,R0
  (4)  036612  104417                                                                  TRAP    C$PNTF
  (4)  036614  062706  000006                                                          ADD     #6,SP
 7566  036620  000137  036170                    JMP     GETCL                 ; THEN GO GET A NEW COMMAND.
 7567  036624  006301          17$:      ASL     R1                    ;# OF MSGS *4 = NEXT FREE PTR BLOCK
 7568  036626  006301                    ASL     R1
 7569  036630  060137  007152                    ADD     R1,TXPTR
 7570  036634  013737  007152  007234            MOV     TXPTR,CPTR            ;SETUP CHAR. COUNT, CURRENT ADDR, & PTR
 7571  036642  013737  007176  007236            MOV     TCURAD,CURADD
 7572  036650  004737  026222                    JSR     PC,ADDCC              ;ADD IN CHAR. COUNT AND CHECK TOTAL
 7573  036654  004737  026320                    JSR     PC,BLDBUF             ;GO BUILD MESSAGE IN BUFFER AND PTRS.
 7574  036660  013737  007234  007152            MOV     CPTR,TXPTR
 7575  036666  013737  007240  007174            MOV     TOTCC,TTOTCC          ;UPDATE CHAR. COUNT, CURR ADDR, & PTR
 7576  036674  013737  007236  007176            MOV     CURADD,TCURAD
 7577  036702  005237  007172                    INC     TXMTOT
 7578  036706  005337  003206                    DEC     QUALVL                ;DEC THE COPY COUNT
 7579  036712  001270                            BNE     5$
 7580  036714  000137  036170                    JMP     GETCL
 7581
 7582  036720  013737  007160  007240  2$:       MOV     CTOTCC,TOTCC          ;SETUP CHAR. COUNT, CURR. ADDR. & PTR
 7583  036726  023727  007240  001000            CMP     TOTCC,#BUFLIM         ;SEE IF BUFFER ALREADY FULL
 7584  036734  002414                            BLT     16$                   ; BR IF NOT FULL (BUFLIM # OF CHARS.)
 7585  036736                                    PRINTF  #MSGTRN,#BUFEX        ; ELSE TELL OPR. AND DON'T BUILD MSG.
  (8)  036736  012746  015035                                                          MOV     #BUFEX,-(SP)
  (7)  036742  012746  015053                                                          MOV     #MSGTRN,-(SP)
  (6)  036746  012746  000002                                                          MOV     #2,-(SP)
  (3)  036752  010600                                                                  MOV     SP,R0
  (4)  036754  104417                                                                  TRAP    C$PNTF
  (4)  036756  062706  000006                                                          ADD     #6,SP
 7586  036762  000137  036170                    JMP     GETCL                 ; THEN GO GET A NEW COMMAND
 7587  036766  005737  007160          16$:      TST     CTOTCC                ;IF FIRST "SET" THEN GET RID OF DEFAULT
 7588  036772  001002                            BNE     7$
 7589  036774  005037  007156                    CLR     CMPTOT
 7590  037000  012701  006400          7$:       MOV     #PTRTAB,R1
 7591  037004  012702  000017                    MOV     #MSGLIM,R2
 7592  037010  006302                            ASL     R2
 7593  037012  006302                            ASL     R2
 7594  037014  010137  007154                    MOV     R1,CMPPTR
 7595  037020  060237  007154                    ADD     R2,CMPPTR             ;INIT COMPARE MESSAGE POINTER
 7596  037024  013701  007156                    MOV     CMPTOT,R1
 7597  037030  020127  000017                    CMP     R1,#MSGLIM            ;SEE IF MSG COUNT EXCEEDED.
 7598  037034  002414                            BLT     18$                   ; BR IF NO
 7599  037036                                    PRINTF  #MSGTRN,#TABEX        ; ELSE TELL OPR. AND DON'T BUILD MSG.
```

```
 (8)  037036  012746  014775                                                    MOV     #TABEX,-(SP)
 (7)  037042  012746  015053                                                    MOV     #MSGTRN,-(SP)
 (6)  037046  012746  000002                                                    MOV     #2,-(SP)
 (3)  037052  010600                                                            MOV     SP,R0
 (4)  037054  104417                                                            TRAP    C$PNTF
 (4)  037056  062706  000006                                                    ADD     #6,SP
7600  037062  000137  036170                 JMP     GETCL          ; THEN GO GET A NEW COMMAND.
7601  037066  006301              18$:       ASL     R1             ;# OF MSGS *4 = NEXT FREE PTR BLOCK
7602  037070  006301                         ASL     R1
7603  037072  060137  007154                 ADD     R1,CMPPTR
7604  037076  013737  007154  007234         MOV     CMPPTR,CPTR
7605  037104  013737  007162  007236         MOV     CCURAD,CURADD
7606  037112  004737  026222                 JSR     PC,ADDCC       ;ADD IN XHAR. COUNT AND CHECK TOTAL
7607  037116  004737  026320                 JSR     PC,BLDBUF
7608  037122  013737  007234  007154         MOV     CPTR,CMPPTR
7609  037130  005237  007156                 INC     CMPTOT
7610  037134  013737  007236  007162         MOV     CURADD,CCURAD  ;UPDATE CHAR. COUNT, CURR ADDRR. & PTR
7611  037142  013737  007240  007160         MOV     TOTCC,CTOTCC
7612  037150  005337  003206                 DEC     QUALVL         ;IF COPY WAS GIVEN, PUT MSG IN BUFF
7613  037154  001261                         BNE     2$             ; AGAIN
7614  037156  000137  036170                 JMP     GETCL          ;GO BACK UNTIL GET A 'RUN''
7615
7616
7617
7618
7619
```

```
                                        .SBTTL          ACTION TABLE AND ROUTINES
                                        ;       USER MUST CLEAR/SET P$GDBD IF USE "CLIBIF" IN CONNECTION WITH ACTION
                                        ;       R2 WILL HOLD ACTION CODE FROM PARSING (CLI) NODE
                                        ;
037162                                  CLIACT:
037162  006302                                  ASL     R2              ;MULTIPLY ACTION CODE BY 2
037164  016202  037200                          MOV     10$(R2),R2      ;OFFSET VALUE
037170  062702  037200                          ADD     #10$,R2         ;ADD BASE VALUE
037174  004712                                  JSR     PC,(R2)         ;GO DO ACTION
037176  000207                                  RTS     PC              ;RETURN TO TRVACT:

                                                                        ;BRIEF DESCRIPTION OF ACTIONS TAKEN
037200  000150                          10$:    .WORD   ACTNUL-10$      ;NULL
037202  000152                                  .WORD   ACTCLR-10$      ;CLEAR
037204  000162                                  .WORD   ACTSHO-10$      ;SHOW
037206  001604                                  .WORD   ACTCHK-10$      ;CHECK
037210  000262                                  .WORD   ACTRUN-10$      ;RUN
037212  000172                                  .WORD   ACTHLP-10$      ;HELP
037214  000306                                  .WORD   ACTCSE-10$      ;CLEAR OR SHOW EXPECTED
037216  000434                                  .WORD   ACTCST-10$      ;CLEAR OR SHOW TRANSMIT
037220  000756                                  .WORD   ACTSTE-10$      ;SET EXPECTED
037222  000766                                  .WORD   ACTSTT-10$      ;SET TRANSMIT
037224  001004                                  .WORD   ACTSZE-10$      ;SIZE
037226  001014                                  .WORD   ACTCOP-10$      ;COPY
037230  001024                                  .WORD   ACTNUM-10$      ;NUMERIC VALUE FOR SIZE OR COPY
037232  001116                                  .WORD   ACTOPM-10$      ;QUOTED MESSAGE FROM USER
037234  001612                                  .WORD   ACTSTS-10$      ;STATUS
037236  001136                                  .WORD   ACTEQO-10$      ;END OF QUOTED MESSAGE FROM USER
037240  001216                                  .WORD   ACTMS0-10$      ;ONES
037242  001224                                  .WORD   ACTMS1-10$      ;ZEROS
037244  001234                                  .WORD   ACTMS2-10$      ;1ALT
037246  001244                                  .WORD   ACTMS3-10$      ;0ALT
037250  001254                                  .WORD   ACTMS4-10$      ;ITEP
037252  001264                                  .WORD   ACTMS5-10$      ;CCITT
037254  001302                                  .WORD   ACTMS6-10$      ;ALPHA
037256  001370                                  .WORD   ACTATV-10$      ;ACTIVE MODE
037260  001400                                  .WORD   ACTPAS-10$      ;PASSIVE MODE
037262  001420                                  .WORD   ACTREC-10$      ;RECEIVE MODE
037264  001426                                  .WORD   ACTLIS-10$      ;LISTEN MODE
037266  001436                                  .WORD   ACTDLL-10$      ;DOWNLINE LOAD
037270  001446                                  .WORD   ACTTRA-10$      ;TRANSMIT MODE
037272  001456                                  .WORD   ACTTAL-10$      ;TALK MODE
037274  001504                                  .WORD   ACTNO-10$       ;NO
037276  001514                                  .WORD   ACTECH-10$      ;ECHO
037300  001620                                  .WORD   ACTCRC-10$      ;SET CRC BIT
037302  001634                                  .WORD   ACTPRO-10$      ;SET PROTOCOL BIT
037304  001666                                  .WORD   ACTRPS-10$      ;STATUS
037306  001676                                  .WORD   ACTMOP-10$      ;REMOTE STATION IN MAINTENANCE LOOP MODE
037310  001706                                  .WORD   ACTTLP-10$      ;INTERNAL TTL
037312  001716                                  .WORD   ACTCLP-10$      ;CABLE LOOP
037314  001726                                  .WORD   ACTLLP-10$      ;LOCAL MODEM LOOP
037316  001736                                  .WORD   ACTRLP-10$      ;REMOTE MODEM LOOP
037320  000142                                  .WORD   ACTNUF-10$      ;MORE COMMAND LINE NEEDED
037322  001174                                  .WORD   ACTBCR-10$      ;BAD CHARACTER IN OPERATOR MESSAGE
037324  000712                                  .WORD   ACTDMS-10$      ;DUMP MEMORY START ADDRESS
037326  000742                                  .WORD   ACTDME-10$      ;DUMP MEMORY END ADDRESS
```

```
 7677  037330  000734                    .WORD   ACTDMQ-10$      ;DUMP WORD
 7678  037332  000246                    .WORD   ACTPRT-10$      ;PRINT
 7679  037334  001626                    .WORD   ACTMOS-10$      ;MODEM ACTION
 7680  037336  000236                    .WORD   ACTEXT-10$      ;EXIT ACTION
 7681  037340  001326                    .WORD   ACTSEX-10$      ;SET E=T ACTION
 7682
```

```
 7684
 7685  037342  112737  177777  003376  ACTNUF: MOVB    #-1,P$NNUF      ;SET FLAG TO SAY NEED MORE OF COMMAND
 7686  037350  000207                  ACTNUL: RTS     PC              ;RETURN TO PARSER
 7687
 7688  037352  012737  000001  003202  ACTCLR: MOV     #CLEAR,KEYWD1   ;SET LOC TO SAY A CLEAR WAS TYPED
 7689  037360  000207                          RTS     PC
 7690
 7691  037362  012737  000002  003202  ACTSHO: MOV     #SHOW,KEYWD1    ;SET LOC. TO SAY A SHOW WAS TYPED
 7692  037370  000207                          RTS     PC
 7693
 7694  037372  012702  003210          ACTHLP: MOV     #HLPTAB,R2      ;SETUP R2 AS A POINTER TO HELP MSG TABLE
 7695  037376                          1$:     PRINTF  #HLPF,(R2)+     ;PRINT HELP INFORMATION MESSAGES
   (8) 037376  012246                                                          MOV     (R2)+,-(SP)
   (7) 037400  012746  013037                                                  MOV     #HLPF,-(SP)
   (6) 037404  012746  000002                                                  MOV     #2,-(SP)
   (3) 037410  010600                                                          MOV     SP,R0
   (4) 037412  104417                                                          TRAP    C$PNTF
   (4) 037414  062706  000006                                                  ADD     #6,SP
 7696  037420  020227  003230                  CMP     R2,#HLPEND      ;SEE IF ALL INFO PRINTED YET
 7697  037424  001364                          BNE     1$              ;IF NO KEEP PRINTING
 7698  037426  012737  000005  003202          MOV     #HLP,KEYWD1     ;SET LOC. TO SAY A HELP WAS TYPED
 7699  037434  000207                          RTS     PC
 7700  037436  012737  000057  003202  ACTEXT: MOV     #EXIT,KEYWD1    ;EXIT COMMAND
 7701  037444  000207                          RTS     PC
 7702  037446  012737  000055  003202  ACTPRT: MOV     #PRNT,KEYWD1    ;SET LOC. TO SAY A HELP WAS TYPED
 7703  037454  004737  023604                  JSR     PC,REPORT       ;CALL ROUTINE TO PRINT EVENT LOG AND BASE TABLE
 7704  037460  000207                          RTS     PC
 7705
 7706  037462  012737  000004  003202  ACTRUN: MOV     #RUN,KEYWD1     ;SET RUN FLAG
 7707  037470  112737  177777  003376          MOVB    #-1,P$NNUF      ;SET FLAG TO SAY NEED MORE OF COMMAND
 7708  037476  012737  000001  007316          MOV     #1,RPASS        ;SET DEFAULT RUN "PASS" TO 1
 7709  037504  000207                          RTS     PC
 7710
 7711  037506  012701  006400          ACTCSE: MOV     #PTRTAB,R1
 7712  037512  012702  000017                  MOV     #MSGLIM,R2
 7713  037516  006302                          ASL     R2
 7714  037520  006302                          ASL     R2
 7715  037522  010137  007154                  MOV     R1,CMPPTR
 7716  037526  060237  007154                  ADD     R2,CMPPTR       ;INIT COMPARE MESSAGE POINTER
 7717  037532  013701  007154                  MOV     CMPPTR,R1
 7718
 7719  037536  013702  007156                  MOV     CMPTOT,R2
 7720  037542  105037  003376                  CLRB    P$NNUF          ;FLAG THAT HAVE VALID COMMAND AT THIS PT.
 7721  037546  023727  003202  000002          CMP     KEYWD1,#SHOW    ;SEE IF A CLEAR OR SHOW WAS TYPED
 7722  037554  001500                          BEQ     ACTSHW          ;BR IF A SHOW WAS TYPED
 7723  037556  012737  000001  007156          MOV     #1,CMPTOT       ;CLEAR COMPARE MESSAGE COUNT, CHAR. COUNT
 7724  037564  005037  007160                  CLR     CTOTCC          ; AND RESET POINTER
 7725
 7726  037570  012701  006400                  MOV     #PTRTAB,R1
 7727  037574  012702  000017                  MOV     #MSGLIM,R2
 7728  037600  006302                          ASL     R2
 7729  037602  006302                          ASL     R2
 7730  037604  010137  007154                  MOV     R1,CMPPTR
 7731  037610  060237  007154                  ADD     R2,CMPPTR       ;INIT COMPARE MESSAGE POINTER
 7732  037614  013737  007154  007234          MOV     CMPPTR,CPTR     ;SET UP TO FILL IN DEFAULT MESSAGE
 7733  057622  012701  005400                  MOV     #CMPBUF,R1
```

```
 7734  037626  010137  007162                 MOV     R1,CCURAD
 7735  037632  000431                         BR      ACTCLB
 7736
 7737  037634  012701  006400         ACTCST: MOV     #PTRTAB,R1
 7738  037640  013702  007172                 MOV     TXMTOT,R2
 7739  037644  105037  003376                 CLRB    P$NNUF          ;FLAG THAT HAVE VALID COMMAND AT THIS PT.
 7740  037650  023727  003202  000002         CMP     KEYWD1,#SHOW    ;SEE IF A CLEAR OR SHOW WAS TYPED
 7741  037656  001437                         BEQ     ACTSHW          ;BR IF A SHOW WAS TYPED
 7742  037660  012737  000001  007172         MOV     #1,TXMTOT       ;CLEAR TRANSMIT MESSAGE COUNT, CHAR. COUNT
 7743  037666  005037  007174                 CLR     TTOTCC          ; AND RESET POINTER
 7744  037672  012737  006400  007152         MOV     #PTRTAB,TXPTR
 7745  037700  013737  007152  007234         MOV     TXPTR,CPTR
 7746  037706  012701  003400                 MOV     #TXBUF,R1
 7747  037712  010137  007176                 MOV     R1,TCURAD
 7748
 7749  037716  012702  001000         ACTCLB: MOV     #BUFLIM,R2
 7750  037722  010137  007236                 MOV     R1,CURADD       ;SET UP TO PUT DEFAULT MSG IN LIST AFTER 033'S
 7751  037726  012737  000005  007226         MOV     #5,MSGTYP
 7752  037734  013737  002162  007230         MOV     MSG5C,CURCC
 7753  037742  105021                 1$:     CLRB    (R1)+           ;FILL EXPT OR TRAN BUFFER WITH 0'S IF A CLEAR
 7754  037744  005302                         DEC     R2              ;DO 'BUFLIM" NUMBER OF BYTE LOCATIONS
 7755  037746  001375                         BNE     1$
 7756  037750  004737  026320                 JSR     PC,BLDBUF       ;"CLEAR" REALLY MEANS TO PUT DEFAULT MSG IN
 7757  037754  000207                         RTS     PC              ;WHEN DONE, RETURN TO PARSER
 7758
 7759
 7760  037756  012705  003322         ACTSHW: MOV     #SHTAB,R5
 7761  037762  122571  000000         5$:     CMPB    (R5)+,@(R1)     ;LOOK AT FIRST BYTE OF MSG TO DECIPHER TYPE
 7762  037766  001404                         BEQ     6$
 7763  037770  020527  003331                 CMP     R5,#SHTEND      ;SEE IF LOOKED AT ALL OF DEFAULTS YET
 7764  037774  001372                         BNE     5$
 7765  037776  005205                         INC     R5              ;MUST BE OPR. SPEC'D THEN
 7766  040000  162705  003323         6$:     SUB     #SHTAB+1,R5
 7767  040004  006305                         ASL     R5
 7768  040006  016137  000002  007244         MOV     2(R1),TEMP
 7769  040014                                 PRINTF  #SHMSG,SHTYTB(R5),TEMP  ;PRINT MSG SIZE & TYPE
  (9)  040014  013746  007244                                                 MOV     TEMP,-(SP)
  (8)  040020  016546  003302                                                 MOV     SHTYTB(R5),-(SP)
  (7)  040024  012746  014242                                                 MOV     #SHMSG,-(SP)
  (6)  040030  012746  000003                                                 MOV     #3,-(SP)
  (3)  040034  010600                                                         MOV     SP,R0
  (4)  040036  104417                                                         TRAP    C$PNTF
  (4)  040040  062706  000010                                                 ADD     #10,SP
 7770  040044  062701  000004                 ADD     #4,R1           ;BUMP R1 TO NEXT SET OF POINTERS
 7771  040050  005302                         DEC     R2
 7772  040052  001341                         BNE     ACTSHW
 7773  040054  013737  007306  010472         MOV     MODTYP,DEV1
 7774  040062  013737  007310  010474         MOV     MLTYP,DEV2
 7775  040070  013737  007316  010476         MOV     RPASS,DEV3
 7776  040076  013737  007314  010500         MOV     PARAM,DEV4
 7777  040104  004737  026650                 JSR     PC,SHWOP        ;SHOW THE OPERATOR THE CURRENT MODE..... ALSO
 7778  040110  000207                         RTS     PC
 7779
 7780  040112  013737  003372  007220 ACTDMS: MOV     P$NUM,STADD     ;SETUP STARTING ADDRESS FOR DUMP
 7781  040120  005037  007224                 CLR     BYTBIT          ;SET DEFAULT OF WORD DUMP
 7782  040124  012737  000052  003202         MOV     #DMPS,KEYWD1    ;FLAG THAT A DUMP WAS TYPED
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 24-14
CZKMUA.P11    30-MAR-82 09:13                    ACTION TABLE AND ROUTINES

  7783  040132  000403                           BR      ACTDME
  7784
  7785  040134  012737  177777  007224  ACTDMQ: MOV     #-1,BYTBIT      ;SET DUMP FLAG TO 'DUMP-WORD''
  7786  040142  013737  003372  007222  ACTDME: MOV     P$NUM,ENADD     ;SETUP END ADDRESS FOR DUMP (=START IF NO 'EEE''
  7787  040150  105037  003376          ACTDMX: CLRB    P$NNUF          ;CLEAR NOT-ENOUGH FLAG, 'DUMP N-N/B'' IS VALID
  7788  040154  000207                          RTS     PC
  7789
```

```
 7791
 7792
 7793  040156  012737  000010  003202  ACTSTE: MOV     #SETEXP,KEYWD1
 7794  040164  000403                  BR      ACTSTX
 7795
 7796  040166  012737  000011  003202  ACTSTT: MOV     #SETTRN,KEYWD1
 7797  040174  012737  000001  003206  ACTSTX: MOV     #1,QUALVL       ;SET UP DEFAULT COPY TO 1 (/COPY=0)
 7798  040202  000207                  RTS     PC
 7799
 7800  040204  012737  000012  003204  ACTSZE: MOV     #SIZE,QUALFG
 7801  040212  000207                  RTS     PC
 7802
 7803  040214  012737  000013  003204  ACTCOP: MOV     #QCOPY,QUALFG
 7804  040222  000207                  RTS     PC
 7805
 7806  040224  023727  003204  000012  ACTNUM: CMP     QUALFG,#SIZE    ;SEE IF A SIZE OR COPY TYPED
 7807  040232  001023                  BNE     1$              ;BR IF IT WAS A COPY
 7808  040234  005737  003372          TST     P$NUM           ;CHECK TO BE SURE DIDN'T TRY SIZE=0
 7809  040240  001014                  BNE     3$              ; BR IF NO
 7810  040242                          PRINTF  #CLISE0
   (7) 040242  012746  012613                                                  MOV     #CLISE0,-(SP)
   (6) 040246  012746  000001                                                  MOV     #1,-(SP)
   (3) 040252  010600                                                          MOV     SP,R0
   (4) 040254  104417                                                          TRAP    C$PNTF
   (4) 040256  062706  000004                                                  ADD     #4,SP
 7811  040262  112737  177777  003377          MOVB    #-1,P$GDBD      ;SEE ERROR-IN-CMD FLAG
 7812  040270  000411                  BR      2$
 7813  040272  013737  003372  007230  3$:     MOV     P$NUM,CURCC     ;IF A SIZE LOAD CURCC WITH BYTE COUNT
 7814  040300  000405                  BR      2$
 7815  040302  013737  003372  003206  1$:     MOV     P$NUM,QUALVL    :IF A COPY, LOAD COPY COUNT
 7816  040310  005237  003206                  INC     QUALVL          ;INCREMENT SO FIRST DEC MAKES IT REAL #
 7817  040314  000522                  2$:     BR      ACTMEX
 7818
 7819  040316  012737  000007  007226  ACTOPM: MOV     #7,MSGTYP
 7820  040324  010437  007244                  MOV     R4,TEMP         ;KEEP TRACK OF START OF QUOTED TEXT
 7821  040330  005237  007244                  INC     TEMP            ; SO CAN CALC OPCNT AT END OF QUOTES
 7822  040334  000207                  RTS     PC
 7823
 7824  040336  010402                  ACTEQO: MOV     R4,R2
 7825  040340  163702  007244                  SUB     TEMP,R2
 7826  040344  010237  007230                  MOV     R2,CURCC        ;CALC BYTE COUNT FOR QUOTED TEXT
 7827  040350  010237  002166                  MOV     R2,OPCNT
 7828  040354  013701  007244                  MOV     TEMP,R1
 7829  040360  012705  002524                  MOV     #OPBUF,R5
 7830  040364  112125          1$:     MOVB    (R1)+,(R5)+     ;COPY QUOTED TEXT TO OPBUF
 7831  040366  005302                  DEC     R2
 7832  040370  001375                  BNE     1$
 7833  040372  000473                  BR      ACTMEX
 7834
 7835  040374                  ACTBCR: PRINTF  #CLIBCR         ;BAD CHAR. IN OPR. QUOTED STRING
   (7) 040374  012746  012546                                                  MOV     #CLIBCR,-(SP)
   (6) 040400  012746  000001                                                  MOV     #1,-(SP)
   (3) 040404  010600                                                          MOV     SP,R0
   (4) 040406  104417                                                          TRAP    C$PNTF
   (4) 040410  062706  000004                                                  ADD     #4,SP
 7836  040414  000207                  RTS     PC
```

```
 CZKMUAO KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 25-1
CZKMUA.P11    30-MAR-82 09:13                    ACTION TABLE AND ROUTINES

 7837                                            ;SET THE MESSAGE TYPE AS PER COMMAND LINE
 7838  040416  005037  007226          ACTMS0: CLR     MSGTYP
 7839  040422  000435                          BR      ACTME1
 7840  040424  012737  000001  007226  ACTMS1: MOV     #1,MSGTYP       ;ALL ONES
 7841  040432  000431                          BR      ACTME1
 7842  040434  012737  000002  007226  ACTMS2: MOV     #2,MSGTYP       ;ONES & ZEROS
 7843  040442  000425                          BR      ACTME1
 7844  040444  012737  000003  007226  ACTMS3: MOV     #3,MSGTYP       ;ZEROS & ONES
 7845  040452  000421                          BR      ACTME1
 7846  040454  012737  000004  007226  ACTMS4: MOV     #4,MSGTYP       ;CCITT
 7847  040462  000415                          BR      ACTME1
 7848  040464  012737  000005  007226  ACTMS5: MOV     #5,MSGTYP       ;QUICK FOX
 7849  040472  013737  002162  007230          MOV     MSG5C,CURCC             ;SETUP DEFAULT SIZE FOR THIS TYPE
 7850  040500  000430                          BR      ACTMEX
 7851  040502  012737  000006  007226  ACTMS6: MOV     #6,MSGTYP       ;ALPHA/NUM
 7852  040510  013737  002164  007230          MOV     MSG6C,CURCC             ;SETUP DEFAULT SIZE FOR THIS TYPE
 7853
 7854  040516  012737  000100  007230  ACTME1: MOV     #64.,CURCC              ;SETUP DEFAULT SIZE FOR MSG0-4
 7855  040524  000416                          BR      ACTMEX                  ;GO TO EXIT
 7856
 7857  040526  022737  000010  003202  ACTSEX: CMP     #SETEXP,KEYWD1          ;DID WE GET HERE FROM "SET E =" COMMAND?
 7858  040534  001404                          BEQ     10$                     ;YES,BRANCH
 7859  040536  112737  177777  003377          MOVB    #-1,P$GDBD              ;SET ERROR FLAG
 7860  040544  000406                          BR      ACTMEX                  ;GO TO EXIT
 7861  040546  004737  026444          10$:    JSR     PC,FACSIMILE            ;GO COPY TRANMIT BUFFER TO EXPECT BUFFER
 7862  040552  012737  000060  003202          MOV     #SETET,KEYWD1           ;SET FLAG TO BE USED IN T1::
 7863  040560  000400                          BR      ACTMEX                  ;GO TO EXIT
 7864
 7865  040562  105037  003376  ACTMEX: CLRB    P$NNUF                  ;CLEAR NOT-ENOUGH FLAG
 7866  040566  000207                          RTS     PC
 7867
```

```
 7869  040570  012737  000003  007306  ACTATV: MOV     #ACT,MODTYP
 7870  040576  000432                  BR      ACTM2X
 7871
 7872  040600  012737  000002  007306  ACTPAS: MOV     #PAS,MODTYP
 7873  040606  105037  003376                  CLRB    P$NNUF                  ;CLEAR NOT-ENOUGH FLAG
 7874  040612  005037  007310                  CLR     MLTYP                   ;CLEAR MAINT LOOP TYPE
 7875  040616  000207                  RTS     PC
 7876
 7877  040620  005037  007306          ACTREC: CLR     MODTYP
 7878  040624  000417                  E?      ACTM2X
 7879
 7880  040626  012737  000006  007306  ACTLIS: MOV     #LIS,MODTYP
 7881  040634  000413                  BR      ACTM2X
 7882
 7883  040636  012737  000004  007306  ACTDLL: MOV     #DOW,MODTYP
 7884  040644  000407                  BR      ACTM2X
 7885
 7886  040646  012737  000001  007306  ACTTRA: MOV     #TRA,MODTYP
 7887  040654  000403                  BR      ACTM2X
 7888
 7889  040656  012737  000005  007306  ACTTAL: MOV     #TAL,MODTYP
 7890
 7891  040664  042737  000004  007314  ACTM2X: BIC     #ECHOB,PARAM            ;DISABLE /ECHO (ALL BUT PASSIVE MODE)
 7892  040672  105037  003376                  CLRB    P$NNUF                  ;CLEAR NOT-ENOUGH FLAG
 7893  040676  005037  007310                  CLR     MLTYP                   ;CLEAR MAINT LOOP TYPE
 7894  040702  000207                  RTS     PC
 7895
```

```
 7897  040704  012737  000036  003204  ACTNO:  MOV     #NO,QUALFG
 7898  040712  000207                  RTS     PC
 7899
 7900  040714  022737  000036  003204  ACTECH: CMP     #NO,QUALFG
 7901  040722  001422                  BEQ     1$
 7902  040724  052737  000004  007314          BIS     #ECHOB,PARAM
 7903  040732  022737  000002  007306          CMP     #PAS,MODTYP             ;BE SURE IN PASSIVE MODE IF
 7904  040740  001416                  BEQ     2$                              ;IF TRYING TO SET /ECHO
 7905  040742                          PRINTF  #CLINPS
   (7) 040742  012746  012503                                                          MOV     #CLINPS,-(SP)
   (6) 040746  012746  000001                                                          MOV     #1,-(SP)
   (3) 040752  010600                                                                  MOV     SP,R0
   (4) 040754  104417                                                                  TRAP    C$PNTF
   (4) 040756  062706  000004                                                          ADD     #4,SP
 7906  040762  112737  177777  003377          MOVB    #-1,P$GDBD
 7907  040770  042737  000004  007314  1$:     BIC     #ECHOB,PARAM
 7908  040776  005037  003204          2$:     CLR     QUALFG                  ;CLEAR 'NO' OUT OF QUALIFIER FLAG
 7909  041002  000501                  BR      ACTLXX
 7910
 7911  041004  012701  000002          ACTCHK: MOV     #DATCKB,R1              ;SET DATA CHECK BIT
 7912  041010  000413                  BR      ACTQFG
 7913
 7914  041012  012701  000001          ACTSTS: MOV     #STATB,R1       ;SET THE STATUS BIT
 7915  041016  000410                  BR      ACTQFG
 7916
 7917  041020  012701  000020          ACTCRC: MOV     #CRCB,R1        ;SET THE CRC BIT
 7918  041024  000405                  BR      ACTQFG
 7919
 7920  041026  012701  000010          ACTMOS: MOV     #MOCHK,R1       ;MODEM BIT
 7921  041032  000402                  BR      ACTQFG
 7922
 7923  041034  012701  000040          ACTPRO: MOV     #PROTOB,R1      ;SET THE PROTOCOL BIT
 7924
 7925  041040  050137  007314          ACTQFG: BIS     R1,PARAM
 7926  041044  022737  000036  003204          CMP     #NO,QUALFG
 7927  041052  001002                  BNE     1$
 7928  041054  040137  007314                  BIC     R1,PARAM
 7929  041060  005037  003204          1$:     CLR     QUALFG                  ;CLEAR 'NO' OUT OF QUALIFIER FLAG
 7930  041064  000450                  BR      ACTLXX
 7931
 7932  041066  013737  003372  007316  ACTRPS: MOV     P$NUM,RPASS             ;GET NUMBER OF 'RUN PASSES'
 7933  041074  000444                  BR      ACTLXX
 7934
 7935  041076  012737  000005  007310  ACTMOP: MOV     #5,MLTYP
 7936  041104  000417                  BR      ACTLPX
 7937  041106  012737  000001  007310  ACTTLP: MOV     #1,MLTYP
 7938  041114  000413                  BR      ACTLPX
 7939  041116  012737  000002  007310  ACTCLP: MOV     #2,MLTYP
 7940  041124  000407                  BR      ACTLPX
 7941  041126  012737  000003  007310  ACTLLP: MOV     #3,MLTYP
 7942  041134  000403                  BR      ACTLPX
 7943  041136  012737  000004  007310  ACTRLP: MOV     #4,MLTYP
 7944
 7945  041144  022737  000003  007306  ACTLPX: CMP     #ACT,MODTYP     ;BE SURE IN ACTIVE IF TRYING TO SET LOOP
 7946  041152  001415                  BEQ     ACTLXX          ; BR IF IN ACTIVE
 7947  041154  112737  177777  003377          MOVB    #-1,P$GDBD
```

```
 7948  041162  005037  007310                  CLR     MLTYP           ;CLEAR ANY LOOP TYPE THAT MAY HAVE GOT SET
 7949  041166                                  PRINTF  #CLIBDL
  (7)  041166  012746  012441                                                  MOV     #CLIBDL -(SP)
  (6)  041172  012746  000001                                                  MOV     #1,-(SP)
  (3)  041176  010600                                                          MOV     SP,R0
  (4)  041200  104417                                                          TRAP    C$PNTF
  (4)  041202  062706  000004                                                  ADD     #4,SP
 7950  041206  105037  003376          ACTLXX: CLRB    P$NNUF          ;CLEAR NOT-ENOUGH FLAG
 7951  041212  000207                          RTS     PC
 7952
```

```
7954
7955  041214  005737  007310          GTR9:   TST     MLTYP           ;LOOP MODE ?
7956  041220  001422                          BEQ     10$             ;NO,BRANCH
7957  041222  032737  000002  007314          BIT     #DATCKB,PARAM   ;DATA CHECK ?
7958  041230  001416                          BEQ     10$             ;NO,BRANCH
7959  041232  023737  007156  007172          CMP     CMPTOT,TXMTOT   ;TX & EX EQUAL
7960  041240  001412                          BEQ     10$             ;YES,BRANCH
7961  041242                                  PRINTF  #CLIPW          ;PRINT WARNING
  (7) 041242  012746  012644                                                      MOV     #CLIPW,-(SP)
  (6) 041246  012746  000001                                                      MOV     #1,-(SP)
  (3) 041252  010600                                                              MOV     SP,R0
  (4) 041254  104417                                                              TRAP    C$PNTF
  (4) 041256  062706  000004                                                      ADD     #4,SP
7962  041262  000137  036170                  JMP     GETCL           ;TRY AGAIN
7963
7964                                          ;RX ALLOCATE CODE
7965  041266  012701  006400          10$:    MOV     #PTRTAB,R1      ;INIT TRANSMIT MESSAGE POINTER
7966  041272  010137  007152                  MOV     R1,TXPTR
7967  041276  012702  000017                  MOV     #MSGLIM,R2
7968  041302  006302                          ASL     R2
7969  041304  006302                          ASL     R2
7970  041306  010137  007154                  MOV     R1,CMPPTR
7971  041312  060237  007154                  ADD     R2,CMPPTR       ;INIT COMPARE MESSAGE POINTER
7972  041316  013701  007154                  MOV     CMPPTR,R1
7973  041322  012702  000017                  MOV     #MSGLIM,R2
7974  041326  006302                          ASL     R2
7975  041330  006302                          ASL     R2
7976  041332  010137  007150                  MOV     R1,RXPTR
7977  041336  060237  007150                  ADD     R2,RXPTR        ;INIT RECEIVE MESSAGE POINTER
7978
7979  041342  013737  007156  007206          MOV     CMPTOT,RXMTOT   ;MAKE COMPARE AND RX MESSAGE COUNTS EQUAL
7980
7981
7982  041350  005037  007320          GTREX:  CLR     FLAG            ;CLEAR FLAG
7983  041354  005037  007212                  CLR     NOBUF           ;CLEAR NO BUFFER COUNTER
7984  041360  005037  007214                  CLR     PSCNT           ;CLEAR PASS COUNT
7985  041364  005037  007216                  CLR     ERRCNT          ;CLEAR ERROR COUNT
7986  041370  005037  007210                  CLR     LNCNT           ;CLEAR LINE COUNTER
7987
7988  041374  004737  023210                  JSR     PC,LOGDVI       ;LOG ABOUT TO INIT DEVICE
7989  041400  004737  044174                  JSR     PC,DVINIT       ;INIT DEVICE
7990
7991  041404  012737  001000  007230  GTRX2:  MOV     #BUFLIM,CURCC   ;SET CHAR COUNT TO 'BUFLIM'' NO. OF BYTES
7992  041412  012737  004400  007236          MOV     #RXBUF,CURADD   ;SET UP RX BUFFER AS CURRRENT ADD.
7993  041420  013737  007150  007234          MOV     RXPTR,CPTR
7994  041426  012737  000010  007226          MOV     #10,MSGTYP      ;SET UP FOR 33 TO FILL RX BUFFERS
7995  041434  004737  026320                  JSR     PC,BLDBUF       ;CLEAR RX BUFFER
7996  041440  013702  007306                  MOV     MODTYP,R2
7997  041444  006302                          ASL     R2
7998  041446  000172  007322                  JMP     @MODE(R2)       ;MODE DISPATCH
7999
```

```
                                        RECEIVE MODE SECTION

8001                                .SBTTL          RECEIVE MODE SECTION
8002                                ;++
8003                                ; FUNCTIONAL DESCRIPTION:
8004                                ;       RECEIVE-ONLY (OR ONE-WAY-IN) ROUTINE
8005                                ;       IN THIS MODE OF TESTING THE DEVICE'S RECEIVER IS ENABLED IN EXPECTATION
8006                                ;       OF RECEIVING A MESSAGE.  AFTER RECEIVING AN "EXPECTED" NUMBER OF
8007                                ;       MESSAGES, THE DATA RECEIVED CAN BE COMPARED AGAINST A LIST OF "EXPECT
8008                                ;       TO RECEIVE" MESSAGES IF DATA-CHECKING IS ENABLED.
8009                                ;
8010                                ; SUBORDINATE ROUTINES USED:
8011                                ;               "ALLTR"
8012                                ;
8013                                ; CALLING SEQUENCE:
8014                                ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
8015                                ;--
8016
8017  041452                        RXONLY:
8018  041452  013737  007150  007232 RXON2:  MOV     RXPTR,CPTRR
8019  041460  013737  007206  007204         MOV     RXMTOT,DVRCT    ;SET UP MESSAGE COUNT
8020  041466  052737  000104  007320         BIS     #QRX+#ERX,FLAG  ;SET UP RX QUE
8021  041474  005037  007234               CLR     CPTR            ;CLEAR THE TX POINTER
8022  041500  000137  041642               JMP     ALLTR           ;GO RX.
8023
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 25-7
CZKMUA.P11    30-MAR-82 09:13                    TRANSMIT MODE SECTION

  8025                                  .SBTTL          TRANSMIT MODE SECTION
  8026
  8027                                  ;++
  8028                                  ; FUNCTIONAL DESCRIPTION:
  8029                                  ;       TRANSMIT-ONLY (OR ONE-WAY-OUT) ROUTINE
  8030                                  ;       IN THIS MODE OF TESTING A LIST OF MESSAGES IS TRANSMITTED WITHOUT
  8031                                  ;       EXPECTING ANY DATA TO BE RECEIVED.  A REPETITION COUNT CAN BE
  8032                                  ;       SPECIFIED TO REPETITIVELY TRANSMIT THE LIST.
  8033                                  ;
  8034                                  ; SUBORDINATE ROUTINES USED:
  8035                                  ;               "ALLTR"
  8036                                  ;
  8037                                  ; CALLING SEQUENCE:
  8038                                  ;       JMP     @MODE(R2)       ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
  8039                                  ;--
  8040
  8041  041504  042737  000002  007314  TXONLY: BIC     #DATCKB,PARAM   ;SET NOCHECK
  8042  041512  013737  007152  007234  TXON2:  MOV     TXPTR,CPTR
  8043  041520  013737  007172  007170          MOV     TXMTOT,DVTCT    ;COPY COUNTER FOR THIS PASS
  8044  041526  052737  000210  007320          BIS     #QTX+#ETX,FLAG  ;SET THE QUE TX FLAG
  8045  041534  005037  007232                  CLR     CPTRR           ;CLEAR RX POINTER
  8046  041540  000137  041642                  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
                                ;--

041544                          PLCK:
041544  013737  007172  007170 PLCK2:  MOV     TXMTOT,DVTCT    ;SET UP  THE TRANSMIT COUNT
041552  013737  007152  007234         MOV     TXPTR,CPTR      ;SET UP CPTR TO TRANSMIT POINTER
041560  013737  007150  007232 PLCK3:  MOV     RXPTR,CPTRR     ;SET UP CPTRR TO REC POINTER
041566  052737  000104  007320         BIS     #QRX+#ERX,FLAG  ;SET UP Q AND EXPECT RX
041574  000137  041642                 JMP     ALLTR           ;AND GO RX FIRST MSG.
```

```
                                .SBTTL          ACTIVE MODE SECTION

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

041600  013737  007172  007170  ALCK:   MOV     TXMTOT,DVTCT
041606  013737  007152  007234          MOV     TXPTR,CPTR      ;SET UP TX COUNTS
041614  013737  007206  007204          MOV     RXMTOT,DVRCT    ;SET UP COUNTS
041622  013737  007150  007232          MOV     RXPTR,CPTRR
041630  052737  000314  007320          BIS     #QRX+#QTX+#ETX+#ERX,FLAG
041636  000137  041642                  JMP     ALLTR
```

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

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

041642                  ALLTR:
041642  032737  000004  007320  ALCK5:  BIT     #QRX,FLAG
041650  001420                  BEQ     ALCK1           ;IF NOT RX GO TO TX'S
041652  013702  007232          MOV     CPTRR,R2
041656  011237  007250          MOV     (R2),TEMP2
041662  012237  007200          MOV     (R2)+,DVRXA
041666  011237  007252          MOV     (R2),TEMP3
041672  011237  007202          MOV     (R2),DVRCC
041676  010237  007232          MOV     R2,CPTRR
041702  004737  044530          JSR     PC,DVRXQ        ;GO QUE DEVICE
041706  004737  023144          JSR     PC,LOGRXQ       ;LOG REC QUED
041712  032737  000010  007320  ALCK1:  BIT     #QTX,FLAG
```

```
8159  041720  001416                          BEQ     ALCK2           ;IF NO TX'S GO TO 2
8160  041722  013702  007234                  MOV     CPTR,R2
8161  041726  011237  007250                  MOV     (R2),TEMP2
8162  041732  012237  007164                  MOV     (R2)+,DVTXA
8163  041736  011237  007252                  MOV     (R2),TEMP3
8164  041742  012237  007166                  MOV     (R2)+,DVTCC
8165  041746  010237  007234                  MOV     R2,CPTR
8166  041752  004737  023110                  JSR     PC,LOGTXQ
8167
8168  041756  004737  044610          ALCK2:  JSR     PC,DVTXRX       ;GO TO TX AND RX SUB ROUT.
8169
8170  041762  032737  000004  007320          BIT     #QRX,FLAG       ;CHECK FOR REC. MSG.
8171  041770  001514                          BEQ     ALCK3
8172  041772  013737  007200  007250          MOV     DVRXA,TEMP2
8173  042000  013737  007202  007252          MOV     DVRCC,TEMP3
8174  042006  004737  023162                  JSR     PC,LOGRXC       ;LOG REC COMPLETE
8175  042012  032737  000004  007314  UPTABL: BIT     #ECHOB,PARAM    ;IS THIS ECHO MODE(PASSIVE)
8176  042020  001406                          BEQ     UPTA4           ;IF NOT GO TO 4
8177  042022  013702  007234                  MOV     CPTR,R2         ;ELSE SET R2 TO PRESENT TX TABL
8178  042026  013722  007250                  MOV     TEMP2,(R2)+     ;STORE OFF RX ADD
8179  042032  013712  007252                  MOV     TEMP3,(R2)      ;AND CC
8180  042036  032737  000002  007314  UPTA4:  BIT     #DATCKB,PARAM   ;DATA CHECK?
8181  042044  001015                          BNE     UPTA1           ;YES,BRANCH
8182  042046  012737  000001  007204          MOV     #01,DVRCT       ;ELSE SET DVRCT TO A 1
8183  042054  013737  007150  007232          MOV     RXPTR,CPTRR     ;RESET POINTER
8184  042062  022737  000003  007306          CMP     #ACT,MODTYP     ;IS THIS ACTIVE
8185  042070  001002                          BNE     UPTA3
8186  042072  005237  007204                  INC     DVRCT           ;IF YES BUMP COUNT
8187  042076  000424                  UPTA3:  BR      UPTEX
8188  042100  013702  007232          UPTA1:  MOV     CPTRR,R2
8189  042104  011237  007244                  MOV     (R2),TEMP       ;LOAD TEMP WITH PREV. COUNT
8190  042110  163737  007252  007244          SUB     TEMP3,TEMP      ;LOAD TEMP WITH PREV.COUNT-CURRENT
8191  042116  013722  007252                  MOV     TEMP3,(R2)+
8192  042122  063737  007252  007250          ADD     TEMP3,TEMP2
8193  042130  013722  007250                  MOV     TEMP2,(R2)+     ;STORE OF NEW ADD
8194  042134  013712  007244                  MOV     TEMP,(R2)       ;AND NEW CC
8195  042140  162702  000002                  SUB     #2,R2           ;PUT POINTER BACK TO ADDR.
8196  042144  010237  007232                  MOV     R2,CPTRR        ;AND RESTORE IT.
8197  042150                          UPTEX:
8198  042150  022737  000002  007306          CMP     #PAS,MODTYP
8199  042156  001007                          BNE     ALCK2A          ;IF NOT PASSIVE LOOP THEN GO TO 2A
8200  042160  042737  000104  007320          BIC     #QRX+#ERX,FLAG  ;CLEAR BOTH EXPECTED AND COMPLETED FLAGS
8201  042166  052737  000210  007320          BIS     #QTX+#ETX,FLAG  ;SET THE TX FLAGS
8202  042174  000646                          BR      ALCK1
8203
8204  042176  005337  007204          ALCK2A: DEC     DVRCT           ;DEC REC COUNT
8205  042202  005737  007204                  TST     DVRCT           ;IS IT ALL DONE
8206  042206  001005                          BNE     ALCK3           ;NO. GO CHECK TX
8207  042210  042737  000004  007320          BIC     #QRX,FLAG       ;CLEAR THE RX FLAG
8208  042216  005037  007232                  CLR     CPTRR           ;YES. CLEAR POINTER
8209  042222  032737  000010  007320  ALCK3:  BIT     #QTX,FLAG       ;IS IT TX
8210  042230  001447                          BEQ     ALCK4           ;IF NOT TX THEN GO BACK
8211  042232  013737  007164  007250          MOV     DVTXA,TEMP2
8212  042240  013737  007166  007252          MOV     DVTCC,TEMP3     ;LOG TX COMPLETED
8213  042246  004737  023126                  JSR     PC,LOGTXC
8214  042252  005337  007170                  DEC     DVTCT           ;DEC TX COUNT
```

```
 CZKMUA0 KMS11-BL P/P-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 25-12
CZKMUA.P11    30-MAR-82 09:13                    TRANSMIT - RECEIVE FOR ALL STANDARD MODES

 8215  042256  022737  000002  007306          CMP     #PAS,MODTYP
 8216  042264  001013                          BNE     ALCK3A          ;IF NOT PASSIVE MODE GO TO 3A
 8217  042266  042737  000210  007320          BIC     #QTX+ETX,FLAG   ;CLEAR THE TX FLAGS
 8218  042274  052737  000104  007320          BIS     #QRX+ERX,FLAG   ;AND SET THE RX FLAGS
 8219  042302  005737  007170                  TST     DVTCT
 8220  042306  001005                          BNE     ALCK3C          ;IF MORE RX'S DO IT
 8221  042310  000137  042370                  JMP     CMPSR           ; ELSE COMPARE
 8222  042314  005737  007170          ALCK3A: TST     DVTCT           ;IS IT ALL DONE
 8223  042320  001402                          BEQ     ALCK3B          ;IF NOT GO BACK TO 5
 8224  042322  000137  041642          ALCK3C: JMP     ALCK5
 8225  042326  005037  007234          ALCK3B: CLR     CPTR            ;IF SO CLEAR POINTER
 8226  042332  042737  000010  007320          BIC     #QTX,FLAG       ;CLEAR TX FLAG
 8227  042340  032737  000002  007314          BIT     #DATCKB,PARAM   ;IS IT DAT CK
 8228  042346  001403                          BEQ     ALCK4A          ;IF NOT THEN END WO CKING RX.
 8229  042350  005737  007232          ALCK4:  TST     CPTRR
 8230
 8231  042354  001362                          BNE     ALCK3C          ;IF SOME RX'S LEFT GO BACK
 8232  042356  005737  007234          ALCK4A: TST     CPTR
 8233  042362  001402                          BEQ     ALCK4B          ;BRANCH IF ANY TX'S LEFT
 8234  042364  000137  041756                  JMP     ALCK2
 8235  042370                          ALCK4B:
 8236
 8237
 8238
```

```
                        .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
                        ;--

042370  032737  000002  007314  CMPSR:  BIT     #DATCKB,PARAM   ;IS DATA CHECKING TO BE DONE
042376  001522                        BEQ     CMPSEX          ;IF NOT THEN EXIT
042400  013737  007150  007234        MOV     RXPTR,CPTR      ;PUT START OF RX POINTERS TO CPTR
042406  013737  007154  007232        MOV     CMPPTR,CPTRR    ; AND START OF COMPARE POINTS TO CPTRR
042414  013737  007206  007204        MOV     RXMTOT,DVRCT

042422                          CMPS3:
042422  013702  007234                MOV     CPTR,R2         ;MOVE CURRET RX PT.TO R2
042426  011237  007250                MOV     (R2),TEMP2      ;MOVE RX ADD TO EVENT LOG
042432  012201                        MOV     (R2)+,R1        ;SET R1 TO START ADD OF RX
042434  012237  007252                MOV     (R2)+,TEMP3     ;SET CHAR  COUNT TO EVENT LOG
042440  010237  007234                MOV     R2,CPTR         ;RESTORE RX POINT

042444  013702  007232                MOV     CPTRR,R2        ;PUT R2 AT COMPARE TABLE
042450  012203                        MOV     (R2)+,R3        ;SET R3 TO COMPARE ADD
042452  012204                        MOV     (R2)+,R4        ;SET R4 TO COMP CC
042454  010237  007232                MOV     R2,CPTRR        ;RESTORE POINTER
042460  010437  007254                MOV     R4,TEMP4
042464  004737  023256                JSR     PC,LOGCMP       ;LOG COMPARE START.

042470  020437  007252                CMP     R4,TEMP3        ;IS COMPARE COUNT = TO RX COUNT
042474  001410                        BEQ     CMPS7           ;IF SO GO TO 7
042476  005237  007216                INC     ERRCNT
042502                                ERRSOFT 1,EDDLE,ERR10   ;PRINT ERROR
```

```
  (4)  042502  104457                                                          TRAP    C$ERSOFT
  (5)  042504  000001                                                          .WORD   1
  (5)  042506  015660                                                          .WORD   EDDLE
  (5)  042510  022456                                                          .WORD   ERR10
8296   042512  004737  023274          JSR     PC,LOGCML       ;LOG LENGTH ERROR
8297
8298   042516  005037  007254  CMPS7:  CLR     TEMP4           ;CLEAR BAD BYTE COUNTER
8299   042522  012737  000001  007242  MOV     #1,OFSET        ;SET OFSET BYTE COUNT TO 1
8300   042530  122123          CMPS1:  CMPB    (R1)+,(R3)+     ;COMPARE RX WITH EXPETED
8301   042532  001422                  BEQ     CMPS6           ;IF EQUAL THEN GO TO 6
8302
8303   042534  005237  007254  CMPS2:  INC     TEMP4           ;INC BAD COUNT
8304   042540  023727  007254  000005  CMP     TEMP4,#5        ;IS IT MORE THEN 5
8305   042546  101014                  BHI     CMPS6           ;IF SO GO  FOR MORE
8306   042550  114337  007264          MOVB    -(R3),GOOD      ;STORE GOOD BYTE FOR ERROR
8307   042554  114137  007265          MOVB    -(R1),BAD       ;STORE BAD BYTE FOR ERROR
8308   042560  005237  007216          INC     ERRCNT
8309   042564                          ERRSOFT 2,EDDDE,ERR1    ;REPORT COMPARISON FAILURE TO OPR.
  (4)  042564  104457                                                          TRAP    C$ERSOFT
  (5)  042566  000002                                                          .WORD   2
  (5)  042570  015715                                                          .WORD   EDDDE
  (5)  042572  022366                                                          .WORD   ERR1
8310   042574  005201                  INC     R1
8311   042576  005203                  INC     R3
8312   042600  005237  007242  CMPS6:  INC     OFSET           ;INC OFFSET
8313   042604  005304                  DEC     R4              ;ELSE DEC CHAR COUNT AND SEE IF 0
8314   042606  001350                  BNE     CMPS1           ;IF NOT GO BACK
8315   042610  005737  007254          TST     TEMP4           ;SEE IF ANY CMP ERRS FOR THIS MSG
8316   042614  001410                  BEQ     CMPS5A          ;BR IF NONE
8317   042616  005237  007216          INC     ERRCNT
8318   042622                          ERRSOFT 3,EDDDE,ERR2    ;REPORT # OF MISMATCHES FOR MESSAGE
  (4)  042622  104457                                                          TRAP    C$ERSOFT
  (5)  042624  000003                                                          .WORD   3
  (5)  042626  015715                                                          .WORD   EDDDE
  (5)  042630  022430                                                          .WORD   ERR2
8319   042632  004737  023312  CMPS5:  JSR     PC,LOGCMD       ;LOG DATA ERROR IN COMPARE
8320   042636                  CMPS5A:
8321   042636  005337  007204          DEC     DVRCT
8322   042642  001267                  BNE     CMPS3           ;IF NOT ALL DONE GO BACK
8323
```

```
 8325                                   .SBTTL          INTERNAL END OF PASS CODE
 8326
 8327
 8328                                   ;++
 8329                                   ; FUNCTIONAL DESCRIPTION:
 8330                                   ;       THIS CODE INCREMENTS THE PASS COUNT FOR THE
 8331                                   ;       EVENT LOG. LOGS THE END OF PASS EVENT
 8332                                   ;       IF "RPASS" IS A MINUS ONE RETURN TO MODE
 8333                                   ;       DISPATCHER. IF NOT -1 THEN DECREMENT RPASS
 8334                                   ;       AND IF "RPASS" IS THEN = TO 0 GO TO DCLT PROMT
 8335                                   ;       IN NOT = TO 0 THEN GO BACK TO MODE DISPATCHER
 8336                                   ;
 8337                                   ; SUBORDINATE ROUTINES USED:
 8338                                   ;
 8339                                   ;               "LOGEOP" - LOG END OF PASS TO EVENT LOG
 8340                                   ;------
 8341
 8342  042644  005237  007214          CMPSEX: INC     PSCNT           ;BUMP PASS COUNT
 8343
 8344  042650  013737  007212  007254          MOV     NOBUF,TEMP4
 8345  042656  013737  007214  007250          MOV     PSCNT,TEMP2
 8346  042664  013737  007216  007252          MOV     ERRCNT,TEMP3
 8347  042672  004737  023330                  JSR     PC,LOGEOP       ;LOG END OF PASS
 8348  042676  022737  177777  007316  5$:     CMP     #-1,RPASS       ;SEE IF RPASS=-1
 8349  042704  001403                          BEQ     1$              ;IF IT IS DON'T DECRMNT, LOOP FOREVER
 8350  042706  005337  007316                  DEC     RPASS           ;DEC PASS COUNT
 8351  042712  001402                          BEQ     2$              ;IF DONE EXIT TEST
 8352  042714  000137  041404          1$:     JMP     GTRX2           ;ELSE GO BACK AND DISPATCH
 8353  042720  000137  036104          2$:     JMP     GTRA5           ;WHEN RPASS=0 GO BACK TO "DCLT>"
 8354
```

```
                                .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
                                ;--

                                DLL:    GMANID  DLLQ1,TEMP3,0,377,0,377,NO      ;GET PASSWORD
                                                                                        TRAP    C$GMAN
                                                                                        BR      10001$
                                                                                        .WORD   TEMP3
                                                                                        .WORD   T$CODE
                                                                                        .WORD   DLLQ1
                                                                                        .WORD   377
                                                                                        .WORD   T$LOLIM
                                                                                        .WORD   T$HILIM
                                                                                10001$:
042744  113737  007252  002650          MOVB    TEMP3,PASS1                     ;PUT PASSWORD IN MESSAGE
042752  113737  007252  002651          MOVB    TEMP3,PASS2                     ;PASSWORD IS DUPLICATE
042760  113737  007252  002652          MOVB    TEMP3,PASS3                     ;;HERE
042766  113737  007252  002653          MOVB    TEMP3,PASS4                     ;;;AND HERE.
042774  052737  000100  007320          BIS     #ERX,FLAG       ;SET EXPECTED TO RX
043002  042737  000002  007314          BIC     #DATCKB,PARAM   ;CLEAR NOCHECK
043010  012737  002647  007236          MOV     #DLLM1,CURADD   ;SET THE DOWN LINE LOAD MSG TO #1
043016  013737  002172  007230          MOV     DLLM1C,CURCC    ;SET THE CC
043024  004737  043116                  JSR     PC,DLTXRX       ;GO TO THE DOWN LINE TX RX ROUTINE

                                        ;RETURN WHEN TX AND RX ARE COMPLETED

043030  012737  002654  007236          MOV     #DLLM2,CURADD   ;SET THE DOWN LINE LOAD MSG TO #2
043036  013737  002174  007230          MOV     DLLM2C,CURCC    ;SET CC
043044  042737  000400  007320          BIC     #DLLGA,FLAG     ;CLEAR THE GO AHEAD FLAG
043052  004737  043116                  JSR     PC,DLTXRX       ;GO TO THE DOWN LINE TX RX ROUTINE

                                        ; RETURN WHEN TX AND RX ARE COMPLETED
043056                          DLLPRI:
```

```
 8403  043056                                 PRINTF  #DLLCM
  (7)  043056  012746  014642                                                  MOV     #DLLCM,-(SP)
  (6)  043062  012746  000001                                                  MOV     #1,-(SP)
  (3)  043066  010600                                                          MOV     SP,R0
  (4)  043070  104417                                                          TRAP    C$PNTF
  (4)  043072  062706  000004                                                  ADD     #4,SP
 8404  043076  000137  036104                 JMP     GTRA5
 8405
 840_  043102                          DLLEA:
 8407  043102                                 ERRHRD  20,DLLAB,ERR14
  (4)  043102  104456                                                          TRAP    C$ERHRD
  (5)  043104  000024                                                          .WORD   20
  (5)  043106  021752                                                          .WORD   DLLAB
  (5)  043110  022700                                                          .WORD   ERR14
 8408
 8409  043112  000137  036104                 JMP     GTRA5           ;PRINT ABORT AND EXIT
 8410
 8411
 8412
 8413  043116                          DLTXRX:
 8414  043116  052737  000004  007320         BIS     #QRX,FLAG       ;SET THE QUE RX FLAG
 8415  043124  012737  004400  007200         MOV     #RXBUF,DVRXA    ;SET THE DEVICE RX BUFFER TO RXBUF
 8416  043132  012737  004400  007250         MOV     #RXBUF,TEMP2    ;SET UP FOR LOG
 8417  043140  012737  000400  007202         MOV     #256.,DVRCC     ;SET UP FOR CC OF 256
 8418  043146  012737  000400  007252         MOV     #256.,TEMP3     ;SET UP FOR LOG
 8419  043154  004737  044530                 JSR     PC,DVRXQ        ; GO QUE RX
 8420  043160  004737  023144                 JSR     PC,LOGRXQ       ;AND LOG IT...
 8421
 8422  043164  013737  007236  007164         MOV     CURADD,DVTXA    ;SET UP FOR TX
 8423  043172  013737  007236  007250         MOV     CURADD,TEMP2    ;AND LOG
 8424  043200  013737  007230  007166         MOV     CURCC,DVTCC     ;SE UP FOR TX COUNT
 8425  043206  013737  007230  007252         MOV     CURCC,TEMP3     ;AND LOG IT
 8426  043214  004737  023110                 JSR     PC,LOGTXQ       ;LOG THE TX QUEUED
 8427  043220  052737  000210  007320         BIS     #QTX+#ETX,FLAG  ;SET UP TO QUE AND EXPECTED
 8428  043226  004737  044610         DLLE2:  JSR     PC,DVTXRX       ;GO TO DEVICE ROUTINE
 8429  043232  032737  000400  007320         BIT     #DLLGA,FLAG     ;TEST FOR GO AHEAD BIT
 8430  043240  001047                         BNE     DLLE1           ;IF SET GO TO ONE
 8431  043242  032737  000010  007320         BIT     #QTX,FLAG       ;ELSE CHECK FOR TX DONE
 8432  043250  001020                         BNE     DLLE6           ;IF DONE THEN BRANCH
 8433                                                                  ;ELSE ERROR
 8434  043252  012737  022257  007260         MOV     #TXNC,CONOTM
 8435  043260  013737  004400  007252  DLLE7:  MOV     RXBUF,TEMP3
 8436  043266  013737  003400  007254         MOV     TXBUF,TEMP4
 8437  043274  012737  021752  007250         MOV     #DLLAB,TEMP2
 8438  043302  004737  023172                 JSR     PC,LGDVE        ;LOG ERROR
 8439  043306  000137  043102                 JMP     DLLEA           ;ABORT TEST
 8440
 8441  043312  013737  007164  007250  DLLE6:  MOV     DVTXA,TEMP2
 8442  043320  013737  007166  007252         MOV     DVTCC,TEMP3
 8443  043326  004737  023126                 JSR     PC,LOGTXC       ;LOG TX DONE
 8444  043332  042737  000210  007320         BIC     #QTX+#ETX,FLAG  ;CLEAR QUE AND EXPECTED
 8445  043340  052737  000400  007320         BIS     #DLLGA,FLAG     ;SET THE GO AHEAD BIT
 8446  043346  023737  002174  007166         CMP     DLLM2C,DVTCC
 8447  043354  001472                         BEQ     DLLE5           ;EXIT IF SECOND MSG.
 8448  043356  000723                         BR      DLLE2           ;AND GO BACK TO 2
 8449  043360  032737  000004  007320  DLLE1:  BIT     #QRX,FLAG       ;IS THE A RX COMPLETED
```

```
 8450  043366  001004                        BNE     DLLE8           ;IF SO GO TO 8
 8451  043370  012737  022277  007260        MOV     #RXNC,CONOTM    ;ELSE SET UP ERROR AND ABORT.
 8452  043376  000730                        BR      DLLE7
 8453  043400  013737  007200  007250  DLLE8: MOV     DVRXA,TEMP2
 8454  043406  013737  007202  007252        MOV     DVRCC,TEMP3
 8455  043414  004737  023162                JSR     PC,LOGRXC       ;LOG RECEIVE COMPLETE
 8456  043420  122737  000010  004400        CMPB    #10,RXBUF       ;CHECK FOR FIRST WORD OF RX
 8457                                                                ;SEC BOOT MSG.
 8458  043426  001404                        BEQ     DLLE3
 8459  043430  012737  022317  007260  DLLE4: MOV     #RXM1,CONOTM    ;SET UP MESG AND ABORT
 8460  043436  000710                        BR      DLLE7           ;ABORT TEST
 8461
 8462  043440  122737  000001  004402  DLLE3: CMPB    #1,RXBUF+2      ;IS SECOND WORD 1?
 8463  043446  001404                        BEQ     DLLE5A          ;YES,BRANCH
 8464  043450  012737  022342  007260        MOV     #RXM2,CONOTM
 8465  043456  000700                        BR      DLLE7           ;SET UP MESSAGE AND ABORT
 8466
 8467  043460  012737  020273  007244  DLLE5A: MOV    #UNKM,TEMP      ;SET UP FOR UNKNOWN DEVICE
 8468  043466  113703  004401                MOVB    RXBUF+1,R3      ;GET DEVTYPE FROM MESSAGE
 8469  043472  120327  000042                CMPB    R3,#34.         ;OUT OF LEGAL RANGE ?
 8470  043476  101006                        BHI     DLLE5B          ;YES,BRANCH
 8471  043500  132703  000001                BITB    #1,R3           ;ODD ?
 8472  043504  001003                        BNE     DLLE5B          ;YES,BRANCH
 8473  043506  016337  010362  007244        MOV     DLLIND(R3),TEMP ;GET ASCIZ MESSAGE FROM TABLE
 8474
 8475  043514                        DLLE5B: PRINTF  #SECRM,TEMP,R3  ;PRINT ID MESSAGE
   (9) 043514  010346                                                        MOV     R3,-(SP)
   (8) 043516  013746  007244                                                MOV     TEMP,-(SP)
   (7) 043522  012746  020135                                                MOV     #SECRM,-(SP)
   (6) 043526  012746  000003                                                MOV     #3,-(SP)
   (3) 043532  010600                                                        MOV     SP,R0
   (4) 043534  104417                                                        TRAP    C$PNTF
   (4) 043536  062706  000010                                                ADD     #10,SP
 8476  043542  000207                DLLE5:  RTS     PC              ;RETURN TO CALLER
 8477
 8478
 8479
```

```
                                        .SBTTL          TALK MODE SECTION

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

                                        TALCK:
                                                BIC     #DATCKB,PARAM   ;SET NOCHECK
                                                MOV     #OPBUF,R2
                                        1$:     MOV     #-1,(R2)+       ;CLEAR OUT OPBUFFER FIRST
                                                CMP     #OPEND,R2
                                                BNE     1$
                                                GMANID  OPRMM,OPBUF,A,-1,1,72.,NO       ;GET TALK MESSAGE
                                                                                                TRAP    C$GMAN
                                                                                                BR      10002$
                                                                                                .WORD   OPBUF
                                                                                                .WORD   T$CODE
                                                                                                .WORD   OPRMM
                                                                                                .WORD   -1
                                                                                                .WORD   T$LOLIM
                                                                                                .WORD   T$HILIM
                                                                                        10002$:
                                                CLR     R2              ;NOW GET CHAR COUNT
                                        2$:     CMPB    #377,OPBUF(R2)
                                                BEQ     3$
                                                INC     R2
                                                BR      2$
                                        3$:     MOV     R2,OPCNT

                                                MOV     #OPBUF,DVTXA    ;SET UP TX ADDR.
                                                MOV     #OPBUF,TEMP2
                                                MOV     OPCNT,TEMP3
                                                MOV     OPCNT,DVTCC     ;SET UP TX CC
                                                JSR     PC,LOGTXQ
                                                BIS     #QTX+#ETX,FLAG  ;SET UP FLAGS
                                                CLR     CPTRR           ;CLEAR RX POINTER

                                                JSR     PC,DVTXRX

                                                MOV     DVTXA,TEMP2
                                                MOV     DVTCC,TEMP3
                                                JSR     PC,LOGTXC
                                                CMP     #'EX,OPBUF      ;CHECK FOR EXIT
```

```
 CZKMUA0 KM:11-BL PDP-11 DCLI   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 25-20
CZKMUA.P11    30-MAR-82 09:13                    TALK MODE SECTION

  8528  043732  001304                          BNE     TALCK
  8529  043734  022737  052111  002526          CMP     #''IT,OPBUF+2
  8530  043742  001300                          BNE     TALCK
  8531  043744  042737  000210  007320          BIC     #QTX+#ETX,FLAG  ;CLEAR THE TX BITS
  8532  043752  012737  000006  007306          MOV     #LIS,MODTYP     ;CHANGE TO LISTEN MODE
  8533  043760  000137  041404                  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
                                ;--

043764  042737  000002  007314  LISCK:  BIC     #DATCKB,PARAM   ;CLEAR CHECK BIT
043772                                  PRINTF  #LISP           ;PRINT PROMPT FOR OPR.
  (7)   043772  012746  014565                                                  MOV     #LISP,-(SP)
  (6)   043776  012746  000001                                                  MOV     #1,-(SP)
  (3)   044002  010600                                                          MOV     SP,R0
  (4)   044004  104417                                                          TRAP    C$PNTF
  (4)   044006  062706  000004                                                  ADD     #4,SP
044012  012737  002524  007200  LISCKA: MOV     #OPBUF,DVRXA    ;SET DEVICE UP TO REC AT OPBUF
044020  012737  002524  007250          MOV     #OPBUF,TEMP2
044026  012737  000122  007202          MOV     #82.,DVRCC      ;SET UP CHAR COUNT TO 82.
044034  012737  000122  007252          MOV     #82.,TEMP3
044042  052737  000104  007320          BIS     #QRX+#ERX,FLAG  ;SET UP FLAG
044050  005037  007234                  CLR     CPTR            ;CLEAR THE TX.

044054  004737  044530                  JSR     PC,DVRXQ        ;QUE RX
044060  004737  023144                  JSR     PC,LOGRXQ

044064  004737  044610                  JSR     PC,DVTXRX       ;GO TO DEVICE RX. SUBROUTINE

044070  013737  007200  007250          MOV     DVRXA,TEMP2
044076  013737  007202  007252          MOV     DVRCC,TEMP3     ;SET UP ADDR.AND CC.
044104  004737  023162                  JSR     PC,LOGRXC       ;LOG COMPLETED
044110  063737  007200  007202          ADD     DVRXA,DVRCC
044116  105077  143060                  CLRB    @DVRCC
044122                                  PRINTF  #OPBFPT
  (7)   044122  012746  002520                                                  MOV     #OPBFPT,-(SP)
  (6)   044126  012746  000001                                                  MOV     #1,-(SP)
  (3)   044132  010600                                                          MOV     SP,R0
  (4)   044134  104417                                                          TRAP    C$PNTF
  (4)   044136  062706  000004                                                  ADD     #4,SP
044142  022737  054105  002524          CMP     #"EX,OPBUF      ;COMPARE FOR EX OF "EXIT"
044150  001320                          BNE     LISCKA          ;IF NOT EXIT THEN GO BACK
044152  022737  052111  002526          CMP     #"IT,OPBUF+2    ;IF FIRST HALF OK CHECK NEXT PART
044160  001314                          BNE     LISCKA          ;IF NOT EXIT THE GO BACK
044162  012737  000005  007306          MOV     #TAL,MODTYP     ;CHANGE MODE TO TALK
044170  000137  041404                  JMP     GTRX2           ;RETURN TO DISPATCHER
```

```
 CZKMUAO KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 25-22
CZKMUA.P11    30-MAR-82 09:13                    LISTEN MODE SECTION

  8581
  8582
```

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

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

044174                          DVINIT:
                                        ;MASTER CLEAR DEVICE

044174  012737  000100  007360          MOV     #100,TIMER1     ;SET UP TIMER 1 FOR 100(OCTAL) TICKS
044202  005077  146010                  CLR     @SEL6
044206  005077  146000                  CLR     @SEL4
044212  005077  145764                  CLR     @SEL0           ;TURN OFF THE RUN BIT
044216  012777  040000  145756          MOV     #MCLR,@SEL0     ;DO A MASTER CLEAR
044224  112777  000200  145752          MOVB    #200,@BSEL1     ;SET RUN BIT
044232  005777  145744          DVIN2:  TST     @SEL0           ;IS RUN BIT SET
044236  100426                  BMI     DVIN1           ;IF YES GO TO 1 ELSE...
044240                          BREAK                                           TRAP    C$BRK
  (3)   044240  104422
044242  005737  007360          TST     TIMER1          ;SEE IF TIME HAS EXPIRED
044246  001371                  BNE     DVIN2           ;IF NOT GO BACK AND CHECK
                                                        ;AGAIN ELSE...PRINT ERROR
044250  012737  021162  007250          MOV     #DVEM3,TEMP2
044256  017737  145720  007252          MOV     @SEL0,TEMP3
044264  017737  145716  007254          MOV     @SEL2,TEMP4     ;LOAD UP ERRM. AND REG OUTPUTS
044272  004737  023172                  JSR     PC,LGDVE        ;LOG TIME OUT WAITING FOR RUN
044276  005237  007216                  INC     ERRCNT
044302                                  ERRSOFT 11,DVEM3,ERR13
                                                                                TRAP    C$ERSOFT
  (4)   044302  104457
  (5)   044304  000013                                                          .WORD   11
  (5)   044306  021162                                                          .WORD   DVEM3
  (5)   044310  022646                                                          .WORD   ERR13

044312  000730                  BR DVINIT               ;GO BACK AND TRY MSTR CLR AGAIN IF ERROR
```

```
8664
8665  044314                          DVIN1:
8666
8667                                          ; DO BASE IN COMMAND
8668
8669  044314  042737  000003  007320          BIC     #3,FLAG         ;CLEAR INPUT AND OUTPUT INT FLAGS
8670  044322  112777  000143  145652          MOVB    #143,@BSEL0     ;SET UP BASE IN INT EN
8671  044330  004737  045436                  JSR     PC,TOORIO       ;GO WAIT FOR INTERRUPT OR TIME OUT
8672  044334  012777  017370  145650          MOV     #BASE,@SEL4
8673  044342  012777  000000  145646          MOV     #0,@SEL6        ;SET UP SEL 6
8674  044350  052777  000100  145630  DVIN7:  BIS     #IEO,@SEL2      ;SET IEO
8675  044356  042777  004000  145616          BIC     #LULOOP,@SEL0   ;CLEAR LU LOOP
8676  044364  022737  000001  007310          CMP     #TTL,MLTYP      ;IS TTL SELECTED
8677  044372  001003                          BNE     DVIN3           ; IF NOT GO TO 3
8678  044374  052777  004000  145600          BIS     #LULOOP,@SEL0   ;ELSE SET LU LOOP
8679  044402  004737  045330          DVIN3:  JSR     PC,CLRAW
8680
8681                                          ; DO CONTROL IN COMMAND
8682
8683  044406  112777  000141  145566  DVIN8:  MOVB    #141,@BSEL0     ;SET UP CONTROL IN
8684  044414  004737  045436                  JSR     PC,TOORIO       ;WAIT FOR INT OR TIME OUT
8685  044420  005077  145572                  CLR     @SEL6           ;CLEAR HALF/DUP
8686  044424  022737  000004  007306          CMP     #DOW,MODTYP     ;IS THIS DOWN LINE LOAD?
8687  044432  001004                          BNE     DVIN5           ; BR IF NOT
8688  044434  052777  002400  145554          BIS     #MAINTB+HALFDB,@SEL6    ;IF SO SET MAINT MODE  BIT
8689  044442  000406                          BR      DVIN4           ; AND FORCE HALF DUPLEX
8690
8691  044444  005737  007312          DVIN5:  TST     FHDPLX          ;IS THIS A HALF/DUP
8692  044450  001003                          BNE     DVIN4           ;IF NOT GO TO 4
8693  044452  052777  002000  145536          BIS     #HALFDB,@SEL6   ;ELSE SET HALF/DUP
8694
8695  044460  017737  145532  007262  DVIN4:  MOV     @SEL6,CONTIN    ;SET UP CONTROL IN FOR MODS
8696  044466  004737  045330                  JSR     PC,CLRAW        ;GO CLEAR RQI AND WAIT
8697                                                                  ;FOR RDI TO GO AWAY.
8698  044472  000207                  DVINEX: RTS     PC              ;RETURN TO CALLER
8699
8700
8701
8702
8703
```

```
                                .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
                                ;--
044474  112777  000141  145500  DVMODS: MOVB    #141,@SEL0      ;SET UP CONTORL IN
044502  004737  045436          JSR     PC,TOORIO       ;GO TIME OUT CHECK
044506  017737  145500  010274  MOV     @SEL4,MODS      ;SET UP MODEM STATUS
044514  013777  007262  145474  MOV     CONTIN,@SEL6    ;SET UP OLD CONTORL IN
044522  004737  045330          JSR     PC,CLRAW
044526  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
                                ;--

044530                          DVRXQ:
044530  032737  000004  007320          BIT     #QRX,FLAG
044536  001423                        BEQ     DVREX           ;IF NOT RX THEN EXIT
                                                                ;ELSE QUE RX
044540  042737  000004  007320        BIC     #QRX,FLAG       ;CLEAR FLAG FOR RX
044546  112777  000144  145426        MOVB    #144,@BSEL0
044554  004737  045436                JSR     PC,TOORIO       ;GO CHECK FOR IN OR OUT
044560  017737  145426  010274        MOV     @SEL4,MODS      ;SET UP NEW MOD STATUS
044566  013777  007200  145416        MOV     DVRXA,@SEL4
044574  013777  007202  145414        MOV     DVRCC,@SEL6     ;LOAD CC AND ADDR
044602  004737  045330                JSR     PC,CLRAW        ;CLEAR AND WAIT
044606  000207                DVREX:  RTS     PC              ;RETURN TO CALLER
```

```
                        .SBTTL                  DEVICE TRANSMIT AND RECEIVE SUBROUTINE
                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       DVTXRX-DEVICE TRANSMIT AND RECEIVE ROUTINE
                        ;       THIS CODE QUES THE TRANSMIT BUFFER TO THE DEVICE
                        ;       IF NEEDED. THE CODE THEN WAITS FOR A TX COMPLE,
                        ;       RX COMPLETE OR BOTH. THE CODE REPORTS A TIME OUT
                        ;       ERROR IF NO BACC OUTPUT INTERRUPT IS RECIEVED BEFORE
                        ;       60 SECONDS. AFTER REPORTING ERROR TIMER IS RE STARTED
                        ;       AND DEVICE WILL CONTINUE TO WAIT FOR INTERRUPT. CODE
                        ;       ALSO REPORTS ERROR IF INPUT INTERRUPT OCCURS WHEN
                        ;       EXPECTING OUTPUT INTERRUPT;WHEN RX BACC OCCURS WHEN
                        ;       EXPECTING TX,AND WHEN TX INT. OCCURS WHEN EXPECTING
                        ;       RECIEVE.
                        ;
                        ;INPUTS:
                        ;       'DVTXA' = ADDRESS OF TRANSMIT MSG.
                        ;       'DVTCC' = BYTE COUNT OF TRANSMIT MSG.
                        ;    'QTX' BIT  = SET IF TRANSMIT REQUESTED
                        ;    'ETX' BIT  = SET IF TRNASMIT EXPECTED
                        ;    'ERX' BIT  = SET IF RECIEVE EXPECTED
                        ;
                        ;OUTPUTS:
                        ;       'DVTXA' = ADDRESS OF TX MSG. COMPLETED
                        ;       'DVTCC' = BYTE COUNT OF TX MSG. COMPLETED
                        ;       'QTX'   = SET IF TX COMPLETED
                        ;       'DVRXA' = ADDRESS OF RX MSG. COMPLETED
                        ;       'DVRCC' = BYTE COUNT OF RX MSG. COMPLETED
                        ;       'QRX'   = SET IF RX COMPLETED
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;               'TOORIO' - TIME OUT OR OUTPUT INTERRUPT OR INTPUT INTERRUPT
                        ;               'CLRAW'  - CLEAR RQI AND WAIT FOR RDI TO CLEAR
                        ;               'LGDVE'  - LOG DEVICE ERROR TO EVENT LOG
                        ;               'OUTHDL' - OUTPUT INTERRUPT HANDLER CODE
                        ;
                        ; CALLING SEQUENCE:
                        ;       JSR     PC,DVTXRX
                        ;--
044610  032737  000010  007320  DVTXRX: BIT     #QTX,FLAG       ;ANY TX TO QUE
044616  001423                        BEQ     DVTR3           ;IF NOT GO WAIT FOR OUPUT
044620  042737  000010  007320        BIC     #QTX,FLAG       ;CLEAR FLAG
044626  112777  000140  145346        MOVB    #140,@BSEL0
044634  004737  045436                JSR     PC,TOORIO       ;GO CHECK FOR IN OR OUT
044640  017737  145346  010274        MOV     @SEL4,MODS      ;PUT IN NEW MOD STAT
044646  013777  007164  145336        MOV     DVTXA,@SEL4
044654  013777  007166  145334        MOV     DVTCC,@SEL6
044662  004737  045330                JSR     PC,CLRAW        ;CLEAR RQUI ANDWAIT
044666                          DVTR3:
044666  012737  000074  007364        MOV     #60.,TIMERS     ;SET TIMER FOR 60 SECS
044674  032737  000060  007320  TOINOT: BIT     #CRX+#CTX,FLAG  ;IS IT TX OR RX COMP ALREADY?
```

```
 8886  044702  001071                          BNE     DVTR4           ;IS SO EXIT
 8887
 8888  044704  005737  007364                  TST     TIMERS          ;IS TIMER EXPIRED
 8889  044710  001022                          BNE     TOIN1
 8890  044712  012737  021246  007250          MOV     #DVEM4,TEMP2
 8891  044720  017737  145256  007252          MOV     @SEL0,TEMP3
 8892  044726  017737  145254  007254          MOV     @SEL2,TEMP4
 8893  044734  004737  023172                  JSR     PC,LGDVE
 8894  044740  005237  007216                  INC     ERRCNT
 8895  044744                                  ERRSOFT 12,DVEM4,ERR13
   (4) 044744  104457                                                      TRAP    C$ERSOFT
   (5) 044746  000014                                                      .WORD   12
   (5) 044750  021246                                                      .WORD   DVEM4
   (5) 044752  022646                                                      .WORD   ERR13
 8896  044754  000744                          BR      DVTR3           ;RETURN TO CHECK TIMER
 8897
 8898
 8899  044756                          TOIN1:  BREAK
   (3) 044756  104422                                                      TRAP    C$BRK
 8900  044760  032737  000001  007320          BIT     #ININT,FLAG     ;IS IT INPUT INTERRUPT
 8901  044766  001425                          BEQ     TOIN2           ;IF SO LOG ERROR
 8902
 8903  044770  012737  021340  007250          MOV     #DVEM5,TEMP2
 8904  044776  017737  145200  007252          MOV     @SEL0,TEMP3
 8905  045004  017737  145176  007254          MOV     @SEL2,TEMP4
 8906  045012  004737  023172                  JSR     PC,LGDVE
 8907  045016  042737  000001  007320          BIC     #ININT,FLAG     ;CLEAR BIT
 8908  045024  005237  007216                  INC     ERRCNT
 8909  045030                                  ERRSOFT 13,DVEM5,ERR13
   (4) 045030  104457                                                      TRAP    C$ERSOFT
   (5) 045032  000015                                                      .WORD   13
   (5) 045034  021340                                                      .WORD   DVEM5
   (5) 045036  022646                                                      .WORD   ERR13
 8910  045040  000715                          BR      TOINOT
 8911
 8912  045042  032737  000002  007320  TOIN2:  BIT     #OTINT,FLAG
 8913  045050  001711                          BEQ     TOINOT          ;IF NOT OUTPUT GO BACK AND
 8914                                                                  ;CHECK TIMER AGAIN
 8915  045052  004737  045560                  JSR     PC,OUTHDL       ;ELSE HANDLE OUTPUT AND RETURN
 8916  045056  032737  000060  007320          BIT     #CTX+#CRX,FLAG  ;IS IT TX OR RX
 8917  045064  001703                          BEQ     TOINOT          ;IF NOT GO BACK AND TRY AGAIN
 8918  045066  032737  000020  007320  DVTR4:  BIT     #CTX,FLAG       ;IS IT TX
 8919  045074  001440                          BEQ     DVTR5           ;IF NOT TRY RX
 8920  045076  032737  000200  007320          BIT     #ETX,FLAG       ;IF SO SHOULD IT BE
 8921  045104  001020                          BNE     DVTR4A          ;IF IT SHOULD GO TO 4A
 8922  045106  012737  021663  007250          MOV     #DVEM9,TEMP2    ;ELSE LOG ERROR
 8923  045114  013737  046250  007252          MOV     TSEL4,TEMP3
 8924  045122  013737  046252  007254          MOV     TSEL6,TEMP4
 8925  045130  004737  023172                  JSR     PC,LGDVE        ;
 8926  045134                                  ERRSOFT 14,DVEM9,ERR13  ;REPORT ERROR
   (4) 045134  104457                                                      TRAP    C$ERSOFT
   (5) 045136  000016                                                      .WORD   14
   (5) 045140  021663                                                      .WORD   DVEM9
   (5) 045142  022646                                                      .WORD   ERR13
 8927
 8928  045144  000411                          BR      DVTR4B          ;THEN CLEAR COMPL.FLAG
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 25-29
CZKMUA.P11    30-MAR-82 09:13                    DEVICE TRANSMIT AND RECEIVE SUBROUTINE

  8929  045146  013737  046250  007164  DVTR4A: MOV     TSEL4,DVTXA
  8930  045154  013737  046252  007166          MOV     TSEL6,DVTCC
  8931  045162  052737  000010  007320          BIS     #QTX,FLAG       ;AND SET TX COMPL FLAG
  8932  045170  042737  000020  007320  DVTR4B: BIC     #CTX,FLAG       ;ELSE CLEAR FLAG
  8933  045176  032737  000040  007320  DVTR5:  BIT     #CRX,FLAG       ;IS IT RX TOO?
  8934  045204  001440                          BEQ     DVTREX          ;IF NOT THEN EXIT.
  8935  045206  032737  000100  007320          BIT     #ERX,FLAG       ;TEST IS THIS SUPPOSED TO BE RX
  8936  045214  001020                          BNE     DVTR5A          ;IF YES PROCESS AS SUCH
  8937  045216  012737  021574  007250          MOV     #DVEM8,TEMP2
  8938  045224  013737  046254  007252          MOV     RSEL4,TEMP3
  8939  045232  013737  046256  007254          MOV     RSEL6,TEMP4     ;ELSE
  8940  045240  004737  023172                  JSR     PC,LGDVE        ;LOG ERROR
  8941  045244                                  ERRSOFT 15,DVEM8,ERR13
    (4) 045244  104457                                                                  TRAP    C$ERSOFT
    (5) 045246  000017                                                                  .WORD   15
    (5) 045250  021574                                                                  .WORD   DVEM8
    (5) 045252  022646                                                                  .WORD   ERR13
  8942
  8943  045254  000411                          BR      DVTRX1          ;AND EXIT
  8944  045256  013737  046254  007200  DVTR5A: MOV     RSEL4,DVRXA
  8945  045264  013737  046256  007202          MOV     RSEL6,DVRCC
  8946  045272  052737  000004  007320          BIS     #QRX,FLAG
  8947  045300  042737  000040  007320  DVTRX1: BIC     #CRX,FLAG       ;CLEAR FLAG FOR RX DONE
  8948  045306  000207                  DVTREX: RTS     PC              ;AND EXIT
  8949
```

```
                                ; DEVICE DEPENDENT SUBROUTINES
                                
                                
                                .SBTTL                  DEVICE INTERRUPT SERVICE ROUTINES
                                
                                
045310                                  BGNSRV  DVINS
045310                                                                          DVINS::
045310  052737  000001  007320          BIS     #ININT,FLAG
045316                                  ENDSRV
045316                                                                          L10021:
045316  000002                                                                          RTI
                                
045320                                  BGNSRV  DVOUTS
045320                                                                          DVOUTS::
045320  052737  000002  007320          BIS     #OTINT,FLAG
045326                                  ENDSRV
045326                                                                          L10022:
045326  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
                                ;--
                                
                                
045330  011637  007300  CLRAW:  MOV     (SP),PCADD      ;SAVE PC OF CALLING ROUTINE
045334  042777  000040  144640          BIC     #RQI,@SEL0
045342  012737  000050  007360  CLRA3:  MOV     #50,TIMER1      ;SET UP TIMER FOR 50(OCTAL) TICKS
045350  005737  007360  CLRA1:  TST     TIMER1
045354  001406                  BEQ     CLRA2           ;IF TIMER EXPIRED ERROR
045356                          BREAK
045356  104422                                                          TRAP    C$BRK
045360  032777  000200  144614          BIT     #RDI,@SEL0      ;IS RDI CLEAR
045366  001370                  BNE     CLRA1           ;IF NOT GO CHECK TIMER
                                                        ; ELSE
045370  000207                  RTS     PC              ;RETURN TO CALLER
045372  012737  021010  007250  CLRA2:  MOV     #DVEM0,TEMP2
045400  017737  144 '76  007252          MOV     @SEL0,TEMP3
045406  017737  144     007254          MOV     @SEL2,TEMP4
045414  004737  0231,                  JSR     PC,LGDVE        ;LOG DEVEICE EVENT 0
045420  005237  007216                  INC     ERRCNT
```

```
 9028  045424                    ERRSOFT 16,DVEM0,ERR9   ;WHILE WAITING FOR RDI
  (4)  045424  104457                                                  TRAP    C$ERSOFT
  (5)  045426  000020                                                  .WORD   16
  (5)  045430  021010                                                  .WORD   DVEM0
  (5)  045432  022570                                                  .WORD   ERR9
 9029  045434  000742                    BR      CLRA3           ;RESET TIMER AND CONTINUE
```

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

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       TOORIO - TIME OUT OR INPUT INTERRUPT OR OUTPUT INTERRUPT
                                ;       THIS ROUTINE SETS UP A TIMER FOR 100 (OCTAL) TICKS
                                ;       THEN CHECKS FOR TIME OUT,OR INPUT INTERRUPT,OR OUTPUT
                                ;       INTERRUPT. IF TIME OUT OCCURS IT REPORTS ERROR AND
                                ;       RESTARTS TIMER. IF INPUT INTERRUPT OCCURS RETURN TO CALLER
                                ;       IF OUTPUT INTERRPUT OCCURS LOG IT AND CONTINUE WAITING FOR
                                ;       INPUT INTERRUPT.
                                ;
                                ; USE OF FLAGS:
                                ;               "OTINT" - SET BY OUTPUT INT ROUTINE
                                ;               "ININT" - SET BY INPUT INT. ROUTINE
                                ;                         CLEARED BY THIS ROUTINTE.
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;               "OUTHDL" - OUTPUT INTERRUPT HANDLER
                                ;
                                ; CALLING SEQUENCE:
                                ;       JSR     PC,TOORIO
                                ;--
045436  011637  007300          TOORIO: MOV     (SP),PCADD      ;SAVE ADDR. OF CALLING ROUTINE
045442  012737  000100  007360          MOV     #100,TIMER1     ;SET UP TIMER
045450  005737  007360          TOOR3:  TST     TIMER1          ;IS TIME EXPIRED
045454  001022                  BNE     TOOR1           ;IF NOT CONTINUE
                                                        ;IF YES ERROR
045456  012737  021076  007250          MOV     #DVEM1,TEMP2
045464  017737  144516  007254          MOV     @SEL2,TEMP4
045472  017737  144504  007252          MOV     @SEL0,TEMP3
045500  004737  023172                  JSR     PC,LGDVE
045504  005237  007216                  INC     ERRCNT
045510                                  ERRSOFT 17,DVEM1,ERR9
 (4)    045510  104457                                          TRAP    C$ERSOFT
 (5)    045512  000021                                          .WORD   17
 (5)    045514  021076                                          .WORD   DVEM1
 (5)    045516  022570                                          .WORD   ERR9
045520  000746                  BR      TOORIO

045522                  TOOR1:  BREAK                                   TRAP    C$BRK
 (3)    045522  104422
045524  032737  000002  007320          BIT     #OTINT,FLAG     ;IS THERE AN OUTPUT
                                                        ;PENDING
045532  001402                  BEQ     TOOR2           ;IF NOT GO TO 2
                                                        ;ELSE GO HANDL IT
045534  004737  045560                  JSR     PC,OUTHDL       ;
045540  032737  000001  007320  TOOR2:  BIT     #ININT,FLAG     ;IS THERE AN INPUT PENDING
045546  001740                  BEQ     TOOR3           ;IF NOT GO BACK TO TIMER CK.
045550  042737  000001  007320          BIC     #ININT,FLAG     ;ELSE CLEAR THE INPUT PEND FLAG
045556  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
                        ;               "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
                        ;
                        ;--
                        ;
045560 011637 007300            OUTHDL: MOV     (SP),PCADD      ;SAVE ADDR. OF CALLING ROUTINE
045564 042737 000002 007320             BIC     #OTINT,FLAG
045572 032777 000200 144406             BIT     #RDO,@SEL2      ;CLEAR PEND FLAG AND CHK FOR RDO
045600 001023                           BNE     OUTH1           ;IF RDO OK ...ELSE LOG ERR...
045602 012737 021432 007250             MOV     #DVEM6,TEMP2
045610 017737 144372 007252             MOV     @SEL2,TEMP3
045616 017737 144374 007254             MOV     @SEL6,TEMP4
045624 004737 023172                    JSR     PC,LGDVE        ;GO LOG ERROR
045630 005237 007216                    INC     ERRCNT
045634                                  ERRSOFT 18,DVEM6,ERR9
  (4) 045634 104457                                                     TRAP    C$ERSOFT
  (5) 045636 000022                                                     .WORD   18
  (5) 045640 021432                                                     .WORD   DVEM6
  (5) 045642 022570                                                     .WORD   ERR9

                                        ;EXIT TEST IF ERROR

045644                                  ESCAPE  TST
  (3) 045644 104410                                                     TRAP    C$ESCAPE
  (3) 045646 000414                                                     .WORD   L10020-.

045650 032777 000001 144330 OUTH1:      BIT     #BACC,@SEL2     ;IS THE OUTPUT BACC
045656 001002                           BNE     1$              ; BR IF NO
045660 000137 046154                    JMP     OUTH2           ;IF SO GO TO 2
                                                                ;ELSE LOG ERROR AND PRINT IT
045664 017737 144326 007254 1$:         MOV     @SEL6,TEMP4
```

```
 9132                                          ; IF NO BUFFER OUTPUT JUST COUNT THEM
 9133
 9134  045672  032737  000004  007254          BIT     #BIT2,TEMP4
 9135  045700  001404                          BEQ     OUTH6           ;IF NO BUFF INC COUNT AND EXIT
 9136                                                                  ;ELSE GO TO 6
 9137  045702  005237  007212                  INC     NOBUF
 9138  045706  000137  046240                  JMP     OUTHEX
 9139
 9140  045712                          OUTH6:
 9141  045712  012737  021507  007250  51$:    MOV     #DVEM7,TEMP2
 9142  045720  017737  144262  007252          MOV     @SEL2,TEMP3
 9143
 9144  045726  004737  023172                  JSR     PC,LGDVE
 9145  045732  012737  014447  007260          MOV     #LPO,CONOTM     ;LOAD 'NULL STRING'' TO INIT CONOTM
 9146  045740  032737  000001  007254          BIT     #BIT0,TEMP4     ;IS THIS DATA CHECK
 9147  045746  001403                          BEQ     1$
 9148  045750  012737  022162  007260          MOV     #DATCKM,CONOTM
 9149  045756  032737  000002  007254  1$:     BIT     #BIT1,TEMP4     ;IS THIS TIMEOUT
 9150  045764  001403                          BEQ     2$
 9151  045766  012737  022151  007260          MOV     #TIMOM,CONOTM
 9152  045774  032737  000010  007254  2$:     BIT     #BIT3,TEMP4     ;IS THIS DDCMP MAINT RECVD
 9153  046002  001403                          BEQ     4$
 9154  046004  012737  022131  007260          MOV     #DDCMRM,CONOTM
 9155  046012  032737  000020  007254  4$:     BIT     #BIT4,TEMP4     ;IS THIS LOST DATA
 9156  046020  001403                          BEQ     5$
 9157  046022  012737  022117  007260          MOV     #LOSDAM,CONOTM
 9158  046030  032737  000100  007254  5$:     BIT     #BIT6,TEMP4     ;IS THIS DISCONNECT
 9159  046036  001403                          BEQ     6$
 9160  046040  012737  022104  007260          MOV     #DISCOM,CONOTM
 9161  046046  032737  000200  007254  6$.     BIT     #BIT7,TEMP4     ;IS THIS DDCMP START RECVD
 9162  046054  001403                          BEQ     7$
 9163  046056  012737  022064  007260          MOV     #DDCSRM,CONOTM
 9164  046064  032737  000400  007254  7$:     BIT     #BIT8,TEMP4     ;IS THIS NON-EXSISTENT MEMORY
 9165  046072  001403                          BEQ     8$
 9166  046074  012737  022046  007260          MOV     #NXMM,CONOTM
 9167  046102  032737  001000  007254  8$:     BIT     #BIT9,TEMP4     ;IS THIS PROCEDURE ERROR
 9168  046110  001403                          BEQ     9$
 9169  046112  012737  022026  007260          MOV     #PROEM,CONOTM
 9170  046120                          9$:
 9171  046120  032737  010000  007254  11$:    BIT     #BIT12,TEMP4    ;IS THIS CD GLITCHED
 9172  046126  001403                          BEQ     12$             ;BR IF NO
 9173  046130  012737  022227  007260          MOV     #CDGLM,CONOTM   ;IF SO SET UP MESSAGE
 9174
 9175  046136  005237  007216          12$:    INC     ERRCNT
 9176  046142                                  ERRSOFT 19,DVEM7,ERR8
   (4) 046142  104457                                                                    TRAP    C$ERSOFT
   (5) 046144  000023                                                                    .WORD   19
   (5) 046146  021507                                                                    .WORD   DVEM7
   (5) 046150  022506                                                                    .WORD   ERR8
 9177  046152  000432                          BR      OUTHEX          ;CLEAR RDO AND RETURN TO CALLER
 9178
 9179  046154                          OUTH2:
 9180  046154  032777  000004  144024          BIT     #RXBIT,@SEL2    ;IS THIS RX BACC OUT
 9181  046162  001012                          BNE     OUTH3           ;IF NOT THEN IT MUST BE TX.
 9182  046164  052737  000020  007320          BIS     #CTX,FLAG
 9183  046172  017737  144014  046250          MOV     @SEL4,TSEL4
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 25-35
CZKMUA.P11    30-MAR-82 09:13                      OUTPUT INTERRUPT HANDLER

  9184  046200  017737  144012  046252           MOV     @SEL6,TSEL6
  9185  046206  000414                           BR      OUTHEX
  9186
  9187  046210  052737  000040  007320  OUTH3:   BIS     #CRX,FLAG       ;SET RX COMPL
  9188  046216  017737  143770  046254  OUTH4:   MOV     @SEL4,RSEL4     ;THEN MOVE TO TEMP
  9189  046224  017737  143766  046256           MOV     @SEL6,RSEL6     ;AND SEL6 TO TEMP
  9190  046232  042737  140000  046256           BIC     #BIT15!BIT14,RSEL6 ;CLEAR Q SYNC & SELECT BITS
  9191  046240  042777  000200  143740  OUTHEX:  BIC     #RDO,@SEL2      ;CLEAR RDO
  9192  046246  000207                           RTS     PC              ;RETURN TO CALLER
  9193  046250  000000                   TSEL4:  .WORD   0
  9194  046252  000000                   TSEL6:  .WORD   0
  9195  046254  000000                   RSEL4:  .WORD   0
  9196  046256  000000                   RSEL6:  .WORD   0
  9197
  9198  046260  000207                           RTS     PC
  9199
  9200
```

```
 9213
 9214                                   .EVEN
 9215
 9216
 9217
 9223
 9224  046262                           ENDTST
  (3)  046262                                                  L10020:
  (3)  046262  104401                                                 TRAP    C$ETST
 9225
 9226
 9227
```

```
 9229                                   .SBTTL  HARDWARE PARAMETER CODING SECTION
 9230
 9231
 9232                                   ;++
 9233                                   ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
 9234                                   ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
 9235                                   ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
 9236                                   ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
 9237                                   ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
 9238                                   ; WITH THE OPERATOR.
 9239                                   ;--
 9240
 9241  046264                                   BGNHRD
   (3)  046264  000020                                                          .WORD L10023-L$HARD/2
   (3)  046266                                                          L$HARD::
 9242
 9252
 9253                                   .SBTTL          DEVICE INDEPENDENT SECTION
 9254
 9255  046266                                   GPRML   DPLX,0,1,YES
   (4)  046266  000130                                                          .WORD   T$CODE
   (4)  046270  046326                                                          .WORD   DPLX
   (4)  046272  000001                                                          .WORD   1
 9256
 9257
 9268
 9269
 9270                                   .SBTTL          DEVICE DEPENDENT SECTION
 9271
 9272  046274                                   GPRMA   CSRADR,2,0,160000,177776,YES
   (4)  046274  001031                                                          .WORD   T$CODE
   (4)  046276  046357                                                          .WORD   CSRADR
   (4)  046300  160000                                                          .WORD   T$LOLIM
   (4)  046302  177776                                                          .WORD   T$HILIM
 9273  046304                                   GPRMA   VECTOR,4,0,300,776,YES
   (4)  046304  002031                                                          .WORD   T$CODE
   (4)  046306  046405                                                          .WORD   VECTOR
   (4)  046310  000300                                                          .WORD   T$LOLIM
   (4)  046312  000776                                                          .WORD   T$HILIM
 9274  046314                                   GPRMD   PRIOR,6,0,340,4,7,YES
   (4)  046314  003032                                                          .WORD   T$CODE
   (4)  046316  046440                                                          .WORD   PRIOR
   (4)  046320  000340                                                          .WORD   340
   (4)  046322  000004                                                          .WORD   T$LOLIM
   (4)  046324  000007                                                          .WORD   T$HILIM
 9275
 9276  046326                                   ENDHRD
   (2)                                                                          .EVEN
   (3)  046326                                                          L10023:
 9277
 9278
 9279                                   ;DEVICE INDEPENDENT QUESTIONS
 9280
 9281                                   .NLIST  BEX
 9282
 9283  046326  052506  046114  042040  DPLX:   .ASCIZ  /FULL DUPLEX OPERATION : /
```

```
 CZKMUAO KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 25-38
CZKMUA.P11    30-MAR-82 09:13                    DEVICE DEPENDENT SECTION

  9284
  9292
  9293                                    ;DEVICE DEPENDENT QUESTION
  9294
  9295  046357     104  053105  041511  CSRADR: .ASCIZ  /DEVICE CSR ADDRESS : /
  9296  046405     111  052116  051105  VECTOR: .ASCIZ  /INTERRUPT VECTOR ADDRESS: /
  9297  046440  047111  042524  051122  PRIOR:  .ASCIZ  /INTERRUPT PRIORITY : /
  9298
  9299                                    .LIST   BEX
  9300                                            .EVEN
  9301
  9302
  9309
```

```
                              ;.SBTTL SOFTWARE PARAMETER CODING SECTION

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

                              ;       BGNSFT



                              ;       ENDSFT



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

      046466                  $PATCH:
      046466  000030                  .BLKW   30


      046546                          LASTAD
 (2)                                                                  .EVEN
 (4)  046546  000000                                                  .WORD   0
 (4)  046550  000000                                                  .WORD   0
 (3)  046552                  L$LAST::
      046552                          ENDMOD

              000001          .END
```

```
ABO    = 000026          4454#  7315
ACT    = 000003          4413#  7490    7869    7945    8184
ACTATV  040570           7656   7869#
ACTBCR  040374           7674   7835#
ACTCHK  041004           7636   7911#
ACTCLB  037716           7735   7749#
ACTCLP  041116           7670   7939#
ACTCLR  037352           7634   7688#
ACTCOP  040214           7644   7803#
ACTCRC  041020           7665   7917#
ACTCSE  037506           7639   7711#
ACTCST  037634           7640   7737#
ACTDLL  040636           7660   7883#
ACTDME  040142           7676   7783    7786#
ACTDMQ  040134           7677   7785#
ACTDMS  040112           7675   7780#
ACTDMX  040150           7787#
ACTECH  040714           7664   7900#
ACTEQO  040336           7648   7824#
ACTEXT  037436           7680   7700#
ACTHLP  037372           7638   7694#
ACTLIS  040626           7659   7880#
ACTLLP  041126           7671   7941#
ACTLPX  041144           7936   7938    7940    7942    7945#
ACTLXX  041206           7909   7930    7933    7946    7950#
ACTMEX  040562           7817   7833    7850    7855    7860    7863    7865#
ACTME1  040516           7839   7841    7843    7845    7847    7854#
ACTMOP  041076           7668   7935#
ACTMOS  041026           7679   7920#
ACTMS0  040416           7649   7838#
ACTMS1  040424           7650   7840#
ACTMS2  040434           7651   7842#
ACTMS3  040444           7652   7844#
ACTMS4  040454           7653   7846#
ACTMS5  040464           7654   7848#
ACTMS6  040502           7655   7851#
ACTM2X  040664           7870   7878    7881    7884    7887    7891#
ACTNO   040704           7663   7897#
ACTNUF  037342           7673   7685#
ACTNUL  037350           7633   7686#
ACTNUM  040224           7645   7806#
ACTOPM  040316           7646   7819#
ACTPAS  040600           7657   7872#
ACTPRO  041034           7666   7923#
ACTPRT  037446           7678   7702#
ACTQFG  041040           7912   7915    7918    7921    7925#
ACTREC  040620           7658   7877#
ACTREX  024334           6090   6110#
ACTRHL  024270           6089   6102#
ACTRLG  024344           6091   6114#
ACTRLP  041136           7672   7943#
ACTRNF  024260           6095   6098#
ACTRNL  024266           6088   6099#
ACTRPS  041066           7667   7932#
ACTRUN  037462           7637   7706#
ACTSEX  040526           7681   7857#
```

```
 CZKMUAO KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 27-1
CZKMUA.P11    30-MAR-82 09:13          CROSS REFERENCE TABLE -- USER SYMBOLS

ACTSHO  037362            7635    7691#
ACTSHW  037756            7722    7741    7760#   7772
ACTSTE  040156            7641    7793#
ACTSTS  041012            7647    7914#
ACTSTT  040166            7642    7796#
ACTSTX  040174            7794    7797#
ACTSWE  024360            6092    6119#
ACTSWF  024440            6093    6130#
ACTSWO  024504            6094    6139#
ACTSZE  040204            7643    7800#
ACTTAL  040656            7662    7889#
ACTTLP  041106            7669    7937#
ACTTRA  040646            7661    7886#
ADDCC   026222            6380#   7572    7606
ADDC1   026316            6382    6392#
ADR    = 000020 G         4396#
ALCK    041600            4873    8093#
ALCK1   041712            8149    8158#   8202
ALCK2   041756            8159    8168#   8234
ALCK2A  042176            8199    8204#
ALCK3   042222            8171    8206    8209#
ALCK3A  042314            8216    8222#
ALCK3B  042326            8223    8225#
ALCK3C  042322            8220    8224#   8231
ALCK4   042350            8210    8229#
ALCK4A  042356            8228    8232#
ALCK4B  042370            8233    8235#
ALCK5   041642            8148#   8224
ALLTR   041642            8022    8046    8071    8098    8147#
ASSEMB= 000010            4149
ATVMOD= 000027            4507#   5038
BACC   = 000001           4589#   9127
BAD     007265            4838#   5659    8307*
BADCHR= 000051            4525#   5152
BASE    017370            5489#   6122    6124    6132    6134    6151    6153    8672
BASM1   016065            5441#   6340
BASM1A  016032            5437#
BASM2   016056            5440#   6346
BASM3   016047            5439#   6344
BDATA   007274            4846#   6124*   6125*   6134*   6153*   6154*   6181
BEND    007272            4841#   6122*   6123*   6132*   6133*   6140*   6141    6151*   6152*   6189
BIT0   = 000001 G         4396#   4433    4561    4589    9146
BIT00  = 000001 G         4396#
BIT01  = 000002 G         4396#
BIT02  = 000004 G         4396#
BIT03  = 000010 G         4396#
BIT04  = 000020 G         4396#
BIT05  = 000040 G         4396#
BIT06  = 000100 G         4396#
BIT07  = 000200 G         4396#
BIT08  = 000400 G         4396#
BIT09  = 001000 G         4396#
BIT1   = 000002 G         4396#   4434    9149
BIT10  = 002000 G         4396#   4581    4582
BIT11  = 004000 G         4396#   4580
BIT12  = 010000 G         4396#   9171
```

```
BIT13 = 020000 G        4396#
BIT14 = 040000 G        4396#   4564    4579    9190
BIT15 = 100000 G        4396#   4578    9190
BIT2  = 000004 G        4396#   4435    4559    4587    9134
BIT3  = 000010 G        4396#   4436    4560    9152
BIT4  = 000020 G        4396#   4437    9155
BIT5  = 000040 G        4396#   4438    4562    4586
BIT6  = 000100 G        4396#   4588    9158
BIT7  = 000200 G        4396#   4563    4584    4585    9161
BIT8  = 000400 G        4396#   4583    9164
BIT9  = 001000 G        4396#   4565    9167
BLDBEX  026430          6437    6441#
BLDBUF  026320          6419#   7481    7488    7573    7607    7756    7995
BLDB1   026330          6424#
BLDB2   026372          6432#   6439
BLDB3   026410          6435#   6440
BOE   = 000400 G        4396#
BSEL0   012202          5215#   8670*   8683*   8739*   8793*   8877*
BSEL1   012204          5216#   7050*   7051*   8649*
BSEL2   012206          5218#
BSEL3   012210          5219#   7054*   7055*
BSEL4   012212          5221#
BSEL5   012214          5222#   7058*   7059*
BSEL6   012216          5224#
BSEL7   012220          5225#   7062*   7063*
BTHEAD  025006          6179    6196#
BTUP  = 002000          4467#
BUFEX   015035          5400#   7556    7585
BUFLIM= 001000          4401#   4787    4788    4789    6381    6388    6508    7554    7583    7749    7991
BYTBIT  007224          4818#   6341    7781*   7785*
CABLE = 000002          4420#
CBLLOP= 000045          4521#   5181
CCURAD  007162          4798#   6520*   6528    6532*   7477*   7485    7605    7610*   7734*
CDGLM   022227          5595#   9173
CHECK = 000003          4487#   5086
CLEAR = 000001          4485#   5024    7547    7688
CLIACT  037162          7517    7625#
CLIALN= 000007          4478#   5150
CLIALP= 000006          4477#   5072    5082    5155
CLIBCR  012546          5318#   7835
CLIBDL  012441          5316#   7949
CLIBIF= 000003          4474#
CLIBR = 000002          4473#   5021    5023    5025    5028    5030    5039    5041    5043    5045    5047    5050    5052
                        5067    5069    5079    5087    5099    5101    5117    5119    5122    5129    5131    5133    5135
                        5137    5139    5141    5143    5148    5151    5158    5160    5165    5180    5182    5184    5186
                        5191    6069
CLIBRX  012421          5315#   6758
CLIDEC= 000011          4480#   5164    5190
CLIERM  012324          5312#   6042    7522
CLIERR= 000000          4471#   5031    5152    6079
CLIEXI= 000001          4472#   5013    5015    5017    5019    5058    5060    5196    6061    6063    6065    6067    6072
                        6074    6078    6080
CLINBG  012377          5314#   6796
CLINPS  012503          5317#   7905
CLINUF  012354          5313#   6047    7526
CLINUM= 000005          4476#
```

```
CLIOCT= 000010          4479#  5074   5076   6077
CLIPW   012644          5320#  7961
CLIRAC  024222          6037   6083#
CLIRT   024024          6036   6059#
CLISEO  012613          5319#  7810
CLISPA= 000004          4475#  5011   5035   5056   5065   5073   5149   6059
CLISTR= 000012          4481#  5014   5016   5018   5020   5022   5024   5027   5029   5036   5038   5040   5042
                        5044   5046   5049   5051   5057   5059   5066   5068   5084   5086   5098   5100   5116
                        5118   5121   5128   5130   5132   5134   5136   5138   5140   5142   5157   5159   5179
                        5181   5183   5185   6062   6064   6066   6068   6071   6073   6075
CLITRE  010502          5008#  7516
CLI$PM  012310          5310#  7514
CLI$RP  012316          5311#  6034
CLKBR   007342          4881#  7009*
CLKCSR  007340          4880#  5859*  5880*  6990   7000*  7001*  7002*  7010*  7309*  7446*
CLKEN   007350          4884#  5880   6994*  7003*  7010   7446
CLKHZ   007346          4883#  5862   5877   5996   7011   7018
CLKINT  022766 G        5857#  7096
CLKSET  022742          5812#  6993   6999
CLKVEC  007344          4882#  6988*  7008*  7096
CLRAW   045330          8679   8696   8743   8798   8882   9013#
CLRA1   045350          9016#  9020
CLRA2   045372          9017   9023#
CLRA3   045342          9015#  9029
CMDBUF  003060          4711#  6034   6035   7514   7515
CMPBUF  005400          4789#  6506*  6520   7477   7733
CMPPTR  007154          4795#  6514*  6515*  6528*  6529*  6530*  6534*  7471*  7472*  7484   7594*  7595*  7603*
                        7604   7608*  7715*  7716*  7717   7730*  7731*  7732   7970*  7971*  7972   8275
CMPSEX  042644          8273   8342#
CMPSR   042370          8221   8272#
CMPS1   042530          8300#  8314
CMPS2   042534          8303#
CMPS3   042422          8278#  8322
CMPS5   042632          8319#
CMPS5A  042636          8316   8320#
CMPS6   042600          8301   8305   8312#
CMPS7   042516          8293   8298#
CMPTOT  007156          4796#  6523*  6526   6535*  7489*  7589*  7596   7609*  7719   7723*  7959   7979
CMSG0 = 000020          4500#  5130
CMSG1 = 000021          4501#  5128
CMSG2 = 000022          4502#  5132
CMSG3 = 000023          4503#  5134
CMSG4 = 000024          4504#  5138
CMSG5 = 000025          4505#  5136
CMSG6 = 000026          4506#  5140
CONOTM  007260          4835#  5680   5694   8434*  8451*  8459*  8464*  9145*  9148*  9151*  9154*  9157*  9160*
                        9163*  9166*  9169*  9173*
CONTIN  007262          4836#  8695*  8742
CPTR    007234          4825#  6422   6426*  7480*  7484*  7570*  7574   7604*  7608   7732*  7745*  7993*  8021*
                        8042*  8068*  8094*  8160   8165*  8177   8225*  8232   8274*  8279   8283*  8562*
CPTRR   007232          4824#  8018*  8045*  8069*  8096*  8150   8155*  8183*  8188   8196*  8208*  8229   8275*
                        8285   8288*  8520*
CR      020000          5500#  5989
CRC   = 000040          4516#
CRCB  = 000020          4437#  7917
CRX   = 000040          4463#  8885   8916   8933   8947   9187
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
CSHEXP= 000006          4490#   5057
CSHTRN= 000007          4491#   5059
CSRADR  046357          9272    9295#
CTOTCC  007160          4797#   6522*   6530    6532    6533*   6538*   7464*   7582    7587    7611*   7724*
CTS   = 000004          4559#   4906
CTSFM   022243          5596#
CTX   = 000020          4462#   8885    8916    8918    8932    9182
CURADD  007236          4826#   6424    6429    6431    6441*   7479*   7485*   7571*   7576    7605*   7610    7750*   7992
                        8390*   8396*   8422    8423
CURCC   007230          4823#   6380    6387    6390*   6391    6425    6430    6441    7475*   7487*   7752*   7813*   7926*
                        7849*   7852*   7854*   7991*   8391*   8397*   8424    8425
C$AU  = 000052          4149#   7410
C$AUTO= 000061          4149#   7290
C$BRK = 000022          4149#   8652    8899    9018    9070
C$BSEG= 000004          4149#
C$BSUB= 000002          4149#
C$CEFG= 000045          4149#
C$CLCK= 000062          4149#   6991    6997
C$CLEA= 000012          4149#   7337
C$CLOS= 000035          4149#
C$CLP1= 000006          4149#
C$CVEC= 000036          4149#
C$DCLN= 000044          4149#   6973    7087
C$DODU= 000051          4149#
C$DRPT= 000024          4149#
C$DU  = 000053          4149#   7373
C$EDIT= 000003          4149#   4227
C$ERDF= 000055          4149#
C$ERHR= 000056          4149#   8407
C$ERRO= 000060          4149#
C$ERSF= 000054          4149#
C$ERSO= 000057          4149#   8295    8309    8318    8661    8895    8909    8926    8941    9028    9067    9121    9176
C$ESCA= 000010          4149#   9125
C$ESEG= 000005          4149#
C$ESUB= 000003          4149#
C$ETST= 000001          4149#   9224
C$EXIT= 000032          4149#   7114    7320    7506    7546
C$GETB= 000026          4149#
C$GETW= 000027          4149#
C$GMAN= 000043          4149#   6034    7011    7514    8383    8506
C$GPHR= 000042          4149#   7031
C$GPLO= 000030          4149#
C$GPRI= 000040          4149#
C$INIT= 000011          4149#   7270
C$INLP= 000020          4149#
C$MANI= 000050          4149#   7502
C$MEM = 000031          4149#
C$MSG = 000023          4149#   5660    5664    5668    5682    5687    5691    5695
C$OPEN= 000034          4149#
C$PNTB= 000014          4149#   5659    5663    5667    5680    5681    5685    5686    5690    5694
C$PNTF= 000017          4149#   5989    5992    6028    6042    6047    6103    6143    6179    6188    6340    6344    6346
                        6386    6758    6796    7014    7084    7085    7086    7455    7495    7522    7526    7556    7565
                        7585    7599    7695    7769    7810    7835    7905    7949    7961    8403    8475    8556    8574
C$PNTS= 000016          4149#   6208    6226    6231    6237    6244    6245    6263    6271    6278    6291    6306    6578
                        6616
C$PNTX= 000015          4149#
```

```
C$QIO = 000377           4149#
C$RDBU= 000007           4149#  7006
C$REFG= 000047           4149#  6976  6978  6980  6983
C$RESE= 000033           4149#
C$REVI= 000003           4149#  4227
C$RFLA= 000021           4149#
C$RPT = 000025           4149#  6906
C$SEFG= 000046           4149#
C$SPRI= 000041           4149#  7112  7310
C$SVEC= 000037           4149#  7096  7108  7109
C$TPRI= 000013           4149#
DAM     020233           4947   5520#
DATAHD  035366           7085   7265#
DATALD  035437           7086   7266#
DATCKB= 000002           4434#  6586  7911  7957  8041  8180  8227  6272  8389  8501  8555
DATCKM  022162           5592#  9148
DCD   = 000001           4561#  4908
DCK   = 000014           4449#  5954
DCLFLG  007302           4849#  6970  6972*  7545*
DDCMRM  022131           5590#  9154
DDCSRM  022064           5587#  9163
DDE   = 000022           4452#  5962
DER   = 000010           4447#  5940
DEV1    010472           4987#  6242*  6245  6248*  6250*  6567  7496*  7773*
DEV2    010474           4988#  6243*  6245  6249*  6251*  6570  7497*  7774*
DEV3    010476           4989#  6252*  6577  7498*  7775*
DEV4    010500           4990#  6253*  6582  6586  6590  6594  7499*  7776*
DFPTBL  002130 G         4272#
DIAGMC= 000000           4149
DISCOM  022104           5588#  9160
DLE   = 000020           4451#  5958
DLL     042724           4874   8383#
DLLAB   021752           5582#  8407  8437
DLLCM   014642           5397#  8403
DLLEA   043102           8406#  8439
DLLE1   043360           8430   8449#
DLLE2   043226           8428#  8448
DLLE3   043440           8458   8462#
DLLE4   043430           8459#
DLLE5   043542           8447   8476#
DLLE5A  043460           8463   8467#
DLLE5B  043514           8470   8472  8475#
DLLE6   043312           8432   8441#
DLLE7   043260           8435#  8452  8460  8465
DLLE8   043400           8450   8453#
DLLGA = 000400           4466#  8398  8429  8445
DLLIND  010362           4943#  8473
DLLMOD= 000033           4511#  5046
DLLM1   002647           4612   4670#  8390
DLLM1C  002172           4612#  8391
DLLM1E  002654           4612   4675#
DLLM2   002654           4613   4677#  8396
DLLM2C  002174           4613#  8397  8446
DLLM2E  003060           4613   4704#
DLLPRI  043056           8402#
DLLQ1   012734           5321#  8383
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 27-6
CZKMUA.P11    30-MAR-82 09:13          CROSS REFERENCE TABLE -- USER SYMBOLS

DLM      020225              4945    5518#
DLTXRX   043116              8392    8399    8413#
DLVM     020251              4957    5524#
DMCEND   003300              4748#   6121    6131    6150
DMCIND   003250              4736#   6119    6130    6148
DMCM     020242              4949    5522#
DMC002   020342              4738    5541#
DMC003   020363              4739    5542#
DMC004   020420              4740    5543#
DMC005   020461              4741    5544#
DMC006   020514              4742    5545#
DMC007   020551              4743    5546#
DMC010   020606              4744    5547#
DMC011   020641              4745    5548#
DMC012   020663              4746    5549#
DMC013   020705              4747    5550#
DMC377   020744              4748    5551#
DMFMT    025057              6188    6197#
DMPE   = 000053              4527#   5076
DMPM     020255              4952    5525#
DMPQ   = 000054              4528#   5078
DMPS   = 000052              4526#   5074    7539    7782
DMSGAD   002176              4617#   6433
DMSGCT   002150              4602#   6432
DMUNKN   020322              4736    4737    5540#
DMVM     020316              4960    5533#
DNM      020246              4950    5523#
DOW    = 000004              4414#   7883    8686
DPLX     046326              9255    9283#
DPM      020217              4943    5516#
DQM      020230              4946    5519#
DSR    = 000010              4560#   4907
DTEM     020261              4953    5526#
DUM      020222              4944    5517#
DUMEX    026220              6348    6354#
DUMPSR   026064              6338#   7541
DUM1     026155              6342    6346#
DUM2     026200              6345    6347#
DUM3     026114              6341#   6352
DUM4     026070              6339#   6351
DUPM     020236              4948    5521#
DVEM0    021010              5563#   9023    9028
DVEM1    021076              5565#   9062    9067
DVEM3    021162              5567#   8656    8661
DVEM4    021246              5569#   8890    8895
DVEM5    021340              5571#   8903    8909
DVEM6    021432              5573#   9116    9121
DVEM7    021507              5575#   9141    9176
DVEM8    021574              5578#   8937    8941
DVEM9    021663              5580#   8922    8926
DVI    = 000012              4448#   5945
DVINEX   044472              8698#
DVINIT   044174              7989    8641#   8663
DVINS    045310 G            7108    8975#
DVIN1    044314              8651    8665#
DVIN2    044232              8650#   8654
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 27-7
CZKMUA.P11    30-MAR-82 09:13          CROSS REFERENCE TABLE -- USER SYMBOLS

DVIN3   044402            8677    8679#
DVIN4   044460            8689    8692    8695#
DVIN5   044444            8687    8691#
DVIN7   044350            8674#
DVIN8   044406            8683#
DVM     020265            4954    5527#
DVMODS  044474            5970    8739#
DVOUTS  045320 G          7109    8989#
DVRCC   007202            4808#   8154*   8173    8417*   8454    8559*   8570    8572*   8573*   8797    8945*
DVRCT   007204            4809#   8019*   8095*   8182*   8186*   8204*   8205    8276*   8321*
DVREX   044606            8790    8799#
DVRXA   007200            4807#   8152*   8172    8415*   8453    8557*   8569    8572    8796    8944*
DVRXQ   044530            8156    8419    8564    8788#
DVTCC   007166            4801#   8164*   8212    8424*   8442    8446    8517*   8525    8881    8930*
DVTCT   007170            4802#   8043*   8067*   8093*   8214*   8219    8222
DVTREX  045306            8934    8948#
DVTRX1  045300            8943    8947#
DVTR3   044666            8875    8883#   8896
DVTR4   045066            8886    8918#
DVTR4A  045146            8921    8929#
DVTR4B  045170            8928    8932#
DVTR5   045176            8919    8933#
DVTR5A  045256            8936    8944#
DVTXA   007164            4800#   8162*   8211    8422*   8441    8514*   8524    8880    8929*
DVTXRX  044610            8168    8428    8522    8567    8874#
DZM     020270            4955    5528#
ECHO  = 000037            4515#   5100
ECHOB = 000004            4435#   6590    7891    7902    7907    8175
EDABO   016021            4939    5431#
EDDCK   015602            4934    5425#
EDDDE   015715            4937    5428#   8309    8318
EDDER   015565            4932    5424#
EDDLE   015660            4936    5427#   8295
EDDVI   015632            4933    5426#
EDEOP   015750            4938    5429#
EDMOS   015764            5430#
EDRXC   015537            4931    5423#
EDRXQ   015512            4930    5422#
EDTXC   015463            4929    5421#
EDTXQ   015437            4928    5420#
EFM11   015362            5415#   5667
EFM2    015265            5413#   5663
EF.CON= 000036 G          4396#   6980
EF.NEW= 000035 G          4396#   6983
EF.PWR= 000034 G          4396#
EF.RES= 000037 G          4396#   6978
EF.STA= 000040 G          4396#   6976
EMSG0   002221            4603    4630#
EMSG1   002222            4604    4632#
EMSG2   002223            4605    4634#
EMSG3   002224            4606    4636#
EMSG4   002324            4607    4642#
EMSG5   002416            4608    4647#
EMSG6   002517            4609    4651#
EMSG8   002647            4611    4665#
ENADD   007222            4817#   6347    7786*
```

```
ENDEVT  025742            6209    6219    6223    6282#
ENDIT   031314            6982    7111#
ENDQQ = 000017            4499#   5147
EOP   = 000024            4453#   5966
ERRCNT  007216            4815#   5969    5972    7318    7985*   8294*   8308*   8317*   8346    8660*   8894*   8908*   9027*
                          9066*   9120*   9175*
ERR1    022366 G          5658#   8309
ERR10   022456 G          5666#   8295
ERR13   022646 G          5689#   8661    8895    8909    8926    8941
ERR14   022700 G          5693#   8407
ERR2    022430 G          5662#   8318
ERR8    022506 G          5679#   9176
ERR9    022570 G          5684#   9028    9067    9121
ERX   = 000100            4464#   8020    8070    8097    8200    8218    8388    8561    8935
ETX   = 000200            4465#   8044    8097    8201    8217    8427    8444    8519    8531    8920
EVL   = 000004 G          4396#
EVMCTS  017063            4917    5465#
EVMDCD  017073            4919    5467#
EVMDSR  017067            4918    5466#
EVMOCG  016743            5458#
EVMOHD  016766            5463#   6291
EVMOST  017046            5464#   6306
EVMRI   017103            4921    5469#
EVMRTS  017077            4920    5468#
EVMSQD  017107            4922    5470#
EVMTM   017113            4923    5471#
EVTADD  010434            4967#   6234*   6237    6260*   6263    6268*   6271    6275*   6278
EVTBCT  010436            4968#   6235*   6237    6261*   6263    6269*   6271    6276*   6278
EVTEND  010272            4898#   6004    6216    7022
EVTF0   016133            5447#   6226
EVTF1   016226            5448#   6231
EVTF2   016255            5449#   6237
EVTF3   016327            5450#   6244
EVTF3C  016341            5451#   5685    5690    6245
EVTF3D  016356            5452#   5680    5694
EVTF4   016400            5453#   6271
EVTF4A  016502            5454#   6278
EVTF4B  016600            5455#   6263
EVTF5A  016657            5456#   5659
EVTLOG  007370            4896    4897#   6008    6206    6214    7019
EVTLST  010332            4928#   6231
EVTMIN  010430            4965#   6230*   6231
EVTPTR  007366            4896#   5994    6009*   6205    6221    7020*
EVTSEC  010426            4964#   6229*   6231
EVTTCK  010432            4966#   6228*   6231
EVTTMP  010440            4969#   6241*   6244    6262*   6263    6270*   6271    6277*   6278
EXIT  = 000057            4531#   5018    7311    7543    7700
EXITCL  035546            7312    7320#
EXITIT  031322            7114#
ESEND = 002100            4149#
ESLOAD= 000035            4149#   4227
FACSIM  026444            6503#   7861
FHDPLX  007312            4857#   7034*   8691
FIRMLD  035326            7084    7264#
FLAG    007320            4867#   7982*   8020*   8044*   8070*   8097*   8148    8158    8170    8200*   8201*   8207*   8209
                          8217*   8218*   8226*   8388*   8398*   8414*   8427*   8429    8431    8444*   8445*   8449    8519*
```

```
                    8531*  8561*  8669*  8789   8792*  8874   8876*  8885   8900   8907*  8912   8916   8918
                    8920   8931*  8932*  8933   8935   8946*  8947*  8976*  8990*  9071   9076   9078*  9113*
                    9182*  9187*
FS$AU  = 000015      4149#  7383   7410
FS$AUTO= 000020      4149#  7281   7290
FS$BGN = 000040      4149#  4170   5658   5662   5666   5679   5684   5689   5693   5697   585-   6874   6915
                     6944   7114   7281   7299   7320   7346   7357   7383   7394   7436   7506   7546   8975
                     8989   9125   9224   9241   9360
FS$CLEA= 000007      4149#  7299   7337
FS$DU  = 000016      4149#  7346   7373
FS$END = 000041      4149#  4170   5660   5664   5668   5682   5687   5691   5695   5697   5881   6906   7114
                     7270   7290   7320   7337   7357   7373   7394   7410   7436   7506   7546   8977   8991
                     9125   9224   9276   9360
FS$HARD= 000004      4149#  9241   9276
FS$HW  = 000013      4149#  4272   4318
FS$INIT= 000006      4149#  6944   7270
FS$JMP = 000050      4149#  5697   7114   7320   7357   7394   7506   7546
FS$MOD = 000000      4149#  4170   9360
FS$MSG = 000011      4149#  5658   5660   5662   5664   5666   5668   5679   5682   5684   5687   5689   5691
                     5693   5695
FS$PROT= 000021      4149#  6915   6921
FS$PWR = 000017      4149#
FS$RPT = 000012      4149#  6874   6906
FS$SEG = 000003      4149#
FS$SOFT= 000005      4149#
FS$SRV = 000010      4149#  5857   5881   8975   8977   8989   8991
FS$SUB = 000002      4149#
FS$SW  = 000014      4149#
FS$TEST= 000001      4149#  7436   9224
GETCL    036170      7503   7512#  7523   7527   7530   7533   7535   7542   7548   7550   7557   7566   7580
                     7586   7600   7614   7962
GETPRM   030620      6985   7027#  7032
GETRCL   023632      6030#  6043   6048   6052
GOOD     007264      4837#  5659   8306*
GTRA2    035576      7449#
GTRA3    035640      7452   7457#
GTRA4    035646      7462#
GTRA5    036104      7458   7496#  8353   8404   8409
GTREX    041350      7982#
GTRX2    041404      7991#  8352   8533   8580
GTR9     041214      7508   7538   7955#
GTXRXB   035576      7448#
GS$CNTO= 000200      4149#
GS$DELM= 000372      4149#
GS$DISP= 000003      4149#
GS$EXCP= 000400      4149#
GS$HILI= 000002      4149#
GS$LOLI= 000001      4149#
GS$NO  = 000000      4149#  6034   7514   8383   8506
GS$OFFS= 000400      4149#  6034   7011   7514   8383   8506   9255   9272   9273   9274
GS$OFSI= 000376      4149#  6034   7011   7514   8383   8506   9255   9272   9273   9274
GS$PRMA= 000001      4149#  9272   9273
GS$PRMD= 000002      4149#  6034   7011   7514   8383   8506   9274
GS$PRML= 000000      4149#  9255
GS$RADA= 000140      4149#  6034   7514   8506
GS$RADB= 000000      4149#
```

```
G$RADD= 000040            4149#   7011
G$RADL= 000120            4149#   9255
G$RADO= 000020            4149#   8383    9272    9273    9274
G$XFER= 000004            4149#
G$YES = 000010            4149#   7011    9255    9272    9273    9274
HALFDB= 002000            4582#   8688    8693
HELP  = 000000            2195#   4140    4166    4209    4236    4256    4274    4351    4365    4381    5233    5272    5293
                          5607    5619    5642    5708    6876    6891    6899    6923    6946    7252    7283    7301    7322
                          7348    7359    7385    7396    7423    7430    7438    9202    9218    9243    9303    9325    9338
                          9352
HELPDC= 000000            2200#   2209    4229    4292    4322    4541    4568    5199    5246    5278    5553    5670    7036
                          7100    8586    8605    8707    8749    8804    8953    8964    8979    9258    9285
HLP   = 000005            4489#   5012    5014    7532    7698
HLPEND  003230            4724#   7696
HLPF    013037            5323#   6103    7695
HLPTAB  003210            4716#   7694
HLP0    012761            5322#   7495
HLP1    013044            4716    5324#
HLP2    013057            4717    5325#
HLP3    013174            4718    5329#
HLP3A   013261            4719    5330#
HLP4    013306            4720    5331#
HLP4A   013365            4721    5332#
HLP5    013443            4722    5333#
HLP6    013533            4723    5334#
HOE   = 100000 G          4396#
IBE   = 010000 G          4396#
IDU   = 000040 G          4396#
IEO   = 000100            4588#   8674
IER   = 020000 G          4396#
INDEX   007266            4839#   6119*   6120*   6130*   6148*   6149*   6180
INDEXE  007270            4840#   6121*   6131*   6150*   6184    6186
ININT = 000001            4458#   3900    8907    8976    9076    9078
INIT1   030276            6971    6975#
INTPRI  012226            5230#   7068*   7108    7109
INVEC   012222            5228#   7065*   7108
ISR   = 000100 G          4396#
IXE   = 004000 G          4396#
I$AU  = 000041            4149#   7383#   7410#
I$AUTO= 000041            4149#   7281#   7290#
I$CLN = 000041            4149#   7299#   7320    7337#
I$DU  = 000041            4149#   7346#   7373#
I$HRD = 000041            9241#   9276#
I$INIT= 000041            4149#   6944#   7114    7270#
I$MOD = 000041            4149#   4170#   9360#
I$MSG = 000041            4149#   5658#   5660#   5662#   5664#   5666#   5668#   5679#   5682#   5684#   5687#   5689#   5691#
                          5693#   5695#
I$PROT= 000040            4149#   6915#
I$PTAB= 000041            4149#
I$PWR = 000041            4149#
I$RPT = 000041            4149#   6874#   6906#
I$SEG = 000041            4149#   7436
I$SETU= 000041            4149#
I$SRV = 000041            4149#   5857#   5881#   8975#   8977#   8989#   8991#
I$SUB = 000041            4149#   7436
I$TST = 000041            4149#   7436#   7506    7546    9125    9224#
```

```
J* JMP = 000167           4149#   5697    7357    7394
KDPM     020303           4957    5530#
KDZM     020307           4958    5531#
KEYWD1   003202           4712#   6050    6106*   6110*   6115*   6969*   7311    7529    7532    7534    7536    7539    7543
                          7547    7549    7551    7688*   7691*   7698*   7700*   7702*   7706*   7721    7740    7782*   7793*
                          7796*   7857    7862*
KLM      020313           4959    5532#
KRUN   = 100000           4578#
LCLKEN= 000100            4427#   6994
LDFIRM   031032           7072#
LGDVE    023172           5938#   8438    8659    8893    8906    8925    8940    9026    9065    9119    9144
LIS    = 000006           4416#   7880    8532
LISCK    043764           4876    8555#
LISCKA   044012           8557#   8576    8578
LISMOD= 000032            4510#   5044
LISP     014565           5386#   8556
LMDLOP= 000046            4522#   5183
LNCNT    007210           4812#   5985    5987*   5991*   7986*
LOE    = 040000 G         4396#
LOGCMD   023312           5960#   8319
LOGCML   023274           5956#   8296
LOGCMP   023256           5952#   8290
LOGDVI   023210           5943#   7988
LOGEOP   023330           5964#   8347
LOGEX    023602           5974    6011#
LOGRXC   023162           5935#   8174    8455    8571
LOGRXQ   023144           5930#   8157    8420    8565
LOGS1    023346           5923    5928    5933    5937    5969#
LOGS2    023574           6005    6009#
LOGS3    023400           5941    5950    5955    5959    5963    5967    5978#
LOGS4    023454           5986    5990#
LOGS5    023500           5980    5982    5993#   7319*
LOGTXC   023126           5925#   8213    8443    8526
LOGTXQ   023110           5920#   8166    8426    8518
LOGUNT   007276           4847#   7025*   7027*   7028    7031
LOOPS    003350           4767#   6576
LOSDAM   022117           5589#   9157
LOT    = 000010 G         4396#
LP0      014447           4767    4935    5361#   6575    9145
LP00     014450           5362#   6572
LP1      014457           4768    5363#
LP2      014470           4769    5364#
LP3      014476           4770    5365#
LP4      014511           4771    5366#
LULOOP= 004000            4580#   8675    8678
L$ACP    002110 G         4227#
L$APT    002036 G         4227#
L$AU     035562 G         4227    7383#
L$AUT    002070 G         4227#
L$AUTO   035466 G         4227    7281#
L$CCP    002106 G         4227#
L$CLEA   035470 G         4227    7299#
L$CO     002032 G         4227#
L$DEPO   002011 G         4227#
L$DESC   012252 G         4227    5290#
L$DESP   002076 G         4227#
```

```
LSDEVP  002060 G          4227#
LSDISP  002124 G          4227    4254#
LSDLY   002116 G          4227#
LSDTP   002040 G          4227#
LSDTYP  002034 G          4227#
LSDU    035554 G          4227    7346#
LSDUT   002072 G          4227#
LSDVTY  012230 G          4227    5270#
LSEF    002052 G          4227#
LSENVI  002044 G          4227#
LSETP   002102 G          4227#
LSEXP1  002046 G          4227#
LSEXP4  002064 G          4227#
LSEXP5  002066 G          4227#
LSHARD  046266 G          4227    9241#
LSHIME  002120 G          4227#
LSHPCP  002016 G          4227#
LSHPTP  002022 G          4227#
LSHW    002130 G          4227    4272#
LSICP   002104 G          4227#
LSINIT  030256 G          4227    6944#
LSLADP  002026 G          4227#
LSLAST  046552 G          4227    9359#
LSLOAD  002100 G          4227#
LSLUN   002074 G          4227#
LSMREV  002050 G          4227#
LSNAME  002000 G          4227#
LSPRIO  002042 G          4227#
LSPROT  030250 G          4227    6915#
LSPRT   002112 G          4227#
LSREPP  002062 G          4227#
LSREV   002010 G          4227#
LSRPT   030242 G          4227    6874#
LSSPC   002056 G          4227#
LSSPCP  002020 G          4227#
LSSPTP  002024 G          4227#
LSSTA   002030 G          4227#
LSTEST  002114 G          4227#
LSTIML  002014 G          4227#
LSUNIT  002012 G          4227#   7028
L10000  002150            4272    4318#
L10001  022426            5660#
L10002  022454            5664#
L10003  022504            5668#
L10004  022566            5682#
L10005  022644            5687#
L10006  022676            5691#
L10007  022734            5695#   5697
L10010  023106            5881#
L10011  030246            6906#
L10013  035464            7114    7270#
L10014  035466            7290#
L10015  035552            7320    7337#
L10016  035560            7357    7373#
L10017  035566            7394    7410#
L10020  046262            7506    7546    9125    9224#
```

CZKMUA0 KMS11-BL PDP-11 DCLT MACY11 30A(1052) 30-MAR-82 09:15 PAGE 27-13
CZKMUA.P11 30-MAR-82 09:13 CROSS REFERENCE TABLE -- USER SYMBOLS

```
L10021  045316            8977#
L10022  045326            8991#
L10023  046326            9241    9276#
L5060   014603            5388#   7011
MAINTB= 000400            4583#   8688
MCBEGL  031326            7073    7117#   7248
MCENDL  035326            7091    7247#
MCLR  = 040000            4579#   8648
MCSIZL= 004000 G          7248#
MLTYP   007310            4854#   5947    7491*   7497    7507*   7774    7874*   7893*   7935*   7937*   7939*   7941*   7943*
                          7948*   7955    8676
MOBITE  010314            4913#   6304
MOBITS  010276            4906#   6292
MOCHK = 000010            4436#   6594    7920
MODE    007322            4870#   7998
MODES   003332            4759#   6569
MODLOC= 000003            4421#
MODREM= 000004            4422#
MODS    010274            4902#   5976    8741*   8795*   8879*
MODTYP  007306            4851#   5946    7490*   7496    7773    7869*   7872*   7877*   7880*   7883*   7886*   7889*   7903
                          7945    7996    8184    8198    8215    8532*   8579*   8686
MOMSGS  010314            4917#   6293
MOP   = 000043            4423#   4519#
MOSC  = 000056            4530#   5121
MO0     014356            4759    5354#
MO1     014366            4760    5355#
MO2     014377            4761    5356#
MO3     014407            4762    5357#
MO4     014416            4763    5358#
MO5     014433            4764    5359#
MO6     014440            4765    5360#
MSG     002734            4690    4698#
MSGLIM= 000017            4403#   6511    7468    7563    7591    7597    7712    7727    7967    7973
MSGTRN  015053            5401#   7556    7565    7585    7599
MSGTRU  015104            5402#   6386
MSGTYP  007226            4821#   6427    7474*   7486*   7751*   7819*   7838*   7840*   7842*   7844*   7846*   7848*   7851*
                          7994*
MSG0    002220            4603    4618    4629#
MSG0C   002150            4603#
MSG1    002221            4604    4619    4631#
MSG1C   002152            4604#
MSG2    002222            4605    4620    4633#
MSG2C   002154            4605#
MSG3    002223            4606    4621    4635#
MSG3C   002156            4606#
MSG4    002224            4607    4622    4637#
MSG4C   002160            4607#
MSG5    002324            4608    4623    4643#
MSG5C   002162            4608#   7475    7487    7752    7849
MSG6    002416            4609    4624    4648#
MSG6C   002164            4609#   7852
MSG8    002646            4611    4626    4664#
MSG8C   002170            4611#
NEW     030612            6984    7025#   7029
NO    = 000036            4514#   5084    7897    7900    7926
NOBUF   007212            4813#   7316    7983*   8344    9137*
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 27-14
CZKMUA.P11    30-MAR-82 09:13          CROSS REFERENCE TABLE -- USER SYMBOLS

NOCLK   014714            5398#   7014    7455
NOD0    010502            5011#
NOD1    010506            5012#
NOD10   010564            5019#
NOD100  011402            5099#
NOD101  011406            5100#
NOD102  011422            5101#
NOD103  011426            5116#
NOD104  011442            5117#
NOD105  011446            5118#
NOD106  011462            5119#
NOD107  011466            5121#
NOD11   010566            5020#
NOD110  011502            5122#
NOD111  011506            5125#
NOD112  011512            5128#
NOD113  011526            5129#
NOD114  011532            5130#
NOD115  011550            5131#
NOD116  011554            5132#
NOD117  011570            5133#
NOD12   010600            5021#
NOD120  011574            5134#
NOD121  011610            5135#
NOD122  011614            5136#
NOD123  011630            5137#
NOD124  011634            5138#
NOD125  011650            5139#
NOD126  011654            5140#
NOD127  011670            5141#
NOD13   010604            5022#
NOD130  011674            5142#
NOD131  011714            5143#
NOD132  011720            5146#
NOD133  011724            5147#
NOD134  011730            5148#
NOD135  011734            5149#
NOD136  011740            5150#
NOD137  011744            5151#
NOD14   010620            5023#
NOD140  011750            5152#
NOD141  011752            5155#
NOD142  011756            5156#
NOD143  011762            5157#
NOD144  011776            5158#
NOD145  012002            5159#
NOD146  012016            5160#
NOD147  012022            5163#
NOD15   010624            5024#
NOD150  012026            5164#
NOD151  012032            5165#
NOD152  012036            5168#
NOD153  012042            5179#
NOD154  012064            5180#
NOD155  012070            5181#
NOD156  012104            5182#
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 27-15
CZKMUA.P11    30-MAR-82 09:13          CROSS REFERENCE TABLE -- USER SYMBOLS

NOD157  012110            5183#
NOD16   010640            5025#
NOD160  012132            5184#
NOD161  012136            5185#
NOD162  012160            5186#
NOD163  012164            5189#
NOD164  012170            5190#
NOD165  012174            5191#
NOD166  012200            5196#
NOD167  024024            6059#
NOD17   010644            5026#
NOD170  024030            6060#
NOD171  024034            6061#
NOD172  024036            6062#
NOD173  024052            6063#
NOD174  024054            6064#
NOD175  024070            6065#
NOD176  024072            6066#
NOD177  024104            6067#
NOD2    010512            5013#
NOD20   010650            5027#
NOD200  024106            6068#
NOD201  024122            6069#
NOD202  024126            6070#
NOD203  024132            6071#
NOD204  024146            6072#
NOD205  024150            6073#
NOD206  024164            6074#
NOD207  024166            6075#
NOD21   010662            5028#
NOD210  024204            6076#
NOD211  024210            6077#
NOD212  024214            6078#
NOD213  024216            6079#
NOD214  024220            6080#
NOD22   010666            5029#
NOD23   010700            5030#
NOD24   010704            5031#
NOD25   010706            5035#
NOD26   010712            5036#
NOD27   010726            5037#
NOD3    010514            5014#
NOD30   010732            5038#
NOD31   010750            5039#
NOD32   010754            5040#
NOD33   010772            5041#
NOD34   010776            5042#
NOD35   011014            5043#
NOD36   011020            5044#
NOD37   011036            5045#
NOD4    010530            5015#
NOD40   011042            5046#
NOD41   011066            5047#
NOD42   011072            5048#
NOD43   011076            5049#
NOD44   011114            5050#
```

```
NOD45   011120            5051#
NOD46   011132            5052#
NOD47   011136            5056#
NOD5    010532            5016#
NOD50   011142            5057#
NOD51   011160            5058#
NOD52   011162            5059#
NOD53   011202            5060#
NOD54   011204            5065#
NOD55   011210            5066#
NOD56   011226            5067#
NOD57   011232            5068#
NOD6    010546            5017#
NOD60   011252            5069#
NOD61   011256            5072#
NOD62   011262            5073#
NOD63   011266            5074#
NOD64   011272            5075#
NOD65   011276            5076#
NOD66   011302            5077#
NOD67   011306            5078#
NOD7    010550            5018#
NOD70   011312            5079#
NOD71   011316            5082#
NOD72   011322            5083#
NOD73   011326            5084#
NOD74   011340            5085#
NOD75   011344            5086#
NOD76   011360            5087#
NOD77   011364            5098#
NONE  = 000000            4418#
NOTNUF= 000050            4524#  5022  5025  5026  5036  5075  5077  5083  5156
NULEVT  016073            5446#  6208
NULL  = 000000            4484#
NUM   = 000014            4496#  5164
NXMM    022046            5586#  9166
N10$    010506            5011   5012#
N100$   011136            5025   5028  5056#
N102$   011142            5057#
N104$   011162            5057   5059#
N110$   011204            5030   5065#
N111$   011210            5066#
N112$   011232            5066   5068#
N114$   011322            5082   5083#
N115$   011316            5039   5041  5043  5045  5047  5050  5052  5082#  5087  5099  5101  5122  5180
                          5182   5184  5186  5191
N116$   011340            5084   5085#
N117$   011364            5085   5086  5098#
N118$   011406            5098   5100#
N120$   011506            5067   5069  5125#
N121$   011752            5129   5131  5133  5135  5137  5139  5141  5148  5155#  5165
N122$   012002            5157   5159#
N123$   011756            5155   5156#
N125$   012200            5075   5077  5079  5083  5143  5156  5196#
N126$   012022            5158   5160  5163#
N130$   011426            5100   5116#
```

```
N131$   011446          5116    5118#
N132$   011466          5118    5121#
N140$   012036          5119    5168#
N141$   012042          5179#
N142$   012070          5179    5181#
N143$   012110          5181    5183#
N144$   012136          5183    5185#
N150$   012164          5117    5189#
N20$    010644          5024    5026#
N25$    010666          5027    5029#
N30$    010704          5026    5029    5031#   5035    5036    5037    5048    5051    5056    5059    5065    5068    5074
                        5076    5078    5121    5125    5146    5159    5163    5164    5168    5185    5189    5190
N40$    010624          5022    5024#
N42$    010514          5012    5014#
N43$    010532          5014    5016#
N44$    010550          5016    5018#
N45$    010566          5018    5020#
N46$    010604          5020    5022#
N50$    011256          5023    5072#
N51$    011262          5072    5073#
N52$    011266          5073    5074#
N60$    011512          5128#
N61$    011532          5128    5130#
N62$    011554          5130    5132#
N63$    011574          5132    5134#
N64$    011614          5134    5136#
N65$    011634          5136    5138#
N66$    011654          5138    5140#
N67$    011674          5140    5142#
N68$    011720          5142    5146#
N70$    011724          5147#   5151
N71$    011734          5147    5149#
N72$    011740          5149    5150#
N73$    011750          5150    5152#
N80$    010706          5021    5035#
N81$    010712          5036#
N82$    010754          5038    5040#
N83$    010776          5040    5042#
N84$    011020          5042    5044#
N85$    011042          5044    5046#
N86$    011072          5046    5048#
N87$    011120          5049    5051#
OFSET   007242          4828#   5659    8299*   8312*
OPBFPT  002520          4657#   8574
OPBUF   002524          4625    4658#   7829    8502    8506    8508    8514    8515    8527    8529    8557    8558    8575
                        8577
OPCNT   002166          4610#   7827*   8512*   8516    8517
OPEND   002646          4659#   8504
OPRMM   014576          5387#   8506
OPRMSG= 000015          4497#   5146
OTINT = 000002          4459#   8912    8990    9071    9113
OUTHDL  045560          8915    9075    9112#
OUTHEX  046240          9138    9177    9185    9191#
OUTH1   045650          9115    9127#
OUTH2   046154          9129    9179#
OUTH3   046210          9181    9187#
```

```
OUTH4   046216            9188#
OUTH6   045712            9135    9140#
OUTVEC  012224            5229#   7066*   7067*   7109
O$APTS= 000000            4149#   4227
O$AU  = 000001            4149#   4207#   4227
O$BGNR= 000001            4149#   4207#   4227
O$BGNS= 000000            4149#   4227
O$DU  = 000001            4149#   4207#   4227
O$ERRT= 000000            4149#   4227
O$GNSW= 000000            4149#   4227
O$POIN= 000001            4149#   4207#   4227
O$SETU= 000000            4149#   4227    9359
PARAM   007314            4859#   5949    5981    7493*   7499    7776    7891*   7902*   7907*   7925*   7928*   7957    8041*
                          8175    8180    8227    8272    8389*   8501*   8555*
PAS   = 000002            4412#   7872    7903    8198    8215
PASC  = 000042            4518#   5190
PASMOD= 000030            4508#   5040
PASS1   002650            4671#   8384*
PASS2   002651            4672#   8385*
PASS3   002652            4673#   8386*
PASS4   002653            4674#   8387*
PCADD   007300            4848#   5681    5686    9013*   9057*   9112*
PCK     014540            5370#   6585
PCLKCT= 001000            4429#   7001
PCLKEN= 000111            4428#   7003
PCPM    015330            5414#   5681    5686
PEC     014550            5372#   6589
PLCK    041544            4872    8066#
PLCK2   041544            8067#
PLCK3   041560            8069#
PMS     014557            5385#   6593
PNCK    014536            5369#   6588
PNEC    014546            5371#   6592
PNMS    014555            5384#   6596
PNST    014525            5367#   6584
PNT   = 001000 G          4396#
PRI   = 002000 G          4396#
PRIOR   046440            9274    9297#
PRI00 = 000000 G          4396#   7112
PRI01 = 000040 G          4396#
PRI02 = 000100 G          4396#
PRI03 = 000140 G          4396#
PRI04 = 000200 G          4396#
PRI05 = 000240 G          4396#
PRI06 = 000300 G          4396#
PRI07 = 000340 G          4227    4396#   7310
PRNT  = 000055            4529#   5016    7534    7702
PROEM   022026            5585#   9169
PROTO = 000041            4517#
PROTOB= 000040            4438#   7923
PSCNT   007214            4814#   7317    7984*   8342*   8345
PST     014527            5368#   6581
PTREND  007150            4791#
PTRTAB  006400            4790#   6514    6519    7465    7561    7590    7711    7726    7737    7744    7965
P$ACT   003366            4777#   6037*   6685    7517*
P$BUFA  003362            4775#   6035*   6640    7515*
```

CZKMUA0 KMS11-BL PDP-11 D. T MACY11 30A(1052) 30-MAR-82 09:15 PAGE 27-19
CZKMUA.P11 30-MAR-82 09: 3 CROSS REFERENCE TABLE -- USER SYMBOLS

```
PSCNT   003370  4778# 6777* 6784  6841* 6848* 6853
PSEXIT  027270  6643  6678# 6702  6707
PSGDBD  003377  4782# 6030* 6040  6144* 6700* 6705* 6713  6759* 6797* 7512* 7520  7811* 7859*
                7906* 7947*
PSNNUF  003376  4781# 6031* 6045  6098* 6139* 6191* 7513* 7524  7685* 7707* 7720* 7739* 7787*
                7865* 7873* 7892* 7950*
PSNUM   003372  4779# 6141  6143  6146  6152  6154  6772* 6775* 6777  6778* 6780* 6784* 6786*
                6792* 7780  7786  7808  7813  7815  7932
PSRADX  003374  4780# 6737* 6740* 6747* 6754  6782  6790
PSTREE  003364  4776# 6036* 6641  7516*
PSTRV   027152  6039  6639# 7519
PSTR5   027162  6642# 6651  6671  6676
QCOPY = 000013  4495# 5159  7803
QRX   = 000004  4460# 8020  8070  8097  8148  8170  8200  8207  8218  8414  8449  8561  8789
                8792  8946
QTX   = 000010  4461# 8044  8097  8158  8201  8209  8217  8226  8427  8431  8444  8519  8531
                8874  8876  8931
QUALFG  003204  4714# 6038* 7518* 7800* 7803* 7806  7897* 7900  7908* 7926  7929*
QUALVL  003206  4715# 7578* 7612* 7797* 7815* 7816*
RAMO  = 002000  4581# 7074
RDI   = 000200  4584# 9019
RDO   = 000200  4585# 9114  9191
REC   = 000000  4410#
RECMOD= 000031  4509# 5042
REPLOG  025104  6114  6202#
REPORT  023604  6023# 6888  7703
RESFLG  007304  4850# 6975* 6987* 7457
RESTRT  030552  6979  6995  7004  7012  7016#
RHLPEN  003246  4733# 6104
RHLPTB  003230  4726# 6102
RHLP0   013632  5336# 6028
RHLP1   013671  4726  5337#
RHLP2   013714  4727  5338#
RHLP3   013747  4728  5339#
RHLP4   014000  4729  5340#
RHLP5   014032  4730  5341#
RHLP6   014071  4731  5342#
RHLP7   014130  4732  5343#
RI    = 000200  4563# 4910
RMDLOP= 000047  4523# 5185
RNOTNF= 000007  4540# 6068  6070  6075
RPASS   007316  4866# 5948  7492* 7498  7504  7708* 7775  7932* 8348  8350*
RPBASE  024646  6126  6135  6155  6177#
RPEXT = 000002  4535# 6050  6064  6110
RPHLP = 000001  4534# 6060  6062  6106
RPLOG = 000003  4536# 6066  6115
RPSWE = 000004  4537# 6071
RPSWF = 000005  4538# 6073
RPSWO = 000006  4539# 6077
RPT     025152  4980  6211# 6239  6246  6257  6265  6272  6280
RPTAA   025222  6226#
RPTABO  025536  4984  6259#
RPTDCK  025666  4979  6275#
RPTDDE  025612  4982  6268#
RPTDER  025376  4977  6241#
RPTDLE  025666  4981  6274#
```

```
RPTDSP  010442           4973#   6232
RPTDVI  025472           4976    6248#
RPTEOP  025536           4983    6260#
RPTIV   014202           5344#   6143
RPTMSB  025752           6238    6291#
RPTTXQ  025324           4973    4974    4975    4976    6234#
RPTO    025216           6207    6218    6222    6225#
RPT1    025204           6215    6221#
RQI   = 000040           4586#   9014
RSEL4   046254           8938    8944    9188*   9195#
RSEL6   046256           8939    8945    9189*   9190*   9196#
RTS   = 000040           4562#   4909
RUN   = 000004           4488#   5020    7536    7706
RUNSBM  022175           5593#
RXBIT = 000004           4587#   9180
RXBUF   004400           4788#   7992    8415    8416    8435    8456    8462    8468
RXC   = 000006           4446#   5936    5979
RXIDM   022217           5594#
RXMTOT  007206           4810#   7979*   8019    8095    8276
RXM1    022317           5599#   8459
RXM2    022342           5600#   8464
RXNC    022277           5598#   8451
RXONLY  041452           4870    8017#
RXON2   041452           8018#
RXPTR   007150           4793#   7467*   7976*   7977*   7993    8018    8069    8096    8183    8274
RXQ   = 000004           4445#   5932
R10$    024030           6059    6060#
R11$    024036           6060    6062#
R12$    024054           6062    6064#
R125$   024220           6070    6080#
R13$    024072           6064    6066#
R14$    024106           6066    6068#
R15$    024126           6069    6070#
R16$    024150           6071    6073#
R17$    024166           6073    6075#
R30$    024216           6068    6075    6076    6077    6079#
SCM     020047           5505#   5953
SCMD    020102           5508#   5961
SCML    020071           5507#   5957
SDVE    020036           5504#   5939
SDVI    020060           5506#   5944
SECRM   020135           5515#   8475
SEL0    012202           5214#   7049*   7074*   8647*   8648*   8650    8657    8675*   8678*   8891    8904    9014*   9019
                         9024    9064
SEL2    012206           5217#   7052*   7053*   8658    8674*   8892    8905    9025    9063    9114    9117    9127    9142
                         9180    9191*
SEL4    012212           5220#   7056*   7057*   7075*   8646*   8672*   8741    8795    8796*   8879    8880*   9183    9188
SEL6    012216           5223#   7060*   7061*   7076*   7077    8645*   8673*   8685*   8688*   8693*   8695    8742*   8797*
                         8881*   9118    9131    9184    9189
SEOP    020113           5509#   5965
SETET = 000060           4532#   5142    7529    7862
SETEXP= 000010           4492#   5066    7551    7793    7857
SETTRN= 000011           4493#   5068    7796
SHF0    015167           5403#   6578
SHF1    015225           5411#   6616
SHMSG   014242           5345#   7769
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 27-21
CZKMUA.P11    30-MAR-82 09:13          CROSS REFERENCE TABLE -- USER SYMBOLS

SHOW  = 000002            4486#  5027   7549   7691   7721   7740
SHTAB   003322            4755#  7760   7766
SHTEND  003331            4756#  7763
SHTYP0  014276            4750   5346#
SHTYP1  014305            4750   5347#
SHTYP2  014312            4750   5348#
SHTYP3  014317            4750   5349#
SHTYP4  014324            4750   5350#
SHTYP5  014332            4750   5351#
SHTYP6  014337            4750   5352#
SHTYP7  014345            4750   5353#
SHTYTB  003302            4750#  7769
SHWOP   026650            6255   6567#  7500   7777
SIZE  = 000012            4494#  5157   7800   7806
SMSC    020124            5510#
SQD   = 040000            4564#  4911
SRXQ    020025            5503#  5931
STADD   007220            4816#  6338   7780*
START   030352            6977   6987#
STATB = 000001            4433#  5981   6582   7914
STATUS= 000016            4498#  5098
STXC    020014            5502#  5926
STXQ    020003            5501#  5921
SVCGBL= 000000            4149#  4158#  4227   4254   4272   5270   5290   5658   5662   5666   5679   5684   5689
                          5693   5857   6874   6915   6944   7281   7299   7346   7383   8975   8989   9241   9359#
SVCINS= 000001            4149#  4155#  4227   4254   4272   5270   5290   5659   5660   5663   5664   5667   5668
                          5680   5681   5682   5685   5686   5687   5690   5691   5694   5695   5697   5881   5989
                          5992   6028   6034   6042   6047   6103   6143   6179   6188   6208   6226   6231   6237
                          6244   6245   6263   6271   6278   6291   6306   6340   6344   6346   6386   6578   6616
                          6758   6796   6906   6973   6976   6977   6978   6979   6980   6981   6983   6984   6991
                          6992   6997   6998   7006   7007   7011   7014   7031   7032   7084   7085   7086   7087
                          7096   7108   7109   7112   7114   7270   7290   7310   7320   7337   7357   7373   7394
                          7410   7455   7495   7502   7503   7506   7514   7522   7526   7546   7556   7565   7585
                          7599   7695   7769   7810   7835   7905   7949   7961   8295   8309   8318   8383   8403
                          8407   8475   8506   8556   8574   8652   8661   8895   8899   8909   8926   8941   8977
                          8991   9018   9028   9067   9070   9121   9125   9176   9224   9241   9255   9272   9273
                          9274   9276   9359
SVCSUB= 000001            4149#  4157#
SVCTAG= 000001            4149#  4159#  4318   5660   5664   5668   5682   5687   5691   5695   5881   6034   6906
                          7011   7270   7290   7337   7373   7410   7514   8383   8506   8977   8991   9224   9276
SVCTST= 000001            4149#  4156#  7436
S$LSYM= 010000            4149#  4318#  5660#  5664#  5668#  5682#  5687#  5691#  5695#  5881#  6034#  6906#  7011#
                          7270#  7290#  7337#  7373#  7410#  7514#  8383#  8506#  8977#  3991#  9224#  9276#

S1      030340            6981   6983#
S2      030414            6992   6997#
S3      030464            6998   7006#
S4      030532            7007   7014#
TABEX   014775            5399#  7565   7599
TAL   = 000005            4415#  7889   8579
TALCK   043544            4875   8500#  8528   8530
TALMOD= 000035            4513#  5051
TCURAD  007176            4805#  7476*  7479   7571   7576*  7747*
TEMP    007244            4829#  5922*  5927*  5932*  5936*  5940*  5945*  5954*  5958*  5962*  5966*  5979   5995
                          5996*  5997*  5998   6343*  6344   6388*  6389*  6390   6429*  6430*  6436   6569*  6578
                          6581*  6584*  6616   7315*  7768*  7769   7820*  7821*  7825   7828   8189*  8190*  8194
                          8467*  8473*  8475
```

```
TEMP1   007246          4830# 5921* 5926* 5931* 5939* 5944* 5953* 5957* 5961* 5965* 5992  6183* 6188
                        6576* 6578  6585* 6588* 6616
TEMP2   007250          4831# 5946* 5947* 6001  6187* 6188  6432* 6434* 6438  6577* 6578  6589* 6592*
                        6616  7081* 7086  7317* 8151* 8161* 8172* 8178  8192* 8193  8211* 8280* 8345*
                        8416* 8423* 8437* 8441* 8453* 8515* 8524* 8558* 8569* 8656* 8890* 8903* 8922*
                        8937* 9023* 9062* 9116* 9141*
TEMP3   007252          4832# 5667  5680  5685  5690  5694  5948* 6002  6182* 6188  6572* 6575* 6578
                        7082* 7086  7318* 8153* 8163* 8173* 8179  8190  8191  8192  8212* 8282* 8292
                        8346* 8383  8384  8385  8386  8387  8418* 8425* 8435* 8442* 8454* 8516* 8525*
                        8560* 8570* 8657* 8891* 8904* 8923* 8938* 9024* 9064* 9117* 9142*
TEMP4   007254          4833# 5663  5680  5685  5690  5694  5949* 5976* 6003  7083* 7086  7316* 8289*
                        8298* 8303* 8304  8315  8344* 8436* 8658* 8892* 8905* 8924* 8939* 9025* 9063*
                        9118* 9131* 9134  9146  9149  9152  9155  9158  9161  9164  9167  9171
TEMP5   007256          4834# 6593* 6596* 6616
TIMERS  007364          4892# 5875  5879* 8884* 8888
TIMER1  007360          4890# 5869  5871* 7450* 7451  8644* 8653  9015* 9016  9058* 9059
TIMER2  007362          4891# 5872  5874*
TIMMIN  007352          4886# 5866* 6000  7016*
TIMOM   022151          5591# 9151
TIMSEC  007354          4887# 5863* 5864  5867* 5999  7017*
TIMTCK  007356          4888# 5860* 5862* 5877  5997  7018*
TM    = 001000          4565# 4912
TOINOT  044674          8885# 8910  8913  8917
TOIN1   044756          8889  8899#
TOIN2   045042          8901  8912#
TOORIO  045436          8671  8684  8740  8794  8878  9057# 9068
TOOR1   045522          9060  9070#
TOOR2   045540          9073  9076#
TOOR3   045450          9059# 9077
TOTCC   007240          4827# 6380* 6381  6387* 6389  6391* 7462* 7553* 7554  7575  7582* 7583  7611
TRA   = 000001          4411# 7886
TRAMOD= 000034          4512# 5049
TRVACT  027272          6672  6683# 6699  6704  6709  6712  6732  6793  6811  6832  6856
TRVALN  030064          6661  6815#
TRVALP  030020          6660  6801#
TRVBIF  027376          6657  6712#
TRVBR   027366          6656  6709#
TRVBRC  027312          6670  6690# 6710  6715  6734  6798  6813  6834  6860
TRVDEC  027472          6663  6737#
TRVERR  027330          6654  6699#
TRVEXI  027350          6655  6704#
TRVNMA  027512          6738  6741#
TRVNOB  027322          6695# 6716  6733  6794  6812  6833
TRVNUM  027504          6659  6740#
TRVOCT  027504          6662  6739#
TRVSPA  027420          6658  6718#
TRVSTR  030152          6664  6838#
TSEL4   046250          8923  8929  9183* 9193#
TSEL6   046252          8924  8930  9184* 9194#
TTL   = 000001          4419# 7507  8676
TTLLOP= 000044          4520# 5179
TTOTCC  007174          4804# 6538  7463* 7553  7558  7575* 7743*
TXBUF   003400          4787# 6506  7476  7746  8436
TXC   = 000002          4444# 5927
TXMTOT  007172          4803# 6526  7482* 7560* 7562  7577* 7738  7742* 7959  8043  8067  8093
TXNC    022257          5597# 8434
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 27-23
CZKMUA.P11    30-MAR-82 09:13          CROSS REFERENCE TABLE -- USER SYMBOLS

TXONLY  041504            4871    8041#
TXON2   041512            8042#
TXPTR   007152            4794#   6519*   6521*   6522    6531*   6533    7466*   7480    7561*   7569*   7570    7574*   7744*
                          7745    7966*   8042    8068    8094
TXQ    = 000000           4443#   5922
T$ARGC= 000001            4227#   5659#   5663#   5667#   5680#   5681#   5685#   5686#   5690#   5694#   5989#   5992#   6028#
                          6042#   6047#   6103#   6143#   6179#   6188#   6208#   6226#   6231#   6237#   6244#   6245#   6263#
                          6271#   6278#   6291#   6306#   6340#   6344#   6346#   6386#   6578#   6616#   6758#   6796#   7014#
                          7084#   7085#   7086#   7455#   7495#   7522#   7526#   7556#   7565#   7585#   7599#   7695#   7769#
                          7810#   7835#   7905#   7949#   7961#   8403#   8475#   8556#   8574#
T$CODE= 003032            6034#   7011#   7514#   8383#   8506#   9255#   9272#   9273#   9274#
T$ERRN= 000023            4149#   8295#   8309#   8318#   8407#   8661#   8895#   8909#   8926#   8941#   9028#   9067#   9121#
                          9176#
T$EXCP= 000000            6034#   7011#   7514#   8383#   8506#   9272#   9273#   9274#
T$FLAG= 000040            5697#   7114#   7320#   7357#   7394#   7506#   7546#   9125#
T$GMAN= 000000            4149#   6034#   7011#   7514#   8383#   8506#
T$HILI= 000007            6034#   7011#   7514#   8383#   8506#   9272#   9273#   9274#
T$LAST= 000001            4149#   9359#
T$LOLI= 000004            6034#   7011#   7514#   8383#   8506#   9272#   9273#   9274#
T$LSYM= 010000            4149#   4318    5660    5664    5668    5682    5687    5691    5695    5881    6906    7270    7290
                          7337    7373    7410    8977    8991    9224    9276
T$LTNO= 000001            9359#
T$NEST= 177777            4149#   4170#   4272#   4318#   5658#   5660#   5662#   5664#   5666#   5668#   5679#   5682#   5684#
                          5687#   5689#   5691#   5693#   5695#   5857#   5881#   6874#   6906#   6915#   6921#   6944#   7270#
                          7281#   7290#   7299#   7337#   7346#   7373#   7383#   7410#   7436#   8975#   8977#   8989#   8991#
                          9224#   9241#   9276#   9360#
T$NS0 = 000000            4170#   9360
T$NS1 = 000004            4272#   4318    5658#   5660    5662#   5664    5666#   5668    5679#   5682    5684#   5687    5689#
                          5691    5693#   5695    5857#   5881    6874#   6906    6915#   6921    6944#   7270    7281#   7290
                          7299#   7337    7346#   7373    7383#   7410    7436#   9224    9241#   9276
T$NS2 = 000010            8975#   8977    8989#   8991
T$PTNU= 000000            4149#
T$SAVL= 177777            4149#
T$SEGL= 177777            4149#
T$SUBN= 000000            4149#   7436#
T$TAGL= 177777            4149#
T$TAGN= 010024            4149#   4272#   5658#   5662#   5666#   5679#   5684#   5689#   5693#   5857#   6874#   6915#   6944#
                          7281#   7299#   7346#   7383#   7436#   8975#   8989#   9241#
T$TEMP= 000000            4254#   4318#   5660#   5664#   5668#   5682#   5687#   5691#   5695#   5697#   5881#   6034#   6906#
                          6921#   7011#   7114#   7270#   7290#   7320#   7337#   7357#   7373#   7394#   7410#   7506#   7514#
                          7546#   8383#   8506#   8977#   8991#   9125#   9224#   9255#   9272#   9273#   9274#   9276#   9360#
T$TEST= 000001            4149#   7436#   9359
T$TSTM= 177777            4149#   5659    5660    5663    5664    5667    5668    5680    5681    5682    5685    5686    5687
                          5690    5691    5694    5695    5989    5992    6028    6034    6042    6047    6103    6143    6179
                          6188    6208    6226    6231    6237    6244    6245    6263    6271    6278    6291    6306    6340
                          6344    6346    6386    6578    6616    6758    6796    6906    6973    6976    6978    6980    6983
                          6991    6997    7006    7011    7014    7031    7084    7085    7086    7087    7096    7108    7109
                          7112    7114    7270    7290    7310    7320    7337    7373    7410    7455    7495    7502    7506
                          7514    7522    7526    7546    7556    7565    7585    7599    7695    7769    7810    7835    7905
                          7949    7961    8295    8309    8318    8383    8403    8407    8475    8506    8556    8574    8652
                          8661    8895    8899    8909    8926    8941    9018    9028    9067    9070    9121    9125    9176
                          9224
T$TSTS= 000001            4149#   7436#
T$$AU = 010017            7383#   7394    7410
T$$AUT= 010014            7281#   7290
T$$CLE= 010015            7299#   7320    7337
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 27-24
CZKMUA.P11    30-MAR-82 09:13          CROSS REFERENCE TABLE -- USER SYMBOLS

T$$DU = 010016            7346#  7357   7373
T$$HAR= 010023            9241#  9276
T$$HW = 010000            4272#  4318
T$$INI= 010013            6944#  7114   7270
T$$MSG= 010007            5658#  5660   5662#  5664   5666#  5668   5679#  5682   5684#  5687   5689#  5691   5693#
                          5695   5697
T$$PRO= 010012            6915#
T$$RPT= 010011            6874#  6906
T$$SRV= 010022            5857#  5881   8975#  8977   8989#  8991
T$$TES= 010020            7436#  7506   7546   9125   9224
T1      035570  G         4254   7436#
UAM   = 000200  G         4396#
UNKM    020273            4956   5529#  8467
UPTABL  042012            8175#
UPTA1   042100            8181   8188#
UPTA3   042076            8185   8187#
UPTA4   042036            8176   8180#
UPTEX   042150            8187   8197#
VECTOR  046405            9273   9296#
X$    = 000215            4173#  5011#  5012#  5013#  5014#  5015#  5016#  5017#  5018#  5019#  5020#  5021#  5022#
                          5023#  5024#  5025#  5026#  5027#  5028#  5029#  5030#  5031#  5035#  5036#  5037#  5038#
                          5039#  5040#  5041#  5042#  5043#  5044#  5045#  5046#  5047#  5048#  5049#  5050#  5051#
                          5052#  5056#  5057#  5058#  5059#  5060#  5065#  5066#  5067#  5068#  5069#  5072#  5073#
                          5074#  5075#  5076#  5077#  5078#  5079#  5082#  5083#  5084#  5085#  5086#  5087#  5098#
                          5099#  5100#  5101#  5116#  5117#  5118#  5119#  5121#  5122#  5125#  5128#  5129#  5130#
                          5131#  5132#  5133#  5134#  5135#  5136#  5137#  5138#  5139#  5140#  5141#  5142#  5143#
                          5146#  5147#  5148#  5149#  5150#  5151#  5152#  5155#  5156#  5157#  5158#  5159#  5160#
                          5163#  5164#  5165#  5168#  5179#  5180#  5181#  5182#  5183#  5184#  5185#  5186#  5189#
                          5190#  5191#  5196#  6059#  6060#  6061#  6062#  6063#  6064#  6065#  6066#  6067#  6068#
                          6069#  6070#  6071#  6072#  6073#  6074#  6075#  6076#  6077#  6078#  6079#  6080#
XSALWA= 000000            4149#
XSFALS= 000040            4149#
XSOFFS= 000400            4149#
XSTRUE= 000020            4149#
$PATCH  046466            9349#
.     = 046552            4145#  4652#  4658#  4690   4697   4711#  4757#  4787#  4788#  4789#  4790#  4897#  4898#
                          5014#  5018#  5022#  5029#  5036#  5038#  5044#  5046#  5057#  5059#  5066#  5068#  5084#
                          5086#  5098#  5100#  5116#  5118#  5128#  5130#  5132#  5134#  5136#  5142#  5157#  5159#
                          5183#  5290#  5389#  5488#  5489#  5490#  5601#  5697   6062#  6064#  6068#  6073#  6075#
                          6198#  7114   7248   7320   7357   7394   7506   7546   9125   9350#
```

```
BCOMPL     15#  4149#  6977  6979  6984  7503
BERROR     19#  4149#
BGNAU      23#  4149#  7383
BGNAUT     31#  4149#  7281
BGNCLN     39#  4149#  7299
BGNDU      47#  4149#  7346
BGNHRD     55#  4149#  9241
BGNHW      66#  4149#  4272
BGNINI     77#  4149#  6944
BGNMOD     85#  4149#  4170
BGNMSG     98#  4149#  5658  5662  5666  5679  5684  5689  5693
BGNPRO    106#  4149#  6915
BGNPTA    114#  4149#
BGNRPT    144#  4149#  6874
BGNSEG    152#  4149#
BGNSET    161#  4149#
BGNSFT    182#  4149#
BGNSRV    193#  4149#  5857  8975  8989
BGNSUB    201#  4149#
BGNSW     225#  4149#
BGNTST    236#  4149#  7436
BNCOMP    266#  4149#  6981  6992  6998  7007  7032
BNERRO    270#  4149#
BREAK     274#  4149#  8652  8899  9018  9070
BRESET    278#  4149#
CKLOOP    282#  4149#
CLI      4177#  5011  5012  5013  5014  5015  5016  5017  5018  5019  5020  5021  5022  5023  5024
         5025  5026  5027  5028  5029  5030  5031  5035  5036  5037  5038  5039  5040  5041  5042
         5043  5044  5045  5046  5047  5048  5049  5050  5051  5052  5056  5057  5058  5059  5060
         5065  5066  5067  5068  5069  5072  5073  5074  5075  5076  5077  5078  5079  5082  5083
         5084  5085  5086  5087  5098  5099  5100  5101  5116  5117  5118  5119  5121  5122  5125
         5128  5129  5130  5131  5132  5133  5134  5135  5136  5137  5138  5139  5140  5141  5142
         5143  5146  5147  5148  5149  5150  5151  5152  5155  5156  5157  5158  5159  5160  5163
         5164  5165  5168  5179  5180  5181  5182  5183  5184  5185  5186  5189  5190  5191  5196
         6059  6060  6061  6062  6063  6064  6065  6066  6067  6068  6069  6070  6071  6072  6073
         6074  6075  6076  6077  6078  6079  6080
CLOCK     286#  4149#  6991  6997
CLOSE     292#  4149#
CLRVEC    296#  4149#
COMMEN    301#  4149#
DELAY     322#  4149#
DESCRI    317#  4149#  5290
DEVTYP    341#  4149#  5270
DISPAT    346#  4149#  4254
DISPLA    360#  4149#
DOCLN     376#  4149#  6973  7087
DODU      380#  4149#
DORPT     385#  4149#
ENDAU     389#  4149#  7410
ENDAUT    401#  4149#  7290
ENDCLN    413#  4149#  7337
ENDCOM    425#  4149#
ENDDU     441#  4149#  7373
ENDHRD    453#  4149#  9276
ENDHW     465#  4149#  4318
ENDINI    475#  4149#  7270
```

CZKMUA0 KMS11-BL PDP-11 DCLT MACY11 30A(1052) 30-MAR-82 09:15 PAGE 28-1
CZKMUA.P11 30-MAR-82 09:13 CROSS REFERENCE TABLE -- MACRO NAMES

```
ENDMOD   487#  4149#  9360
ENDMSG   500#  4149#  5660  5664  5668  5682  5687  5691  5695
ENDPRO   512#  4149#  6921
ENDPTA   520#  4149#
ENDRPT   529#  4149#  6906
ENDSEG   541#  4149#
ENDSET   555#  4149#
ENDSFT   568#  4149#
ENDSRV   580#  4149#  5881  8977  8991
ENDSUB   596#  4149#
ENDSW    614#  4149#
ENDTST   624#  4149#  9224
EQUALS   642#  4149#  4396
ERRDF    714#  4149#
ERRHRD   718#  4149#  8407
ERROR    722#  4149#
ERRSF    726#  4149#
ERRSOF   730#  4149#  8295  8309  8318  8661  8895  8909  8926  8941  9028  9067  9121  9176
ERRTBL   734#  4149#
ESCAPE   744#  4149#  9125
EXIT     771#  4149#  5697  7114  7320  7357  7394  7506  7546
FEQUAL   810#  4149#
GETBYT   824#  4149#
GETPRI   834#  4149#
GETWOR   829#  4149#
GMANIA   839#  4149#
GMANID   848#  4149#  6034  7011  7514  8383  8506
GMANIL   859#  4149#
GPHARD   868#  4149#  7031
GPRMA    874#  4149#  9272  9273
GPRMD    903#  4149#  6034#  7011#  7514#  8383#  8506#  9274
GPRML    934#  4149#  9255
HEADER   954#  4149#  4227
INLOOP   962#  4149#
IOSETU   966#  4149#
IOSTAR   974#  4149#
KT11     982#  4149#
LASTAD  1147#  4149#  9359
MANUAL  1162#  4149#  7502
MEMORY  1166#  4149#
MSBYTE  2000#  4149#  4227#
MSCHEC  2118#  4149#  5697#  7114#  7320#  7357#  7394#  7506#  7546#
MSCNTO  2182#  4149#  6034#  7011#  7514#  8383#  8506#  9255#  9272#  9273#  9274#
MSCOUN  2066#  4149#  5659#  5663#  5667#  5680#  5681#  5685#  5686#  5690#  5694#  5989#  5992#  6028#  6042#
        6047#  6103#  6143#  6179#  6188#  6208#  6226#  6231#  6237#  6244#  6245#  6263#  6271#  6278#  6291#
        6306#  6340#  6344#  6346#  6386#  6578#  6616#  6758#  6796#  7014#  7084#  7085#  7086#  7455#  7495#
        7522#  7526#  7556#  7565#  7585#  7599#  7695#  7769#  7810#  7835#  7905#  7949#  7961#  8403#  8475#
        8556#  8574#
MSDATA  1867#  4149#  4227#  5270#  5290#
MSDECR  2029#  4149#  4318#  5660#  5664#  5668#  5682#  5687#  5691#  5695#  5881#  6906#  6921#  7270#  7290#
        7337#  7373#  7410#  8977#  8991#  9224#  9276#  9360#
MSDEFA  2170#  4149#  6034#  7011#  7514#  8383#  8506#  9255#  9272#  9273#  9274#
MSENDE  2074#  4149#  4318#  5660#  5664#  5668#  5682#  5687#  5691#  5695#  5881#  6906#  7270#  7290#  7337#
        7373#  7410#  8977#  8991#  9224#  9276#  9360#
MSERRI  1649#  4149#  8295#  8309#  8318#  8407#  8661#  8895#  8909#  8926#  8941#  9028#  9067#  9121#  9176#
MSESCA  2006#  4149#  9125#
```

```
MSESCS  2010#  4149#  9125#
MSEXCP  2101#  4149#  6034#  7011#  7514#  8383#  8506#  9272#  9273#  9274#
MSEXIT  2014#  4149#  5697#  7114#  7320#  7357#  7394#  7506#  7546#
MSEXSE  2022#  4149#  5697#  7114#  7320#  7357#  7394#  7506#  7546#
MSEXTJ  2018#  4149#  5697#  7114#  7320#  7357#  7394#  7506#  7546#
MSGEN   2038#  4149#  4227#  4254#  4272#  4318#  5270#  5290#  5658#  5660#  5662#  5664#  5666#  5668#  5679#
        5682#  5684#  5687#  5689#  5691#  5693#  5695#  5857#  5881#  6034#  6874#  6906#  6915#  6944#  7011#
        7270#  7281#  7290#  7299#  7337#  7346#  7373#  7383#  7410#  7436#  7514#  8383#  8506#  8975#  8977#
        8989#  8991#  9224#  9241#  9276#  9359#
MSGENB  1938#  4149#  6034#  7011#  7514#  8383#  8506#
MSGETS  2035#  4149#  4318#  5660#  5664#  5668#  5682#  5687#  5691#  5695#  5881#  6906#  6921#  7270#  7290#
        7337#  7373#  7410#  8977#  8991#  9224#  9276#  9360#
MSGETT  1877#  4149#  5697#  7114#  7320#  7357#  7394#  7506#  7546#  9125#
MSGNGB  1902#  4149#  4170#  4227#  4254#  4272#  5270#  5290#  5658#  5662#  5666#  5679#  5684#  5689#  5693#
        5857#  6874#  6915#  6944#  7281#  7299#  7346#  7383#  8975#  8989#  9241#  9359#
MSGNIN  2049#  4149#  4227#  4254#  4272#  5270#  5290#  5659#  5660#  5663#  5664#  5667#  5668#  5680#  5681#
        5682#  5685#  5686#  5687#  5690#  5691#  5694#  5695#  5697#  5881#  5989#  5992#  6028#  6034#  6042#
        6047#  6103#  6143#  6179#  6188#  6208#  6226#  6231#  6237#  6244#  6245#  6263#  6271#  6278#  6291#
        6306#  6340#  6344#  6346#  6386#  6578#  6616#  6758#  6796#  6906#  6973#  6976#  6977#  6978#  6979#
        6980#  6981#  6983#  6984#  6991#  6992#  6997#  6998#  7006#  7007#  7011#  7014#  7031#  7032#  7084#
        7085#  7086#  7087#  7096#  7108#  7109#  7112#  7114#  7270#  7290#  7310#  7320#  7337#  7357#  7373#
        7394#  7410#  7455#  7495#  7502#  7503#  7506#  7514#  7522#  7526#  7546#  7556#  7565#  7585#  7599#
        7695#  7769#  7810#  7835#  7905#  7949#  7961#  8295#  8309#  8318#  8383#  8403#  8407#  8475#  8506#
        8556#  8574#  8652#  8661#  8895#  8899#  8909#  8926#  8941#  8977#  8991#  9018#  9028#  9067#  9070#
        9121#  9125#  9176#  9224#  9241#  9255#  9272#  9273#  9274#  9276#  9359#
MSGNLS  1913#  4149#  6034#  7011#  7514#  8383#  8506#
MSGNSU  1898#  4149#
MSGNTA  1890#  4149#  4318#  5660#  5664#  5668#  5682#  5687#  5691#  5695#  5881#  6906#  7270#  7290#  7337#
        7373#  7410#  8977#  8991#  9224#  9276#
MSGNTE  1894#  4149#  7436#
MSHAPT  1739#  4149#  4227#
MSHNAP  1824#  4149#  4227#
MSINCR  2026#  4149#  4170#  4272#  5658#  5659#  5660#  5662#  5663#  5664#  5666#  5667#  5668#  5679#  5680#
        5681#  5682#  5684#  5685#  5686#  5687#  5689#  5690#  5691#  5693#  5694#  5695#  5857#  5989#  5992#
        6028#  6034#  6042#  6047#  6103#  6143#  6179#  6188#  6208#  6226#  6231#  6237#  6244#  6245#  6263#
        6271#  6278#  6291#  6306#  6340#  6344#  6346#  6386#  6578#  6616#  6758#  6796#  6874#  6906#  6915#
        6944#  6973#  6976#  6978#  6980#  6983#  6991#  6997#  7006#  7011#  7014#  7031#  7084#  7085#  7086#
        7087#  7096#  7108#  7109#  7112#  7114#  7270#  7281#  7290#  7299#  7310#  7320#  7337#  7346#  7373#
        7383#  7410#  7436#  7455#  7495#  7502#  7506#  7514#  7522#  7526#  7546#  7556#  7565#  7585#  7599#
        7695#  7769#  7810#  7835#  7905#  7949#  7961#  8295#  8309#  8318#  8383#  8403#  8407#  8475#  8506#
        8556#  8574#  8652#  8661#  8895#  8899#  8909#  8926#  8941#  8975#  8989#  9018#  9028#  9067#  9070#
        9121#  9125#  9176#  9224#  9241#
MSIOSE  1700#  4149#
MSLDRO  1942#  4149#  6976#  6978#  6980#  6983#  6991#  6997#  7031#  7112#  7310#
MSMASK  1671#  4149#
MSMCHI     4#  4149#
MSMCLO  1624#  4149#
MSMSK1  1677#  4149#
MSPOP   1881#  4149#  4318#  5660#  5664#  5668#  5682#  5687#  5691#  5695#  5881#  6906#  6921#  7270#  7290#
        7337#  7373#  7410#  8977#  8991#  9224#  9276#  9360#
MSPRIN  1636#  4149#  5659#  5663#  5667#  5680#  5681#  5685#  5686#  5690#  5694#  5989#  5992#  6028#  6042#
        6047#  6103#  6143#  6179#  6188#  6208#  6226#  6231#  6237#  6244#  6245#  6263#  6271#  6278#  6291#
        6306#  6340#  6344#  6346#  6386#  6578#  6616#  6758#  6796#  7014#  7084#  7085#  7086#  7455#  7495#
        7522#  7526#  7556#  7565#  7585#  7599#  7695#  7769#  7810#  7835#  7905#  7949#  7961#  8403#  8475#
        8556#  8574#
MSPUSH  1631#  4149#  4170#  4272#  5658#  5662#  5666#  5679#  5684#  5689#  5693#  5857#  6874#  6915#  6944#
```

```
                7281#  7299#  7346#  7383#  7436#  8975#  8989#  9241#
M$PUT           1972#  4149#  5659#  5663#  5667#  5680#  5681#  5685#  5686#  5690#  5694#  5989#  5992#  6028#  6042#
                6047#  6103#  6143#  6179#  6188#  6208#  6226#  6231#  6237#  6244#  6245#  6263#  6271#  6278#  6291#
                6306#  6340#  6344#  6346#  6386#  6578#  6616#  6758#  6796#  7014#  7084#  7085#  7086#  7096#  7108#
                7109#  7455#  7495#  7522#  7526#  7556#  7565#  7585#  7599#  7695#  7769#  7810#  7835#  7905#  7949#
                7961#  8403#  8475#  8556#  8574#
M$PUT1          1981#  4149#  5659#  5663#  5667#  5680#  5681#  5685#  5686#  5690#  5694#  5989#  5992#  6028#  6042#
                6047#  6103#  6143#  6179#  6188#  6208#  6226#  6231#  6237#  6244#  6245#  6263#  6271#  6278#  6291#
                6306#  6340#  6344#  6346#  6386#  6578#  6616#  6758#  6796#  7014#  7084#  7085#  7086#  7096#  7108#
                7109#  7455#  7495#  7522#  7526#  7556#  7565#  7585#  7599#  7695#  7769#  7810#  7835#  7905#  7949#
                7961#  8403#  8475#  8556#  8574#
M$RADI          2077#  4149#  6034#  7011#  7514#  8383#  8506#  9255#  9272#  9273#  9274#
M$RBR0          1952#  4149#
M$RNR0          1962#  4149#  6991#  6997#  7031#
M$SETS          2032#  4149#  4170#  4272#  5658#  5662#  5666#  5679#  5684#  5689#  5693#  5857#  6874#  6915#  6944#
                7281#  7299#  7346#  7383#  7436#  8975#  8989#  9241#
M$STAR          1733#  4149#
M$SVC           1933#  4149#  5659#  5660#  5663#  5664#  5667#  5668#  5680#  5681#  5682#  5685#  5686#  5687#  5690#
                5691#  5694#  5695#  5697#  5989#  5992#  6028#  6034#  6042#  6047#  6103#  6143#  6179#  6188#  6208#
                6226#  6231#  6237#  6244#  6245#  6263#  6271#  6278#  6291#  6306#  6340#  6344#  6346#  6386#  6578#
                6616#  6758#  6796#  6906#  6973#  6976#  6978#  6980#  6983#  6991#  6997#  7006#  7011#  7014#  7031#
                7084#  7085#  7086#  7087#  7096#  7108#  7109#  7112#  7114#  7270#  7290#  7310#  7320#  7337#  7357#
                7373#  7394#  7410#  7455#  7495#  7502#  7506#  7514#  7522#  7526#  7546#  7556#  7565#  7585#  7599#
                7695#  7769#  7810#  7835#  7905#  7949#  7961#  8295   8309   8318   8383#  8403#  8407   8475#  8506#
                8556#  8574#  8652#  8661   8895   8899#  8909   8926   8941   9018#  9028   9067   9070#  9121   9125#
                9176   9224#
M$TLAB          1929#  4149#  5659#  5660#  5663#  5664#  5667#  5668#  5680#  5681#  5682#  5685#  5686#  5687#  5690#
                5691#  5694#  5695#  5989#  5992#  6028#  6034#  6042#  6047#  6103#  6143#  6179#  6188#  6208#  6226#
                6231#  6237#  6244#  6245#  6263#  6271#  6278#  6291#  6306#  6340#  6344#  6346#  6386#  6578#  6616#
                6758#  6796#  6906#  6973#  6976#  6978#  6980#  6983#  6991#  6997#  7006#  7011#  7014#  7031#  7084#
                7085#  7086#  7087#  7096#  7108#  7109#  7112#  7114#  7270#  7290#  7310#  7320#  7337#  7373#  7410#
                7455#  7495#  7502#  7506#  7514#  7522#  7526#  7546#  7556#  7565#  7585#  7599#  7695#  7769#  7810#
                7835#  7905#  7949#  7961#  8295#  8309#  8318#  8383#  8403#  8407#  8475#  8506#  8556#  8574#  8652#
                8661#  8895#  8899#  8909#  8926#  8941#  9018#  9028#  9067#  9070#  9121#  9125#  9176#  9224#
M$TSTL          1921#  4149#  5659#  5660#  5663#  5664#  5667#  5668#  5680#  5681#  5682#  5685#  5686#  5687#  5690#
                5691#  5694#  5695#  5989#  5992#  6028#  6034#  6042#  6047#  6103#  6143#  6179#  6188#  6208#  6226#
                6231#  6237#  6244#  6245#  6263#  6271#  6278#  6291#  6306#  6340#  6344#  6346#  6386#  6578#  6616#
                6758#  6796#  6906#  6973#  6976#  6978#  6980#  6983#  6991#  6997#  7006#  7011#  7014#  7031#  7084#
                7085#  7086#  7087#  7096#  7108#  7109#  7112#  7114#  7270#  7290#  7310#  7320#  7337#  7373#  7410#
                7455#  7495#  7502#  7506#  7514#  7522#  7526#  7546#  7556#  7565#  7585#  7599#  7695#  7769#  7810#
                7835#  7905#  7949#  7961#  8295#  8309#  8318#  8383#  8403#  8407#  8475#  8506#  8556#  8574#  8652#
                8661#  8895#  8899#  8909#  8926#  8941#  9018#  9028#  9067#  9070#  9121#  9125#  9176#  9224#
M$WORD          1994#  4149#  4227#  4254#  5697#  6034#  7011#  7114#  7320#  7357#  7394#  7506#  7514#  7546#  8295#
                8309#  8318#  8383#  8407#  8506#  8661#  8895#  8909#  8926#  8941#  9028#  9067#  9121#  9176#  9255#
                9272#  9273#  9274#  9359
M$XFER          1682#  4149#
NODCL           4182#  5011   5012   5013   5014   5015   5016   5017   5018   5019   5020   5021   5022   5023   5024
                5025   5026   5027   5028   5029   5030   5031   5035   5036   5037   5038   5039   5040   5041   5042
                5043   5044   5045   5046   5047   5048   5049   5050   5051   5052   5056   5057   5058   5059   5060
                5065   5066   5067   5068   5069   5072   5073   5074   5075   5076   5077   5078   5079   5082   5083
                5084   5085   5086   5087   5098   5099   5100   5101   5116   5117   5118   5119   5121   5122   5125
                5128   5129   5130   5131   5132   5133   5134   5135   5136   5137   5138   5139   5140   5141   5142
                5143   5146   5147   5148   5149   5150   5151   5152   5155   5156   5157   5158   5159   5160   5163
                5164   5165   5168   5179   5180   5181   5182   5183   5184   5185   5186   5189   5190   5191   5196
                6059   6060   6061   6062   6063   6064   6065   6066   6067   6068   6069   6070   6071   6072   6073
                6074   6075   6076   6077   6078   6079   6080
```

```
 CZKMUA0 KMS11-BL PDP-11 DCLT   MACY11 30A(1052)  30-MAR-82  09:15  PAGE 28-4
CZKMUA.P11    30-MAR-82 09:13          CROSS REFERENCE TABLE -- MACRO NAMES

OPEN      1171#   4149#
POINTE    1176#   4149#   4207
PRINTB    1239#   4149#   5659    5663    5667    5680    5681    5685    5686    5690    5694
PRINTF    1279#   4149#   5989    5992    6028    6042    6047    6103    6143    6179    6188    6340    6344    6346    6386
          6758    6796    7014    7084    7085    7086    7455    7495    7522    7526    7556    7565    7585    7599    7695
          7769    7810    7835    7905    7949    7961    8403    8475    8556    8574
PRINTS    1319#   4149#   6208    6226    6231    6237    6244    6245    6263    6271    6278    6291    6306    6578    6616
PRINTX    1359#   4149#
READBU    1399#   4149#   7006
READEF    1403#   4149#   6976    6978    6980    6983
RFLAGS    1408#   4149#
SETPRI    1413#   4149#   7112    7310
SETVEC    1418#   4149#   7096    7108    7109
SLASH     1424#   4149#
STARS     1438#   4149#
SVC       1452#   4148#   4149
XFER      1612#   4149#   5697#   7114#   7320#   7357#   7394#   7506#   7546#
XFERF     1616#   4149#
XFERT     1620#   4149#


. ABS.  046552     000


 ERRORS DETECTED:  0

 CZKMUA.BIN,CZKMUA.LST/CRF=SVC34R.MLB,CZKMUA.P11
 RUN-TIME: 27 34 4 SECONDS
 RUN-TIME RATIO: 75/65=1.1
 CORE USED:  20K  (39 PAGES)
```