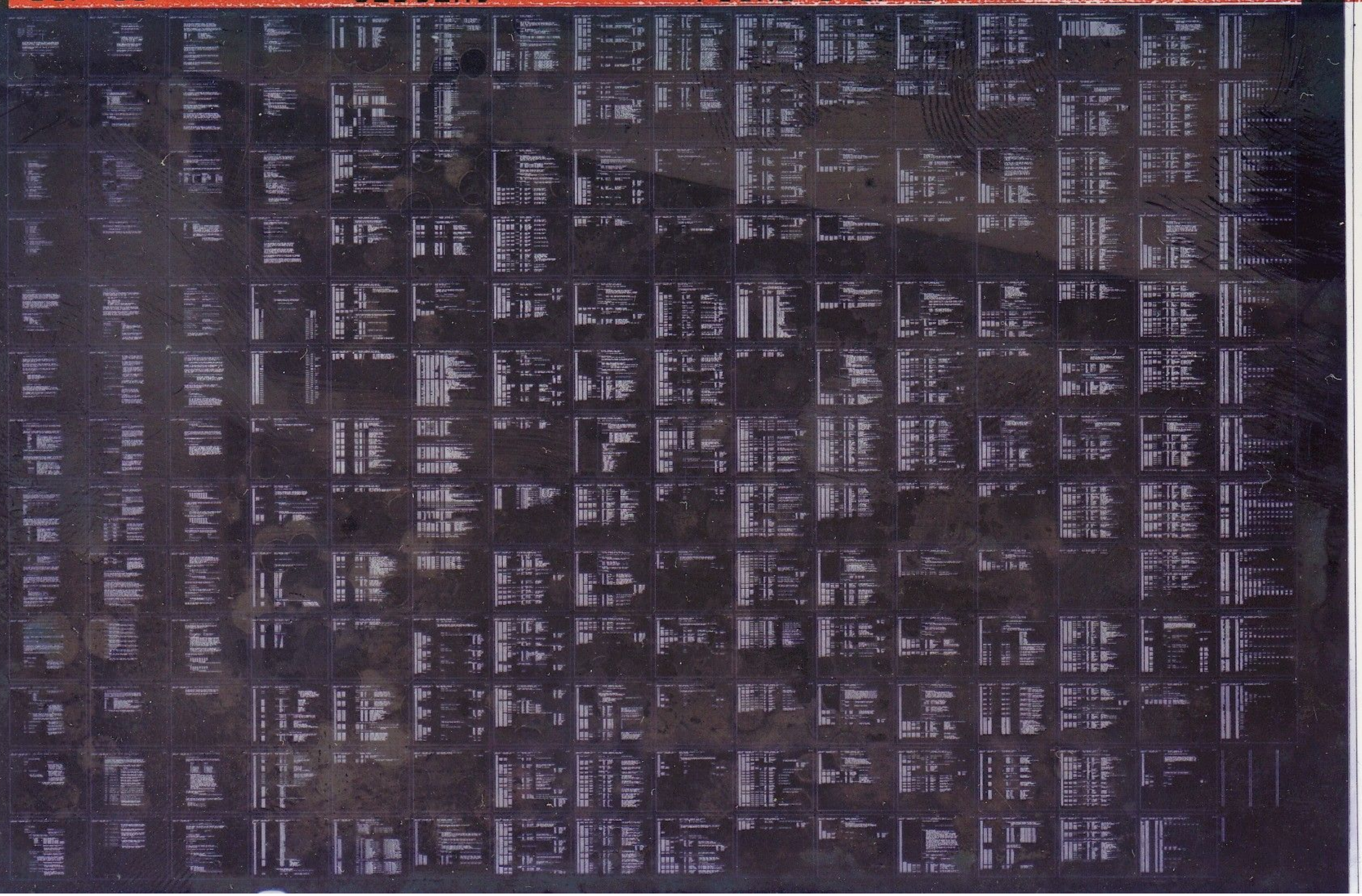
DUP-11

DCLT DUP
CZDCLAO

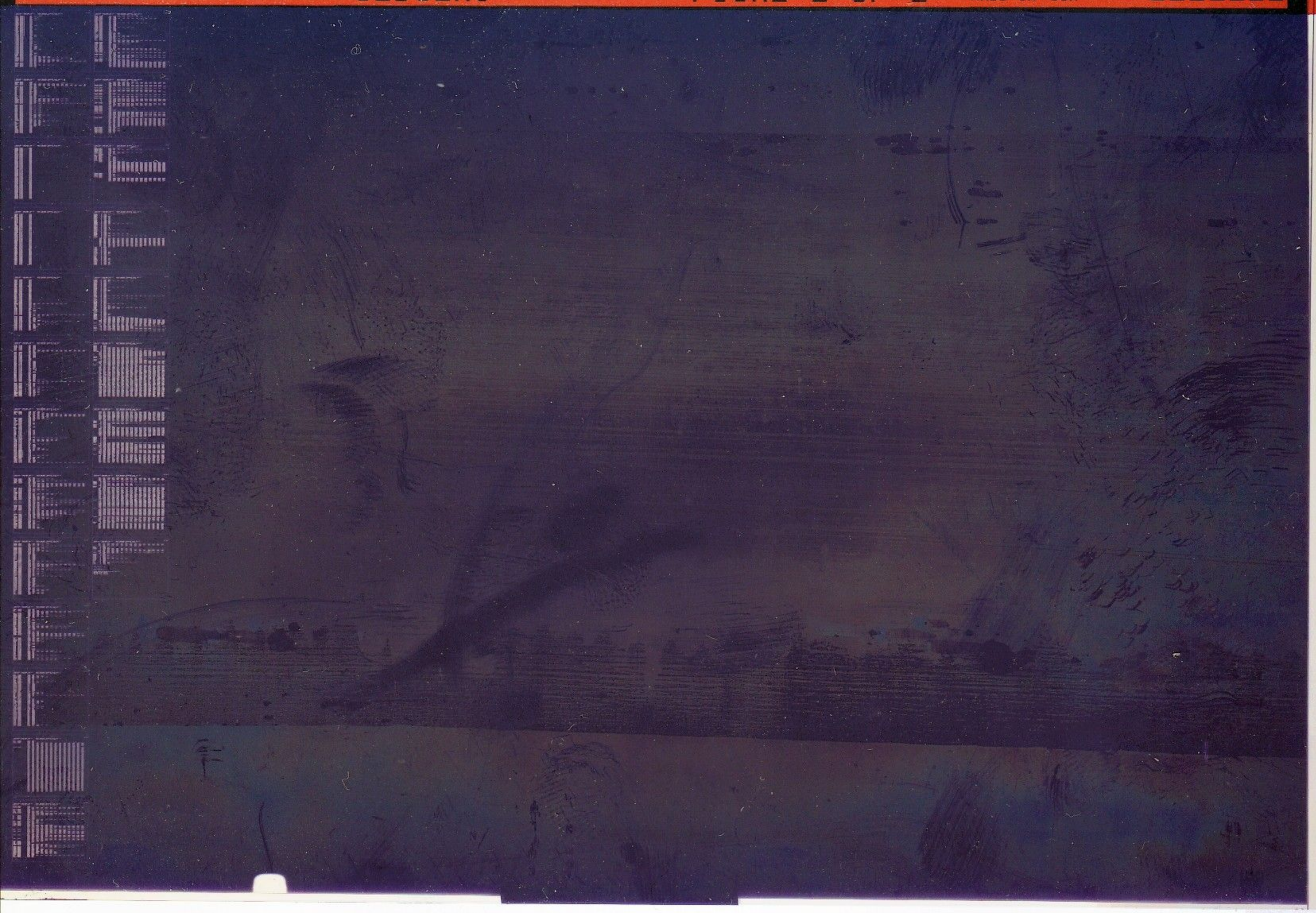
AH-S975A-MC
FICHE 1 OF 2



DUP·11

DCLT DUP
CZDCLAO

AH-S975A-MC
FICHE 2 OF 2



```
1                               .TITLE  CZDCLA DUP-11 DATA COMM. LINK TEST
                                .REM &
```

IDENTIFICATION

PRODUCT CODE: AC-5974A-MC

PRODUCT NAME: CZDCLA0 DUP-11 DATA COMM. LINK TEST

PRODUCT DATE: MARCH 1982

MAINTAINER: MERRIMACK DIAGNOSTIC ENGINEERING

AUTHOR: ERNIE COOPER

REVISION HISTORY:

REV	DATE	AUTHOR	REASON
A	24-MAR-82	ERNIE COOPER	ORIGINAL ISSUE, DCLT FOR THE DUP-11

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 DUP-11 COMMUNICATION LINKS. THIS PROGRAM ALLOWS THE DUP-11 TO COMMUNICATE WITH OTHER SYNCHRONOUS(INCLUDING DDCMP) DEVICES ON POINT TO POINT OR MULTIDROP NETWORKS. THIS DCLT PROGRAM WILL PROVIDE THE COVERAGE NECESSARY TO DETECT FAILURES TO THE COMPUTER EQUIPMENT, THE COMMUNICATION LINK, OR THE MODEM.

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

1.2 SYSTEM REQUIREMENTS

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

- A PDP-11 CPU
- MINIMUM OF 24K WORDS OF MEMORY
- A WORKING CLOCK
- A CONSOLE TERMINAL
- ANY XXDP+ SUPPORTED LOAD MEDIA
- DUP11-DA: M7867 MODULE
  BC05C-25 CABLE
  BC02-1D CABLE
  H325 TEST CONNECTOR

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, DUP-11 AND THE DEVICES AT THE OTHER END OF THE LINK HAVE ALREADY BEEN TESTED.

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

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

SOME OF THE DIAGNOSTICS THAT COULD BE RUN IF THE DUP-11 LOOKS BAD:

DZDPEXX CONFIDENCE TEST
DZDPBXX BASIC TRANSMITTER TESTS
DZDPCXX RECEIVER,MODEM CONTROL AND INTERRUPT TEST
DZDPDXX DATA AND FUNCTION TESTS
DXDPBXX DECX11 MODULE

XX= LATEST REVISION

1.5 ASSUMPTIONS - RESTRICTIONS

IT IS ASSUMED THAT THE 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.

THIS DIAGNOSTIC DOES NOT RUN THE DUP-11 IN BIT STUFF MODE. IT IS ASSUMED THAT IF THE LINK WORKS IN CHAR MODE THE LINK WILL WORK IN BIT STUFF MODE.

THE DUP-11 IS NOT A DMA DEVICE AND THUS MUST RELY ON THE SOFTWARE FOR SERVICE.

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

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

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

2.2 SWITCHES

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

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

EXAMPLE OF SWITCH USAGE:

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

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

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

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

2.3 FLAGS

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

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

APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)

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

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

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

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

2.4 HARDWARE QUESTIONS

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

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

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

UNIT 0
FULL DUPLEX OPERATION :  (L) Y ?
DEVICE CSR ADDRESS : (O)  160170 ?
INTERRUPT VECTOR ADDRESS:  (O)  300  ?
REMOTE NODE "ITEP" :  (L) N ?
IS THIS A MULTIPOINT NETWORK: (L) N ?
```

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

REMOTE NODE ITEP SHOULD BE ANSWERED "Y" IF OTHER NODE IS RUNNING SOFTWARE THAT IS USING "ITEP" FORMATS(I.E. PDP-11 RUNNING ITEP).

IF OTHER NODE IS USING ITEP, THE ABOVE "MULTIPOINT NETWORK" QUESTION WILL NOT APPEAR.

IF TO THE "MULTIPOINT NETWORK" QUESTION, YOU RESPOND WITH "Y"
THEN
        ADDRESS THIS STATION: (D) 1 ?

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

2.5 DATA COMM. LINK TEST COMMANDS

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

MESSAGE COMMANDS AVAILABLE:

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

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

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

2.5.1 MESSAGE COMMANDS

COMMAND	DESCRIPTION
CLEAR EXPECTLIST	ZEROES THE EXPECTLIST (00'S) AND THEN PUTS DEFAULT ITEP MSG IN SO NOT REALLY EMPTY
CLEAR TRANSMITLIST	FILLS TRANSMITLIST (000'S) AND THEN PUTS DEFAULT ITEP MSG IN SO NOT REALLY EMPTY

```
SET     EXPECTMSG=TYPE/QUAL             DEFINE A MESSAGE TO BE PUT ON
                                           THE EXPECTED LIST

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

        WHERE THE OPTIONAL "QUAL" IS:

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

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

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

SET     TRANSMITMSG=TYPE/QUAL           DEFINE A MESSAGE TO BE PUT ON
                                           THE TRANSMIT LIST
                                           (SEE DESCRIPT FOR SET EXP)

SET     EXPECT=TRANSMIT                 MAKES A COPY OF THE TRANSMIT
                                           LIST IN THE EXPECT LIST.

SHOW    EXPECTLIST                      LISTS THE MESSAGE SIZE AND TYPE
                                           FOR THE MESSAGES IN THE
                                           EXPECT LIST

SHOW    TRANSMITLIST                    LISTS THE MESSAGE SIZE AND TYPE
                                           FOR THE MESSAGES IN THE
                                           TRANSMIT LIST
```

2.5.2 STATISTICAL COMMANDS

COMMAND	DESCRIPTION
PRINT	TAKES THE OPERATOR TO THE REPORT LEVEL 'RPT>'. FROM HERE YOU CAN EXAMINE THE EVENT LOG OR IF "/PROTOCOL" IS SELECTED, THE DDCMP STATISTICAL AND ERROR COUNTERS.
DUMP SSSSSS-EEEEEE/B WHERE "/B" IS OPTIONAL: DEFAULT IS PRINT WORDS "/B" CAUSES PRINT BYTES	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. IS PRINTED IN WORD FORMAT.

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

2.5.3 RUN COMMAND

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

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

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

=ACTIVE	(FORCES /NOECHO ,NO LOOPING)
=PASSIVE	(FORCES NO LOOPING)
=RECEIVE	(FORCES /NOECHO ,NO LOOPING)
=LISTEN	(FORCES /NOECHO ,NO LOOPING, /NOCHECK)
=TRANSMIT	(FORCES /NOECHO ,NO LOOPING, /NOCHECK)
=TALK	(FORCES /NOECHO ,NO LOOPING, /NOCHECK)
=DOWNLINELOAD	(DOWN-LINE-LOADING IS NOT SUPPORTED FOR DUP-11 TO DUP-11 LINKS).
	(FORCING NO LOOPING MEANS IT MUST BE SPECIFIED AS A QUALIFIER ANY TIME ITS DESIRED, THERE IS NO DEFAULT)

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

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

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

/STATUS/NOSTATUS	ENABLES/DISABLES PRINTING OF PROGRAM STATUS MESSAGES TO THE OPERATOR
/ECHO/NOECHO	ENABLES/DISABLES THE RETRANSMISSION OF THE DATA RECEIVED IN PASSIVE MODE. (IGNORED IN MODES OTHER THAN PASSIVE)
/MODEM/NOMODEM/	ENABLES/DISABLES THE REPORTING OF MODEM STATUS INTERRUPT CHANGES.
/LOOP=LTYPE	SPECIFIES WHICH, IF ANY, TYPE OF MAINTENANCE LOOPBACK IS BEING USED. (IGNORED IN MODES OTHER THAN ACTIVE) MUST BE SPECIFIED EACH TIME ELSE NO LOOP IS USED.

''LTYPE'' IS:

=INTERNALTTL LOOPS DATA INTERNAL TO USYNRT

=CABLE USE THIS FOR TESTING WITH H325
TURNARROUND CONNECTOR ON END OF CABLE.

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

=LOCALMODEM NOT SUPPORTED BY DUP-11
=REMOTEMODEM '' ''

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

/PROTOCOL ENABLES SUBSET OF DDCMP PROTOCOL- THE DUP-11
CAN NOW COMMUNICATE WITH OTHER 'INTELLIGENT'
SYNCHRONOUS DEVICES THAT SUPPORT DDCMP IN
THEIR MICROCODE. (DMR,DMC,DMV OR DMP).

THIS SWITCH IS NOT SUPPORTED BY ALL DCLT'S.

/NOPROTOCOL DISABLES PROTOCOL- THE DUP-11 CAN COMMUNICATE
ONLY WITH ANOTHER DUP-11 OR DPV-11 RUNNING
DCLT OR ITEP.

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

2.5.4 DEFAULTS

IF NO "SET'S" THEN THE DEFAULT IS SAME AS IF TYPED:

```
SET TRANSMITMSG=ITEP/SIZE=58/COPY=0
SET EXPECTMSG=ITEP/SIZE=58/COPY=0
```

THE DEFAULT COPY AND SIZE FOR EACH OF THE MESSAGE TYPES:

```
ONES - /SIZE=64/COPY=0
ZEROES - /SIZE=64/COPY=0
0ALT - /SIZE=64/COPY=0
1ALT - /SIZE=64/COPY=0
CCITT - /SIZE=64/COPY=0
ALPHA - /SIZE=65/COPY=0
ITEP - /SIZE=58/COPY=0
OPER. SPEC'D - /SIZE=LENGTH-OF-TEXT-TYPED-BETWEEN-QUOTES/COPY=0
```

FOR THE RUN COMMAND THE DEFAULTS ARE:

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

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

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

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

OTHER NOTES..

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

2.5.5 PRINT COMMAND

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

COMMAND	DESCRIPTION
HELP OR ?	PRINTS HELP INFORMATION FOR RPT>
EXIT	RETURNS YOU TO THE LEVEL THAT YOU ENTERED FROM. (DCLT> OR DR>)
LOG	PRINTS THE DCLT EVENT LOG
COUNTERS/FULL	PRINTS THE ENTIRE DDCMP STATISTICAL AND ERROR LOG. SEE SECTION 4.3
COUNTERS/ERRORS	PRINTS ONLY THE DDCMP ERROR LOCATIONS OF THE LOG.
COUNTERS/OFFSET=NN	PRINTS A SINGLE LOCATION OF THE LOG AS SPECIFIED BY THE OCTAL WORD OFFSET VALUE(NN). NOTE:: THE DDCMP COUNTERS WILL BE VALID ONLY WITH PROTOCOL ENABLED(/PROTOCOL).

2.5.6 MISC COMMANDS

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

2.6 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

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

6. ANSWER ALL THE HARDWARE QUESTIONS. THE NUMBER OF UNITS THAT DCLT CAN USE IS ALWAYS "1".

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

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

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

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

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

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

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

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

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

3.2 SPECIFIC ERROR MESSAGES

3.2.1 COMMAND LINE INTERPRETER ERRORS

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

?''LOOP'' VALID ONLY IN ACTIVE?	THE ''/LOOP=..'' SWITCH WAS TYPED IN A RUN COMMAND BUT THE MODE WAS NOT SET TO ACTIVE. MAINTENANCE LOOP IS ONLY POSSIBLE IF THE MODE OF OPERATION IS ACTIVE.
?''ECHO'' VALID ONLY IN PASSIVE?	THE ''/ECHO'' SWITCH WAS TYPED IN A RUN COMMAND BUT THE MODE WAS NOT SET TO PASSIVE. ECHOING OF RECEIVED DATA IS ONLY POSSIBLE IF THE MODE OF OPERATION IS PASSIVE.
?ILL CHR- ''A-Z,0-9,SP,TAB'' ONLY?	A CHARACTER TYPED WITHIN QUOTES WHEN TRYING TO DEFINE THE CONTENTS OF A TRANSMIT OR EXPECT MESSAGE WAS NOT A ''A-Z,0-9,SPACE OR TAB''. RETYPE THE COMMAND WITH ONLY THESE CHARACTERS BETWEEN QUOTES.
?''SIZE=0'' NOT VALID?	A MESSAGE ZERO BYTES LONG CAN NOT BE BUILT. RETYPE THE COMMAND WITH A ''/SIZE=NNN''. IF NO ''/SIZE='' IS TYPED A DEFAULT SIZE WILL BE USED.
?TRANSMIT AND EXPECT LIST MUST BE IDENTICAL FOR LOOP?	IF RUN COMMAND WITH ''/LOOP/CH'' IS TYPED THE TRANSMIT LIST AND EXPECT LIST MUST BE EQUAL. IF THEY ARE NOT THIS ERROR WILL BE DISPLAYED. USE 'SE E=T' COMMAND.

3.2.2 DCLT OR DEVICE ERROR MESSAGES:

CLOCK NOT FOUND	THIS MEANS THAT NO CLOCK WAS FOUND ON THE SYSTEM THE DIAGNOSTIC WILL STILL RUN BUT NONE OF THE TIME OUT CONDITIONS WILL OCCUR
BAD CLOCK - PROGRAM WILL HANG ON ''TIMEOUT''!!	THIS MEANS THAT THE CLOCK FOUND ON THE SYSTEM DID NOT INTERRUPT WHEN ASKED TO DO A ''TICK''. THE PROGRAM WILL STILL RUN, BUT ANY OF THE PROGRAM THAT TIMES THE DEVICE WILL HANG IF THE DEVICE TIMES OUT. ALSO, THE EVENT LOG WILL CONTAIN A ZERO EVENT TIME FOR ALL EVENTS LOGGED.

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

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

BUFFER FULL - MSG. NOT BUILT !!

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

CHAR. COUNT EXCEEDS BUFF LIMIT - MSG TRUNCATED

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

3.2.3 DEVICE ERROR MESSAGE

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

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

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

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

DATA COMPARISON DATA ERROR
TOTAL MISMATCHES IN MSG = NNN

THIS MEANS THAT WHEN THE MESSAGE

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

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

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

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

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

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

MASTER RESET DID NOT WORK
RXCSR TXCSR
XXXXXX XXXXXXX
THIS MEANS THAT AFTER A MASTER
RESET WAS ISSUED TO DUP THE
RXCSR REGISTER WAS NON ZERO.

NO CLEAR TO SEND FROM MODEM
RXCSR TXCSR
XXXXXXX XXXXXXX
WHEN REQUEST TO SEND (RTS)
IS SET, MODEM DOES NOT RESPOND
WITH CLEAR TO SEND(CTS).

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

MODEM DID NOT RETURN MODEM READY
RXCSR TXCSR
XXXXXXX XXXXXXX
WHEN THE DTR SIGNAL WAS SET,
DATA SET READY WAS NOT RETURNED

CRC IN ERROR
RXDBUF RXCSR
XXXXXX XXXXXX
A CRC ERROR WAS DETECTED BY
THE DUP ON AN INCOMING MESSAGE.

RECEIVER OVERRUN RXDBUF RXCSR XXXXXXX XXXXXXX	THE RECEIVER WASN'T SERVICED FAST ENOUGHT (SOFTWARE)-- CAUSING A CHARACTER TO BE LOST.
TIMED OUT IN START,STACK ACK SEQ RDATA SDATA XXXXXXX XXXXXXX	THIS USUALLY MEANS THAT THE DUP IS UNABLE TO ESTABLISH A CONNECTION WITH THE OTHER DEVICE BEING TESTED.THE VALUES IN RDATA AND SDATA SHOW THE RECIEVED(RDATA) AND TRANSMITTED (SDATA). SEE DDCMP SPEC. FOR FURTHER EXPLANATION OF STARTUP SEQUENCE.

4.0 PERFORMANCE AND PROGRESS REPORTS

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

4.1 PRINTING OF EVENT LOG

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

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

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

THE TYPES OF EVENTS KEPT IN THE EVENT LOG ARE:

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

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

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

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

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

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

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

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

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

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

4.2 OPERATOR STATUS MESSAGES

THE "/STATUS, /NOSTATUS" 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

NOTE:: BECAUSE THE DUP IS AN INTERRUPT DRIVEN DEVICE, IT IS BEST TO DISABLE STATUS TO PREVENT OVERRUN ERRORS.

4.3 PRINTING DDCMP STATISTICAL AND ERROR LOG

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

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

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

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

5.0 DEVICE INFORMATION TABLES

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

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

```
.WORD   1               ;[0] FULL OR HALF DUPLEX FLAG (BIT0=1 IF FULL)
.WORD   160170          ;[2] CSR ADDRESS
.WORD   300             ;[4] INTERRUPT VECTOR
.WORD   240             ;[6] INTERRUPT PRIORITY
.WORD   0               ;[10] PT-PT=0   MULTIPOINT=1
.WORD   1               ;[12] TRIB ADDRESS THIS STATION
.WORD   0               ;[14] REMOTE NODE "ITEP"
```

6.0 MODE AND MESSAGE DESCRIPTIONS

6.1 MODE DESCRIPTIONS

THE FOLLOWING MODE DESCRIPTIONS REFER TO MESSAGE LISTS BEING
TRANSMITTED AND RECIEVED BUT BE AWARE THAT OTHER DATA IS ALSO
SENT AND RECEIVED. IF "/PROTOCOL" IS SELECTED THE DATA IS
ENCLOSED IN A DDCMP ENVELOPE AND CONTROL MESSAGE WILL ALSO
APPEAR ON THE LINK.

6.1.1 TRANSMIT MODE

A LIST OF MESSAGES IS TRANSMITTED WITHOUT EXPECTING ANY DATA
TO BE RECEIVED. HOWEVER WITH "/PROTOCOL" ENABLED EACH MESSAGE
SENT MUST BE ACKNOWLEDGED(ACK).

6.1.2 RECEIVE MODE

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

6.1.3 PASSIVE MODE

THEN EVERY TIME A MESSAGE IS RECEIVED, A MESSAGE IS TRANSMITTED.
DATA CHECKING CAN BE DONE ON THE RECEIVED DATA.THE "/ECHO, /NOECHO"

ENABLES/DISABLES THE RETRANSMISSION OF THE DATA RECEIVED.

6.1.4 ACTIVE MODE

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

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

6.1.5 DOWN-LINE-LOAD

DOWN-LINE-LOADING IS NOT SUPPORTED IN THE SOFTWARE FOR DUP-11
TO DUP-11 LINKS. HOWEVER IT IS POSSIBLE TO "REQUEST SECONDARY
LOAD" FROM A HOST STATION(IF SUPPORTED) IF THERE IS A DUP(M9312)
DECNET BOOTSTRAP MODULE IN YOUR MACHINE.
SEE BOOTSTRAP OPERATOR'S MANUAL.

6.1.6 TALK MODE

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

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

6.1.7 LISTEN MODE

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

6.1.8 MAINTENANCE "LOOP" MODES

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

INTERNALTTL LOOPS DATA INTERNAL TO THE USYNRT

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

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

*= MOST LIKELY TO BE IN THAT MODE

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

IF PROTOCOL IS ENABLED, THE H/D START COLUMN CAN BE IGNORED.

6.2 MESSAGE DESCRIPTIONS

NAME	DESCRIPTION
ZEROES	MESSAGE OF ALL 0'S (00000000,00000000,00000000,....)
ONES	MESSAGE OF ALL 1'S (11111111,11111111,11111111,....)
1ALT	MESSAGE OF ALTERNATING 1'S (10101010,10101010,....)
0ALT	MESSAGE OF ALTERNATING 0'S (01010101,01010101,....)
CCITT	"CCITT" 512-BIT (VS. 511 BITS) TEST PATTERN
ITEP	"INTERPROCESSOR TEST PROGRAM'S (ITEP)" MESSAGE 1(DP1:) (<177><177>/$A THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.<15><12><001><177><177><177><177>)
ALPHA	ALPHA-NUMERICS (OR FUTURE COMM TURNAROUND MSG) (#$!" (AMPERSAND)'()*+,-.0123456789:;<=>?@ABCDEFGHIJK LMNOPQRSTUVWXYZ/[\]^_%)
OPERATOR-SPECIFIED	"A-Z,0-9,SPACES,TABS" THESE ARE THAT THE CHARACTERS THAT CAN BE TYPED BETWEEN QUOTATION MARKS ("..") TO SPECIFIY A UNIQUE MESSAGE.

7.0 OTHER INFORMATION

7.1 INTERFACING TO AN "ITEP" NODE

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

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

NOTE: DO NOT USE SWITCH 8 WITH ITEP GOING TO DCLT
THE ONLY MESG. DCLT SUPPORTS IS MSG 1.
DCLT IGNORES CRC ERRORS WHEN REC DATA FROM ITEP
BECAUSE ITPE SENDS NO CRC.

7.2 TROUBLESHOOTING HINTS

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

HINT::: IF THIS DOCUMENT IS TOO LARGE TO CONSUME, GET A COPY OF DEC'S
COMMUNICATION OPTIONS MINI REFERENCE GUIDE(EK-CM1N1-RM-001).

REMEMBER, IF YOU ARE HAVING TROUBLE WITH RX OVERRUN ERRORS OR MISSED
MESSAGES, OR DATA CHECK ERRORS-- DISABLE STATUS(/NOSTATUS). THE CPU
IS HAVING A HARD TIME SERVICING BOTH THE TTY AND DUP.

EVEN IF YOU ARE CHECKING OUT DUP-11 TO DUP-11 LINKS, IT IS A GOOD IDEA
TO ENABLE PROTOCOL(/PROTOCOL). BY EXAMINING THE DDCMP STATISTICAL AND
ERROR LOG, YOU WILL GET A COMPLETE PICTURE OF WHAT IS HAPPENING ON THE
LINK. NOISY COMM LINKS WILL BE DETECTED BETTER IF LARGE MESSAGES(512
CHARS) ARE SENT.

BOTH NODES MUST EITHER ENABLE("/PROTOCOL") OR DISABLE("/NOPROTOCOL").

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/NPR
```

WHAT THE ABOVE COMMAND SEQUENCE MEANS:

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

WHAT YOU SHOULD SEE AFTER ENTERING THE RUN COMMAND

IF THINGS ARE RUNNING CORRECTLY :

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

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

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

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

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. ON NON-DDCMP LINKS, THE "/PROTOCOL" SWITCH IS OPTIONAL.

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

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

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 THE
"/LOOP=CABLE" SWITCH.

POSSIBLE COMMAND SEQUENCE IS:

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

WHAT THIS SEQUENCE MEANS:

THIS SEQUENCE HAS THE "/LO=C". THIS INFORMS THE SOFTWARE
NOT TO CHECK FOR DATA SET READY SIGNAL FROM THE MODEM.
ALSO A CLOCK SIGNAL IS FURNISHED BY THE DUP.

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

```
INI RXQ TXQ TXC RXQ TXQ TXC RXQ
TXQ TXC CMP CMP CMP EOP RXQ TXQ
TXC RXQ TXQ TXC RXQ TXQ TXC CMP
CMP CMP EOP RXQ TXQ TXC RXQ TXQ
TXC RXQ TXQ TXC CMP CMP CMP EOP
     MODE=ACTIVE/LOOP=CABLE/PASS=00000
                /STATUS/CHECK/NOECHO/NOMODEM/PROTOCOL
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
R M=TR/NOCH/PAS=3/NPR	R M=R/NOCH/PAS=3/NPR

WHAT THIS SEQUENCE MEANS:

THE "C E " AND "C T" INITIALIZE BOTH THE
TRANSMIT AND EXPECT LISTS. 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 RECEIVE WITH NO DATACHECKING AND
THE PASS COUNT IS SET TO THREE.
PROTOCOL CAN BE OPTIONAL BUT IT MUST BE ENABLE OR
DISABLE ON BOTH ENDS.

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

FOR NODE A:

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

FOR NODE B:

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

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

R M=TR/PAS=3 R M=R/CH/PAS=3

WHAT THIS SEQUENCE MEANS:

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/PROTCOL
DCLT> (A)?
```

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

7.2.3 ONE NODE ACTIVE THE OTHER NODE PASSIVE

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

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

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

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

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

FOR NODE A:

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

FOR NODE B:

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

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

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

7.2.4 BOTH NODES ACTIVE

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

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

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

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

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

A GOOD VARIATION THAT CAN BE USED IS TO LOAD THE TRANSMIT LIST AND
EXPECT LIST WITH A LARGE MESSAGE(512 CHARACTERS),ENABLE PROTOCOL,
AND RUN MANY PASSSES ON BOTH ENDS.

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

7.2.5 TALK AND LISTEN MODES FOR COMMUNICATING

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

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

7.3 EXAMPLES OF COMMANDS

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

7.3.1 EXAMPLES OF MESSAGES COMMANDS

THE CLEAR COMMANDS .

```
C E
C T
```

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

IF THESE COMMANDS ARE FOLLOWED BY A SHOW COMMAND

```
SH E
```

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

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

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

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

AND FOLLOWED IT WITH A SHOW EXPECT LIST COMMAND

```
SH E
```

WHAT YOU WOULD SEE IS

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

```
                7.3.2 EXAMPLES STATISTICAL COMMANDS

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

                DCLT> (A)  ?

                THE SAME WILL HAPPEN IF YOU USE THE ?

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

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

7.3.3 EXAMPLES RUN COMMMANDS

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

```
IF YOU WERE TO EXECUTE THE RUN COMMAND
        R M=TR/NOST/CH/PAS=4
WHAT WOULD HAPPEN IS AFTER 4 PASSES THE PROGRAM WOULD RETURN
TO THE DCLT PROMPT AND PRINT
        MODE=TRANSMIT/PASS=00000
                    /NOSTATUS/CHECK/NOECHO/NOMODEM/NOPRTOCOL
  DCLT> (A)  ?
IF YOU WERE TO EXECUTE THE RUN COMMAND
        C E
        C T
        R M=A/LO=I/ST/CH/PAS=3/PROTOCOL
WHAT YOU WOULD SEE (IF USING DEFAULT TRANSMIT AND EXPECT
MESSAGES) IS
        INI RXQ TXQ TXC CMP EOP RXQ TXQ
        TXC CMP EOP RXQ TXQ TXC CMP EOP
        MODE=ACTIVE/LOOP=INTERNAL/PASS=0000
                    /STATUS/CHECK/NOECHO/NOMODEM/PROTOCOL
  DCLT> (A)  ?

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

        DR>
```

7.3.4 EXAMPLES PRINT COMMANDS

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

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

HERE ARE SOME EXAMPLES OF RPT> LEVEL COMMANDS:

THE HELP OR ? COMMAND

```
        HELP
```

OR

```
        ?
```

PRODUCES THE FOLLOWING:

```
DCLT REPORT CMDS:
LOG - PRINT DCLT EVENT LOG
EXIT - EXIT REPORT LEVEL
HELP - PRINT THIS MESSAGE
COUNTERS/SW - PRINT DDCMP COUNTERS
WHERE /SW=FULL, /ERRORS, /OFFSET=NN(O)

RPT> (A)  ? LOG

THE LOG COMMAND PRODUCES THE FOLLOWING

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

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

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

7.3.5 EXAMPLE EXIT COMMAND

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

```
EXIT
```

WILL RETURN YOU TO THE SUPERVISOR.

```
DR>
```

IF YOU ENTERED REPORT FROM THE DCLT LEVEL
THEN TYPING

```
EXIT
```

WILL RETURN YOU TO THE DCLT LEVEL.

```
DCLT>
```

7.4 THINGS TO WATCH OUT FOR

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

IF YOU ARE USING PASSIVE MODE WITH THE ECHO SWITCH
THEN YOU WILL PROBABLY HAVE TO RE-ENTER THE TRANSMIT
LIST ON THE SIDE WITH THE ECHO SWITCH. THE REASON IS
THAT THE TRANSMIT LIST GETS OVER WRITTEN WITH THE
RECEIVE LIST WHEN USING THE ECHO SWITCH. ALSO DISABLE
DATACHECKING("/NOCHECK").

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

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

```
                                .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  CZDCL,A,0,1800.,0,#PRI07
002000                                                          LSNAME::
002000  103                                                             .ASCII /C/
002001  132                                                             .ASCII /Z/
002002  104                                                             .ASCII /D/
002003  103                                                             .ASCII /C/
002004  114                                                             .ASCII /L/
002005  000                                                             .BYTE   0
002006  000                                                             .BYTE   0
002007  000                                                             .BYTE   0
002010                                                          L$REV::
002010  101                                                             .ASCII  /A/
002011                                                          L$DEPO::
002011  060                                                             .ASCII  /0/
002012                                                          L$UNIT::
002012  000000                                                          .WORD   0
002014                                                          L$TIML::
002014  003410                                                          .WORD   1800.
002016                                                          L$HPCP::
002016  046230                                                          .WORD   L$HARD
002020                                                          L$SPCP::
002020  000000                                                          .WORD   0
002022                                                          L$HPTP::
002022  002130                                                          .WORD   L$HW
002024                                                          L$SPTP::
002024  000000                                                          .WORD   0
002026                                                          L$LADP::
002026  046616                                                          .WORD   L$LAST
002030                                                          L$STA::
002030  000000                                                          .WORD   0
002032                                                          L$CO::
002032  000000                                                          .WORD   0
002034                                                          L$DTYP::
```

```
2010  002034  000000                          .WORD   0
2011  002036                          L$APT::
2012  002036  000000                          .WORD   0
2013  002040                          L$DTP::
2014  002040  002124                          .WORD   L$DISPATCH
2015  002042                          L$PRIO::
2016  002042  000340                          .WORD   #PRIO7
2017  002044                          L$ENVI::
2018  002044  000000                          .WORD   0
2019  002046                          L$EXP1::
2020  002046  000000                          .WORD   0
2021  002050                          L$MREV::
2022  002050     003                          .BYTE   C$REVISION
2023  002051     003                          .BYTE   C$EDIT
2024  002052                          L$EF::
2025  002052  000000                          .WORD   0
2026  002054  000000                          .WORD   0
2027  002056                          L$SPC::
2028  002056  000000                          .WORD   0
2029  002060                          L$DEVP::
2030  002060  011524                          .WORD   L$DVTYP
2031  002062                          L$REPP::
2032  002062  025336                          .WORD   L$RPT
2033  002064                          L$EXP4::
2034  002064  000000                          .WORD   0
2035  002066                          L$EXP5::
2036  002066  000000                          .WORD   0
2037  002070                          L$AUT::
2038  002070  026334                          .WORD   L$AU
2039  002072                          L$DUT::
2040  002072  026326                          .WORD   L$DU
2041  002074                          L$LUN::
2042  002074  000000                          .WORD   0
2043  002076                          L$DESP::
2044  002076  011534                          .WORD   L$DESC
2045  002100                          L$LOAD::
2046  002100  104035                          EMT     E$LOAD
2047  002102                          L$ETP::
2048  002102  000000                          .WORD   0
2049  002104                          L$ICP::
2050  002104  025352                          .WORD   L$INIT
2051  002106                          L$CCP::
2052  002106  026240                          .WORD   L$CLEAN
2053  002110                          L$ACP::
2054  002110  026236                          .WORD   L$AUTO
2055  002112                          L$PRT::
2056  002112  025344                          .WORD   L$PROT
2057  002114                          L$TEST::
2058  002114  000000                          .WORD   0
2059  002116                          L$DLY::
2060  002116  000000                          .WORD   0
2061  002120                          L$HIME::
2062  002120  000000                          .WORD   0
2063
```

```
                                .SBTTL  DISPATCH TABLE

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

                                        DISPATCH 1
002122                                                                  .WORD   1
002122  000001                                                  L$DISPATCH::
002124                                                                  .WORD   T1
002124  026342
```

```
                                .SBTTL  DEFAULT HARDWARE P-TABLE

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

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


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

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


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

002132  160170                          .WORD   160170         ;[2] CSR ADDRESS
002134  000300                          .WORD   300            ;[4] INTERRUPT VECTOR
002136  000240                          .WORD   240            ;[6] INTERRUPT PRIORITY (5)
002140  000000                          .WORD   0              ;[10] MULTI POINT =1   PT TO PT = 0
002142  000001                          .WORD   1              ;[12] TRIB ADDRESS THIS STATION
002144  000000                          .WORD   0              ;[14] OTHER NODE "ITEP"
002146  000000                          .WORD   0              ;[16] SPARE


002150                                  ENDHW
002150                                                          L10000:
```

```
2115
2116
2117
2118
2119                            .SBTTL  GLOBAL EQUATES SECTION
2120
2121
2122
2123
2124                            ;++
2125                            ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
2126                            ; ARE USED IN MORE THAN ONE TEST.
2127                            ;--
2128
2129
2130  002150                            EQUALS
2131                            ;
2132                            ; BIT DIFINITIONS
2133                            ;
2134            100000          BIT15== 100000
2135            040000          BIT14== 40000
2136            020000          BIT13== 20000
2137            010000          BIT12== 10000
2138            004000          BIT11== 4000
2139            002000          BIT10== 2000
2140            001000          BIT09== 1000
2141            000400          BIT08== 400
2142            000200          BIT07== 200
2143            000100          BIT06== 100
2144            000040          BIT05== 40
2145            000020          BIT04== 20
2146            000010          BIT03== 10
2147            000004          BIT02== 4
2148            000002          BIT01== 2
2149            000001          BIT00== 1
2150                            ;
2151            001000          BIT9==  BIT09
2152            000400          BIT8==  BIT08
2153            000200          BIT7==  BIT07
2154            000100          BIT6==  BIT06
2155            000040          BIT5==  BIT05
2156            000020          BIT4==  BIT04
2157            000010          BIT3==  BIT03
2158            000004          BIT2==  BIT02
2159            000002          BIT1==  BIT01
2160            000001          BIT0==  BIT00
2161                            ;
2162                            ; EVENT FLAG DEFINITIONS
2163                            ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
2164                            ;
2165            000040          EF.START==      32.                     ; START COMMAND WAS ISSUED
2166            000037          EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
2167            000036          EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
2168            000035          EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
2169            000034          EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
2170                            ;
```

```
2171                            ;
2172                            ; PRIORITY LEVEL DEFINITIONS
2173                            ;
2174            000340          PRI07== 340
2175            000300          PRI06== 300
2176            000240          PRI05== 240
2177            000200          PRI04== 200
2178            000140          PRI03== 140
2179            000100          PRI02== 100
2180            000040          PRI01== 40
2181            000000          PRI00== 0
2182                            ;
2183                            ;OPERATOR FLAG BITS
2184                            ;
2185            000004          EVL==       4
2186            000010          LOT==      10
2187            000020          ADR==      20
2188            000040          IDU==      40
2189            000100          ISR==     100
2190            000200          UAM==     200
2191            000400          BOE==     400
2192            001000          PNT==    1000
2193            002000          PRI==    2000
2194            004000          IXE==    4000
2195            010000          IBE==   10000
2196            020000          IER==   20000
2197            040000          LOE==   40000
2198            100000          HOE==  100000
2199
```

```
                        ;***** INDEPENDENT EQUATES
                        
001000                          BUFLIM=512.             ;MAX BUFFER SIZE IN BYTES
                                                        ;  APPLIES TO TX,RX AND CMP BUFFS
000017                          MSGLIM=15.              ;MAX NO. OF MESSAGES PER BUFFER
                                                        ;   (FOR EACH INCREMENT (+1) TO MSGLIM,
                                                        ;    ADD 6 WORDS TO THE POINTER TABLE
                                                        ;    (PTRTAB:) SINCE THIS MEANS 2 MORE
                                                        ;    "POINTER" WORDS PER BUFFER.
                        
                        ;MODE OF OPERATION EQUATES
000000                          REC=0                   ;RECEIVE MODE
000001                          TRA=1                   ;TRANSMIT MODE
000002                          PAS=2                   ;PASSIVE MODE
000003                          ACT=3                   ;ACTIVE MODE
000004                          DOW=4                   ;DOWN-LINE-LOAD MODE
000005                          TAL=5                   ;TALK MODE
000006                          LIS=6                   ;LISTEN MODE
                        ;MAINT LOOP TYPE EQUATES
000000                          NONE=   0               ;NO LOOP
000001                          TTL=    1               ;INTERNAL TTL
000002                          CABLE=  2               ;CABLE LOOP
000003                          MODLOC= 3               ;MODEM LOCAL
000004                          MODREM= 4               ;MODEM REMOTE
000005                          MOP=    5               ;MOP
                        
                        
                        ;CLOCK ENABLE VALUES TO BE LOADED IN CLK'S CSR
000100                          LCLKEN= 100             ;L-CLOCK CSR VALUE TO ENABLE THE CLOCK
000111                          PCLKEN= 111             ;P-CLOCK CSR VALUE TO ENABLE THE CLOCK
001600                          PCLKCT= 1600            ;P-CLOCK COUNT SET REGISTER FOR COUNTER
                        
                        ;PARAM WORD EQUATES
                        
000001                          STATB=  BIT0            ;OPERATOR AWAKE ASKED FOR
000002                          DATCKB= BIT1            ;DATA CHECK BIT
000004                          ECHOB=  BIT2            ;ECHO BIT
000010                          MOCHK=  BIT3            ;MODEM STATUS CHECK BIT
000020                          CRCB=   BIT4            ;CRC CALCUALTE ASKED FOR
000040                          PROTOB= BIT5            ;PROTOCOL PROCESSING ASKED FOR
000100                          PRORUN= BIT6            ;PROTOCOL IS RUNNING(NOT STARTING OR MAINT)
000200                          ABORT=  BIT7            ;FATAL PROTOCOL ERROR(SET IN PROTOCOL CODE)
                        
                        ;OPTION TYPE EQUATES
                        
                        
000000                          DPV=    0       ;CODE FOR DPV CHAR MODE
                        
                        ;EVENT LOG MESSAGE TYPES (USED TO LOCATE EVENT DESCRIPTION IN EVENT TABLE
                        ;  AND DISPATCHING TO SEPERATE SECTIONS OF THE EVENT REPORTING SECTION)
000000                          TXQ=    0               ;TRANSMIT MESSAGE QUEUED
000002                          TXC=    2               ;TRANSMIT COMPLETE
000004                          RXQ=    4               ;RECEIVE BUFFER QUEUED
000006                          RXC=    6               ;RECEIVE COMPLETE
000010                          DER=    10              ;DEVICE INFORMATION
000012                          DVI=    12              ;DEVICE ABOUT TO INIT
```

```
2256    000014                  DCK=    14              ;DATA COMPARISON RESULTS
2257
2258    000016                  MSC=    16              ;MODEM STATUS CHANGE
2259
2260    000020                  DLE=    20              ;DATA COMPARISON LENGH ERROR
2261    000022                  DDE=    22              ;DATA COMPARISON DATA ERROR
2262    000024                  EOP=    24              ;END OF PASS
2263    000026                  ABO=    26              ;^C ABORT
2264
2265                    ;EQUATES FOR FLAG WORD
2266
2267    000001                  ININT=  BIT0            ;INPUT INT. REC.
2268    000002                  OTINT=  BIT1            ;OUTPUT INT REC
2269    000004                  QRX=    BIT2            ;RX QUED /COMPL
2270    000010                  QTX=    BIT3            ;TX QUED/COMPL
2271    000100                  ERX=    BIT6            ;EXPECT TO GET A RX COMPLED
2272    000200                  ETX=    BIT7            ;EXPECT TO GET A TX COMPLETED
2273
2274
2275    000020                  TXM=    BIT4            ;INDICATES TO TX INTERRUPT ROUTINE
2276                                                    ;THAT IT IS TIME TO TRANSMIT BODY OF MSG.
2277    000040                  RXM=    BIT5            ;INDICATES TO RX INTERUPPT ROUTINE
2278                                                    ;THAT IT IS TIME TO REC MSG BODY
2279    000400                  BCC=    BIT8            ;TIME FOR CRC CHECK.
2280
2281    001000                  PAD=    BIT9            ;INDICATES THAT PAD MUST BE SENT
2282
2283    002000                  INOVR=  BIT10           ;INIT OVER
2284
2285    004000                  FIRST=  BIT11           ;FIRST  TIME FOR CTS
2286
2287                    ; SPECIAL CLI CODES FOR "CHAR" ARGUMENT IN CLI CALLS
2288                    ;  (COMMAND LINE INTERPRETER DEFINITIONS)
2289    000000                  CLIERR= 0
2290    000001                  CLIEXI= 1
2291    000002                  CLIBR=  2
2292    000003                  CLIBIF= 3
2293    000004                  CLISPA= 4
2294    000005                  CLINUM= 5
2295    000006                  CLIALP= 6
2296    000007                  CLIALN= 7
2297    000010                  CLIOCT= 8.
2298    000011                  CLIDEC= 9.
2299    000012                  CLISTR= 10.
2300
2301                    ; DEFS FOR COMMAND LINE INTERPRETATION ACTION VALUES
2302    000000                  NULL=0
2303    000001                  CLEAR=1
2304    000002                  SHOW=2
2305    000003                  CHECK=3
2306    000004                  RUN=4
2307    000005                  HLP=5
2308    000006                  CSHEXP=6
2309    000007                  CSHTRN=7
2310    000010                  SETEXP=10
2311    000011                  SETTRN=11
```

```
2312    000012                  SIZE=12
2313    000013                  QCOPY=13
2314    000014                  NUM=14
2315    000015                  OPRMSG=15
2316    000016                  STATUS=16
2317    000017                  ENDGO=17
2318    000020                  CMSG0=20
2319    000021                  CMSG1=21
2320    000022                  CMSG2=22
2321    000023                  CMSG3=23
2322    000024                  CMSG4=24
2323    000025                  CMSG5=25
2324    000026                  CMSG6=26
2325    000027                  ATVMOD=27
2326    000030                  PASMOD=30
2327    000031                  RECMOD=31
2328    000032                  LISMOD=32
2329    000033                  DLLMOD=33
2330    000034                  TRAMOD=34
2331    000035                  TALMOD=35
2332    000036                  NO=36
2333    000037                  ECHO=37
2334    000040                  CRC=40
2335    000041                  PROTO=41
2336    000042                  PASC=42
2337    000043                  MOP=43
2338    000044                  TTLLOP=44
2339    000045                  CBLLOP=45
2340    000046                  LMDLOP=46
2341    000047                  RMDLOP=47
2342    000050                  NOTNUF=50
2343    000051                  BADCHR=51
2344    000052                  DMPS=52
2345    000053                  DMPE=53
2346    000054                  DMPQ=54
2347    000055                  PRNT=55
2348    000056                  MOSC=56
2349    000057                  EXIT=57
2350    000060                  SETET=60
2351                    ;FOLLOWING EQUATES USED IN REPORT CLI
2352    000001                  RPHLP=1         ;PRINT HELP MESSAGE
2353    000002                  RPEXT=2         ;EXIT
2354    000003                  RPLOG=3         ;REPORT EVENT LOG
2355    000004                  RPERR=4         ;'COUNTER/ERROR'
2356    000005                  RPFUL=5         ;'COUNTER/FULL'
2357    000006                  RNOTNF=6        ;MORE COMMAND NEEDED
2358    000007                  RPSWO=7         ;VALIDATE OFFSET
2359
2360
2361                    ;***** DEVICE DEPENDENT EQUATES
2362                    ; MODEM SIGNAL BIT DEFINITONS
2363                    ;       IF SIGNAL AVAILABLE IN DEVICE, EQUATE NAME TO BIT POSITION,
2364                    ;       ELSE EQUATE IT TO = 0
2365
2366    020000                  CTS=    BIT13           ;CLEAR TO SEND (CIRCUIT CB)
2367    001000                  DSR=    BIT9            ;DATA SET READY (CIRCUIT CC)
```

```
2368    010000                  DCD=    BIT12           ;DATA CARRIER DETECT (CIRCUIT CF)
2369    000004                  RTS=    BIT2            ;REQUEST TO SEND (CIRCUIT CA)
2370    040000                  RI=     BIT14           ;RING INDICATOR (CIRCUIT CE)
2371    004000                  SRD=    BIT11           ;SECONDARY RECEIVE DATA
2372
2373
2374
2375
2376                    ; DEVICE SIGNALS
2377    000002                  DTR=    BIT1            ;DATA TERMINAL READY
2378    000010                  HDPLX=  BIT3            ;HALF DUPLEX MODE
2379    000020                  RXENA=  BIT4            ;RECEIVER ENABLE
2380    000040                  DSITEN= BIT5            ;DATA SET CHANGE ENABLE
2381    000100                  RINTEN= BIT6            ;REC INT. ENABLE
2382    000200                  RXDONE= BIT7            ;REC DATA READY
2383    004000                  RXACT=  BIT11           ;REC ACTIVE
2384    000400                  RESET=  BIT8            ;MASTER RESET
2385    001000                  TXACT=  BIT9            ;TX ACTIVE
2386    000200                  TXDONE= BIT7            ;TX BUFFER EMPTY
2387    004000                  TTLL=   BIT11           ;TTL LOOP BIT(INTERNAL)
2388    010000                  CABLOP= BIT12           ;CABLE LOOP(TURN AROUND)
2389    000020                  SEND=   BIT4            ;TX ENABLE
2390    000100                  TINTEN= BIT6            ;TX INT ENABLE
2391    000400                  TSOM=   BIT8            ;TX START OF MSG.
2392    001000                  TEOM=   BIT9            ;TX END OF MSG.
2393    100000                  TERR=   BIT15           ;TX ERROR
2394    100000                  RERR=   BIT15           ;REC OVER RUN
2395    010000                  CRCOK=  BIT12           ;CRC CHAR OK
2396    100000                  DSCA=   BIT15           ;DATA SET CHANGE A
2397    000400                  STRIP=  BIT8            ;SYNC STRIP
2398    000226                  SYN=    226             ;SYNC WORD
2399
```

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

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

                                ;MESSAGE BYTE COUNT TABLE

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

                                ;MESSAGE ADDRESS TABLE

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

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

```
2456  002320                                MSG5:                           ;"INTERPROCESSOR TEST PROGRAM'S (ITEP)" MESSAGE
2457                                                                        ; #1, (DP1:)
2458  002320  077577  040444  052040              .ASCII  <177><177>/$A THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG./
2459  002326  042510  050440  044525
2460  002334  045503  041040  047522
2461  002342  047127  043040  054117
2462  002350  045040  046525  042520
2463  002356  020104  053117  051105
2464  002364  052040  042510  046040
2465  002372  055101  020131  047504
2466  002400  027107
2467  002402  005015  077401  077577              .ASCIZ  <15><12><001><177><177><177><177>
2468  002410  000177
2469  002412                                EMSG5:
2470  002412                                MSG6:                           ;ALPHA-NUMERICS (OR FUTURE COMM TURNAROUND MSG)
2471  002412  022043  021041  023040              .ASCII  /#$!" &'()*+,-.0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ/
2472  002420  024047  025051  026053
2473  002426  027055  030460  031462
2474  002434  032464  033466  034470
2475  002442  035472  036474  037476
2476  002450  040500  041502  042504
2477  002456  043506  044510  045512
2478  002464  046514  047516  050520
2479  002472  051522  052524  053526
2480  002500  054530     132
2481  002503     057  056133  057135              .ASCIZ  ?/[\]^_%?
2482  002510  022537     000
2483  002513                                EMSG6:
2484        002514                                .EVEN
2485
2486                                        ; ***************************************************************************
2487                                        ;THESE THREE STORAGE AREAS MUST NOT BE SEPERATED !!!!
2488
2489  002514  047045  040445               OPBFPT: .ASCII  /%N%A/
2490  002520  000122                        OPBUF:  .BLKB   82.             ;BUFFER FOR OPERATOR SPEC'D MESSAGES
2491  002642                                OPEND:
2492
2493                                        ; THE ABOVE THREE LINES MUST BE KEPT TOGETHER
2494                                        ; ***************************************************************************
2495
2496  002642     033                        MSG8:   .BYTE   33              ;RECEIVE BUFFER FILL PATTERN
2497  002643                                EMSG8:
2498        002644                                .EVEN
```

```
2499                                 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2500                                 ; THE FOLLWING IS THE AREA USED TO TRANSMIT AND REC THE ;
2501                                 ; HEADER MSGS. AND THE START,STACK ACK SEQUENCES.       ;
2502                                 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2503
2504                                 ;; THE TRANSMIT HEADER MESSAGE WILL BE STORED HERE
2505  002644     000                 HDMSG:  .BYTE 0         ;FILLER
2506  002645     201                 HDMID:  .BYTE 201       ;MESSAGE TYPE WILL BE STORED HERE
2507  002646                         HDMTYP:                 ;IF CONTROL MESSAGE, TYPE IS STORED HERE
2508  002646  000000                 HDMCC:  .WORD   0       ;CHAR COUNT GOES HERE
2509  002650     001                 HDMREP: .BYTE   1       ;RESPONSE NUMBER
2510  002651     001                 HDMNUM: .BYTE   1       ;MSG. NUMBER
2511  002652     001                 HDMADR: .BYTE   1       ;ADDR TO.
2512  002653                         HSMSE:
2513          002654                         .EVEN
2514  002654  000006                 HDMC:   .WORD   6       ;CHARACTER COUNT OF HEADER
2515
2516                                 ;; THE RECEIVED HEADER WILL BE STORED HERE
2517  002656     000                 RHDMSG: .BYTE 0
2518  002657     201                 RHDMID: .BYTE 201       ;MESSAGE TYPE GOES HERE
2519  002660                         RHDTYP:                 ;IF CONTROL MESSAGE, TYPE GOES HERE
2520  002660  000000                 RHDMCC: .WORD   0       ;BYTE COUNT GOES HERE
2521  002662     001                 RHDREP: .BYTE   1       ;RESP NUM
2522  002663     001                 RHDNUM: .BYTE   1       ;MSG NUM
2523  002664     001                 RHDADR: .BYTE   1       ;ADDRESS OF TRIB
2524          002666                         .EVEN
2525
```

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

002666  000122                  CMDBUF: .BLKB   82.              ;BUFFER FOR OPERATOR COMMANDS
003010  000000                  KEYWD1: .WORD   0                ;THIS LOC WILL =1 IF CLEAR TYPED, 2 FOR SHOW,
                                                                 ; A 4 IF RUN WAS TYPED, 5 IF HELP WAS TYPED
003012  000000                  QUALFG: .WORD   0                ;THIS LOC HOLDS QUALIFIER VALUE (SIZE OR COPY)
003014  000000                  QUALVL: .WORD   0
003016  012276                  HLPTAB: .WORD   HLP1
003020  012311                          .WORD   HLP2
003022  012426                          .WORD   HLP3
003024  012513                          .WORD   HLP3A
003026  012540                          .WORD   HLP4
003030  012617                          .WORD   HLP4A
003032  012675                          .WORD   HLP5
003034  012776                          .WORD   HLP6
003036                          HLPEND:
                                ;INDEX TABLE FOR REPORT 'RPT>' HELP MESSAGES
003036  013133                  RHLPTB: .WORD   RHLP1
003040  013155                          .WORD   RHLP2
003042  013210                          .WORD   RHLP3
003044  013241                          .WORD   RHLP4
003046  013273                          .WORD   RHLP5
003050  013336                          .WORD   RHLP6
003052                          RHLPEN:

003052  013552  013561  013566  SHTYTB: .WORD   SHTYP0,SHTYP1,SHTYP2,SHTYP3,SHTYP4,SHTYP5,SHTYP6,SHTYP7
003060  013573  013600  013606
003066  013613  013621

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

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

003102  013632                  MODES:  .WORD   M00      ;ADDRESSES OF MODE TYPES IN ASCII
003104  013642                          .WORD   M01
003106  013653                          .WORD   M02
003110  013663                          .WORD   M03
003112  013672                          .WORD   M04
003114  013707                          .WORD   M05
003116  013714                          .WORD   M06

003120  013723                  LOOPS:  .WORD   LP0      ;ADDRESSES OF LOOP TYPES IN ASCII
003122  013733                          .WORD   LP1
003124  013744                          .WORD   LP2
003126  013752                          .WORD   LP3
003130  013765                          .WORD   LP4

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

003132  000000                  P$BUFA: .WORD   0                ;LOC. TO HOLD ADDR. OF CMD LINE BUFFER
003134  000000                  P$TREE: .WORD   0                ;LOC. TO HOLD ADDR. OF PARSING TREE
```

```
 2582  003136  000000                P$ACT:  .WORD   0               ;LOC. TO HOLD ADDR. OF ACTION ROUTINE
 2583  003140  000000                P$CNT:  .WORD   0               ;LOC. TO BE A COUNTER LOCATION
 2584  003142  000000                P$NUM:  .WORD   0               ;LOC. TO HOLD NUMERIC VALUE FROM PARSE
 2585  003144  000000                P$RADX: .WORD   0               ;LOC. TO HOLD RADIX USED(LO) AND +/-(HI BYTE)
 2586  003146     000                P$NNUF: .BYTE   0               ;RETURN =0 IF ENOUGH OF COMMAND FOUND
 2587  003147     000                P$GDBD: .BYTE   0               ;RETURN CODE 0 IF NO ERROR FOUND
 2588
```

```
                                .SBTTL          MESSAGE BUFFERS AND POINTER TABLES

003150  001000  TXBUF:  .BLKB   BUFLIM  ;TRANSMITTER BUFFERS
004150  001000  RXBUF:  .BLKB   BUFLIM  ;RECEIVER BUFFERS
005150  001000  CMPBUF: .BLKB   BUFLIM  ;COMPARISON BUFFERS
006150  000036  PTRTAB: .BLKW   MSGLIM*2        ;TABLE FOR MESSAGE ADDRS. & BYTE COUNTS
006244  000036  PTR13:  .BLKW   MSGLIM*2
006340  000036  PTR23:  .BLKW   MSGLIM*2
006434          PTREND:                 ; END OF MSG. PTR. TABLE

006434  000002          .BLKW   2       ;FILLER FOR OVERFLOW OF RX POINTER TABLE

006440  000000  RXPTR:  .WORD   0       ;RECEIVER MESSAGE POINTER
006442  000000  TXPTR:  .WORD   0       ;TRANMITTER BUFFER POINTER
006444  000000  CMPPTR: .WORD   0       ;COMPARISON BUFFER POINTER
006446  000000  CMPTOT: .WORD   0       ;CMP MSG TOTAL
006450  000000  CTOTCC: .WORD   0       ;COMPARE BUFFER CHAR. COUNT
006452  000000  CCURAD: .WORD   0       ;CURRENT ADDR OF CMP BUFF TO ADD AT

006454  000000  DVTXA:  .WORD   0       ;DEVICE TX ADDR
006456  000000  DVTCC:  .WORD   0       ;DEVICE TX CHAR COUNT
006460  000000  DVTCT:  .WORD   0       ;DEVICE TX MESSAGE COUNT
006462  000000  TXMTOT: .WORD   0       ;TX MSG TOTAL
006464  000000  TTOTCC: .WORD   0       ;TX BUFFER CHAR. COUNT
006466  000000  TCURAD: .WORD   0       ;CURRENT ADDR. OF TX BUFF TO ADD AT

006470  000000  DVRXA:  .WORD   0       ;DEVICE RX ADDR
006472  000000  DVRCC:  .WORD   0       ;DEVICE RX CHAR COUNT
006474  000000  DVRCT:  .WORD   0       ;DEVICE RX MESSAGE COUNT
006476  000000  RXMTOT: .WORD   0       ;RX MSG TOTAL

006500  000000  LNCNT:  .WORD   0       ;NUMBER OF OPERATOR AWAKE MSGS
006502  000000  OPVAR:  .WORD   0       ;OPTTIONAL VARIABLE LOCATION
006504  000000  PSCNT:  .WORD   0       ;PASS COUNTER
006506  000000  ERRCNT: .WORD   0       ;ERROR COUNTER
006510  000000  STADD:  .WORD   0       ;START ADDR.
006512  000000  ENADD:  .WORD   0       ;END ADDR. FOR DUMP
006514  000000  BYTBIT: .WORD   0       ;BYTE BIT FOR DUMP ROUTINE

                ;OTHER MESSSAGE RELATED STORAGE LOCATIONS

006516  000000  MSGTYP: .WORD   0       ;TYPE OF DATA 0=0'S,1=1'S,2=10'S,3=01'S
                                        ;4=CCITT,5=QUICK FOX,6=ALPHA/NUM,7=OPER
006520  000000  CURCC:  .WORD   0       ;TX/RX/CMP CHAR COUNT
006522  000000  CPTRR:  .WORD   0       ;CUPRENT RX POINTER
006524  000000  CPTR:   .WORD   0       ;CURRENT POINTER
006526  000000  CURADD: .WORD   0       ;CURRENT TX/RX/CMP START ADDD
006530  000000  TOTCC:  .WORD   0       ;TOTAL CHAR COUNT NOT MORE THEN 'BUFLIM'
006532  000000  OFSET:  .WORD   0       ;OFFSET COUNT
006534  000000  TEMP:   .WORD   0       ;TEMPORARY LOCATIONS (USED A LOT)
006536  000000  TEMP1:  .WORD   0
006540  000000  TEMP2:  .WORD   0
006542  000000  TEMP3:  .WORD   0
006544  000000  TEMP4:  .WORD   0
006546  000000  TEMP5:  .WORD   0
006550  000000  CONOTM: .WORD   0       ;CONTROL OUT ERROR MSG. ADDRESS
```

```
 2645  006552  000000                CONTIN: .WORD   0       ;WORD FOR CONTROL IN
 2646  006554     000                GOOD:   .BYTE   0       ;BYTE TO HOLD EXPECTED MESSAGE DATA BYTE FOR ERR REPORT
 2647  006555     000                BAD:    .BYTE   0       ;BYTE TO HOLD RECEIVED MESSAGE DATA BYTE FOR ERR REPORT
 2648  006556  000000                DATAWORD:.WORD  0       ;STORAGE LOCATION FOR TRANSMIT DATA
 2649
```

```
                                ;MORE INDEPENDENT CODE STORAGE LOCATIONS

006560  000000                LOGUNT: .WORD   0       ;LOC. TO HOLD LOGICAL UNIT NUMBER
006562  000000                PCADD:  .WORD   0       ;LOC. HOLD PC OF CALLING ROUTINE
006564  000000                DCLFLG: .WORD   0       ;CLEANUP & EXIT FLAG -1 = EXIT TEST
006566  000000                RESFLG: .WORD   0       ;LOC TO HOLD FLAG (-1) THAT A RESTART WAS GIVEN
006570  000000                MODTYP: .WORD   0       ;DCLT MODE OF OPERATION TYPE
                                                      ;  (0=REC-ONLY, 1=TX-ONLY, 2=PASSIVE-LOOPBK,
                                                      ;   3=ACTIVE-LOOPBK, 4=DOWN L.L., 5=TALK, 6=LISTEN)
006572  000000                MLTYP:  .WORD   0       ;MAINTENANCE LOOP TYPE (0=NONE, 1=INTERNAL TTL,
                                                      ;  2=CABLE, 3=MODEM-ANALOG LOOPBK (LOCAL),
                                                      ;  4=MODEM-DIGITAL LOOPBK (REMOTE), 5=MOP)
006574  000000                FHDPLX: .WORD   0       ;FULL OR HALF DUPLEX FLAG (1=FULL FROM P-TABLE)
006576  000002                PARAM:  .WORD   2       ;PROGRAM PARAMETERS
                                                      ; BIT0= STATUS MSGS TO OPR PRINTED (1=YES)
                                                      ; BIT1= DATA CHECKING DONE ON RCVD MSGS (1=YES)
                                                      ; BIT2= ECHO (TRANSMIT) RCV'D MSG.(PASSIVE)(1=YES)
                                                      ; BIT3= MODEM STATUS CHECK (1=YES)
                                                      ; BIT4= CRC CALC./CHECK DONE (1=YES)
                                                      ; BIT5= PROTOCOL EMULATION (1=YES)
                                                      ; BIT6= PROTOCOL IS RUNNING
                                                      ; BIT7= FATAL FAULT IN PROTOCOL--ABORTING!!

006600  000000                RPASS:  .WORD   0       ;PASS NUMBER FROM RUN COMMAND
006602  000000                FLAG:   .WORD   0       ;DEVICE FLAG WORD

                                ;MODE DISPATCH TABLE
006604  032210                MODE:   .WORD   RXONLY  ;RX ONLY DISPATCH
006606  032242                        .WORD   TXONLY  ;TX ONLY DISPATCH
006610  032302                        .WORD   PLCK    ;PASSIVE LOOP BACK DISP
006612  032336                        .WORD   ALCK    ;ACTIVE LOOP BACK DISP
006614  033570                        .WORD   DLL     ;DOWN LINE LOAD DISP
006616  033614                        .WORD   TALCK   ;TALK MODE DISPATCH
006620  034060                        .WORD   LISCK   ;LISTEN MODE DISPATCH


                                .SBTTL          CLOCK TABLES, EVENT LOG AND POINTERS
006622  000000                CLKCSR: .WORD   0       ;CLOCK CSR ADDRESS
006624  000000                CLKBR:  .WORD   0       ;CLOCK INTERRUPT LEVEL
006626  000000                CLKVEC: .WORD   0       ;CLOCK INTERRUPT VECTOR
006630  000074                CLKHZ:  .WORD   60.     ;CLOCK'S HERTZ RATE
006632  000000                CLKEN:  .WORD   0       ;CLOCK'S CSR VALUE TO INTRPT. ENABLE IT

006634  000000                TIMMIN: .WORD   0       ;PLACE TO KEEP TIME-SINCE-START
006636  000000                TIMSEC: .WORD   0
006640  000000                TIMTCK: .WORD   0       ;PLACE TO KEEP # OF TICKS/SEC

006642  000000                TIMER1: .WORD   0       ;EVENT TIMER #1 (TICKS)
006644  000000                TIMER2: .WORD   0       ;EVENT TIMER #2 (TICKS)
006646  000000                TIMERS: .WORD   0       ;EVENT TIMER #3 (SECONDS)
```

```
 2701                                   ;EVENT LOG TABLE AND ITS NEXT ENTRY POINTER
 2702  006650  006652                   EVTPTR: .WORD   EVTLOG  ;POINTER TO NEXT FREE SPACE IN EVENT LOG
 2703  006652  000341                   EVTLOG: .BLKW   225.    ;EVENT LOG BUFFER
 2704  007554  000001                   EVTEND: .BLKW   1.      ;APPROXIMATE END OF EVENT TABLE (ALLOWS CIRCULAR QUE)
 2705
 2706                                   .SBTTL          MODEM DATA SECTION
 2707
 2708  007556  000000                   MODS:   .WORD   0       ;MODEM STATUS
 2709
 2710
```

```
2711
2712                                    ;TABLE OF MODEM SIGNAL BIT DEFINITIONS
2713
2714  007560  020000                    MOBITS: .WORD   CTS             ;CLEAR TO SEND (CIRCUIT CB)
2715  007562  001000                            .WORD   DSR             ;DATA SET READY (CIRCUIT CC)
2716  007564  010000                            .WORD   DCD             ;DATA CARRIER DETECT (CIRCUIT CF)
2717  007566  000004                            .WORD   RTS             ;REQUEST TO SEND (CIRCUIT CA)
2718  007570  040000                            .WORD   RI              ;RING INDICATOR (CIRCUIT CE)
2719  007572  004000                            .WORD   SRD             ;SECONDARY RECEIVE DATA (CIRCUIT SBB)
2720  007574                          MOBITE:
2721
2722                                    ;TABLE OF ADDRESSES OF MODEM SIGNAL MESSAGE POSITIONS
2723
2724  007574  016502                    MOMSGS: .WORD   EVMCTS           ;CLEAR TO SEND (CIRCUIT CB)
2725  007576  016506                            .WORD   EVMDSR           ;DATA SET READY (CIRCUIT CC)
2726  007600  016512                            .WORD   EVMDCD           ;DATA CARRIER DETECT (CIRCUIT CF)
2727  007602  016516                            .WORD   EVMRTS           ;REQUEST TO SEND (CIRCUIT CA)
2728  007604  016522                            .WORD   EVMRI            ;RING INDICATOR (CIRCUIT CE)
2729  007606  016526                            .WORD   EVMSRD           ;SECONDARY RECEVIE DATA
2730
2731                                    ;TABLE OF ADDRESSES OF EVENT DESCRIPTION MESSAGES
2732                                    ;       ORDER CORRESPONDS TO MESSAGE TYPE VALUES
2733
2734  007610  015107                    EVTLST: .WORD   EDTXQ   ;TRANSMIT MESSAGE QUEUED
2735  007612  015133                            .WORD   EDTXC   ;TRANSMIT OF MESSAGE COMPLETE
2736  007614  015162                            .WORD   EDRXQ   ;RECEIVE MESSAGE SPACE QUEUED
2737  007616  015207                            .WORD   EDRXC   ;MESSAGE RECEIVED - RECEIVE COMPLETE
2738  007620  015235                            .WORD   EDDER   ;DEVICE INFORMATION
2739  007622  015302                            .WORD   EDDVI   ;DEVICE INITIALIZE STARTED
2740  007624  015252                            .WORD   EDDCK   ;DATA COMPARISON DONE
2741  007626  015434                            .WORD   EDMOS   ;MODEM STATUS CHANGE
2742  007630  015330                            .WORD   EDDLE   ;DATA COMPARE LENGTH ERROR
2743  007632  015365                            .WORD   EDDDE   ;DATA COMPARE DATA ERROR
2744  007634  015420                            .WORD   EDEOP   ;END OF PASS
2745  007636  015460                            .WORD   EDABO   ;^C ABORT
2746
2747                                    ;LOCATIONS USED DURING EVENT REPORTING
2748
2749  007640  000000                    EVTSEC: .WORD   0       ;TEMPORARY LOCS TO KEEP EVENT TIME WHILE REPORTING
2750  007642  000000                    EVTMIN: .WORD   0
2751  007644  000000                    EVTTCK: .WORD   0
2752  007646  000000                    EVTADD: .WORD   0       ;TEMP. LOC. TO HOLD ADDRESS DURING EVENT REPORTING
2753  007650  000000                    EVTBCT: .WORD   0       ;       ''      ''  BYTE COUNT   ''      ''      ''
2754  007652  000000                    EVTTMP: .WORD   0       ;       ''      ''  OTHER DATA   ''      ''      ''
2755
2756                                    ;REPORT CODING DISPATCH TABLE
2757
2758  007654  022300                    RPTDSP: .WORD   RPTTXQ  ;TRANSMIT QUEUED ENTRY DECODING
2759  007656  022300                            .WORD   RPTTXQ  ;TRANSMIT COMPLETE ENTRY DECODING
2760  007660  022300                            .WORD   RPTTXQ  ;RECEIVER QUEUED ENTRY DECODING
2761  007662  022300                            .WORD   RPTTXQ  ;RECEIVER COMPLETE ENTRY DECODING
2762  007664  022352                            .WORD   RPTDER  ;DEVICE ERROR ENTRY DECODING
2763  007666  022446                            .WORD   RPTDVI  ;DEVICE INIT ENTRY DECODING
2764  007670  022642                            .WORD   RPTDCK  ;DATA COMPARISON ENTRY DECODING
2765  007672  022716                            .WORD   RPTMSC  ;REPORT MODEM STATUS CHANGE
2766  007674  022642                            .WORD   RPTDLE  ;DATA COMPARISON LENGH ERROR
```

```
 2767  007676  022566                    .WORD   RPTDDE  ;DATA COMPARISON DATA ERROR
 2768  007700  022512                    .WORD   RPTEOP  ;END OF PASS
 2769  007702  022512                    .WORD   RPTABO  ;^C ABORT
 2770
 2771  007704  000000            DEV1:   .WORD   0               ;TEMP LOCS TO HOLD DATA FOR EVENT REPORTING
 2772  007706  000000            DEV2:   .WORD   0               ;  AND SHOW MODE,... SUBROUTINE
 2773  007710  000000            DEV3:   .WORD   0
 2774  007712  000000            DEV4:   .WORD   0
 2775
```

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

007714                  CLITRE:

                        ;FIRST KEYWORD
007714                          CLI     CLISPA,0,N10$                   ;SKIP ANY LEADING SPACES
007720                  N10$:   CLI     <'?>,HLP,N42$                   ;IS THE FIRST NON-SP CHAR A "?"
007724                          CLI     CLIEXI,0                        ;  IF YES DO "HLP" AND EXIT
007726                  N42$:   CLI     CLISTR,HLP,N43$,<'HELP'>        ;ELSE, IS FIRST WORD A "HELP"
007742                          CLI     CLIEXI,0                        ;  IF YES DO "HLP" AND EXIT
007744                  N43$:   CLI     CLISTR,PRNT,N44$,<'PRINT'>      ;ELSE, IS FIRST WORD A "PRINT"
007760                          CLI     CLIEXI,0                        ;  IF YES DO "PRINT" AND EXIT
007762                  N44$:   CLI     CLISTR,EXIT,N45$,<'EXIT'>       ;ELSE, IS FIRST WORD 'EXIT'
007776                          CLI     CLIEXI,0                        ;  IF YES DO "EXIT" AND EXIT
010000                  N45$:   CLI     CLISTR,RUN,N46$,<'RUN'>         ;ELSE, IS FIRST WORD A "RUN"
010012                          CLI     CLIBR,0,N80$                    ;  IF YES DO "RUN" & GOTO N80$
010016                  N46$:   CLI     CLISTR,NOTNUF,N40$,<'DUMP'>     ;ELSE, IS FIRST WORD A "DUMP"
010032                          CLI     CLIBR,0,N50$                    ;  IF YES GOTO N80$
010036                  N40$:   CLI     CLISTR,CLEAR,N20$,<'CLEAR'>     ;ELSE, IS FIRST WORD A "CLEAR"
010052                          CLI     CLIBR,NOTNUF,N100$              ;  IF YES DO "CLR" & GOTO N100$
010056                  N20$:   CLI     <'S>,NOTNUF,N30$                ;ELSE, IS FIRST CHAR. A "S"
010062                          CLI     CLISTR,SHOW,N25$,<'HOW'>        ;  IF YES IS REST OF WORD "HOW"
010074                          CLI     CLIBR,0,N100$                   ;    IF YES, DO "SHOW",BR N100$
010100                  N25$:   CLI     CLISTR,0,N30$,<'ET'>            ;  ELSE, IS REST OF WORD "ET"
010112                          CLI     CLIBR,0,N110$                   ;    IF YES, DO "SET", BR N110$
010116                  N30$:   CLI     CLIERR,0                        ;OTHERWISE "ILL CMD" - EXIT

                        ;SECOND KEYWORD (MODE=) FOR RUN COMMAND

010120                  N80$:   CLI     CLISPA,0,N30$                   ;SKIP LEADING SPS, IF NONE-ERR
010124                  N81$:   CLI     CLISTR,NOTNUF,N30$,<'MODE'>     ;IS NEXT WORD "MODE="
010140                          CLI     <'=>,0,N30$                     ;  IF NO, IT'S WRONG -ERR -EXIT
010144                          CLI     CLISTR,ATVMOD,N82$,<'ACTIVE'>   ;IS NEXT WORD "ACTIVE"
010162                          CLI     CLIBR,0,N115$                   ;  IF YES, DO "ACTIVE",BR N115$
010166                  N82$:   CLI     CLISTR,PASMOD,N83$,<'PASSIVE'>  ;IS NEXT WORD "PASSIVE"
010204                          CLI     CLIBR,0,N115$                   ;  IF YES, DO "PASSVE",BR N115$
010210                  N83$:   CLI     CLISTR,RECMOD,N84$,<'RECEIVE'>  ;IS NEXT WORD "RECEIVE"
010226                          CLI     CLIBR,0,N115$                   ;  IF YES, DO "RECVE",BR N115$
010232                  N84$:   CLI     CLISTR,LISMOD,N85$,<'LISTEN'>   ;IS NEXT WORD "LISTEN"
010250                          CLI     CLIBR,0,N115$                   ;  IF YES, DO "LISTEN",BR N115$
010254                  N85$:   CLI     CLISTR,DLLMOD,N86$,<'DOWNLINELOAD'>     ;IS NEXT WORD "DOW..."
010300                          CLI     CLIBR,0,N115$                   ;  IF YES, DO "DWNLL",BR N115$
010304                  N86$:   CLI     <'T>,0,N30$                     ;IS NEXT CHAR A "T"
```

```
2832  010310                          CLI     CLISTR,TRAMOD,N87$,<'RANSMIT'>  ; IS REST OF WORD 'RANSMIT'
2833  010326                          CLI     CLIBR,0,N115$                   ;  IF YES, DO 'TRANSM',BR N115$
2834  010332                  N87$:   CLI     CLISTR,TALMOD,N30$,<'ALK'>      ; IS REST OF WORD 'ALK'
2835  010344                          CLI     CLIBR,0,N115$                   ;  IF YES, DO 'TALK',BR N115$
2836                                                                          ; IF NO, ERROR - EXIT
2837
2838                                  ;SECOND KEYWORD (FOR CLEAR OR SHOW)
2839  010350                  N100$:  CLI     CLISPA,0,N30$                   ;SKIP LEADING SPACES, NONE=ERR
2840  010354                  N102$:  CLI     CLISTR,CSHEXP,N104$,<'EXPECTBUFF'>      ;IS NEXT WORD 'EXPE...'
2841  010376                          CLI     CLIEXI,0                        ; IF YES, DO CLR-EXP,EXIT
2842  010400                  N104$:  CLI     CLISTR,CSHTRN,N30$,<'TRANSMITBUFF'>     ;IS NEXT WORD 'TRANS..'
2843  010424                          CLI     CLIEXI,0                        ; IF YES, DO CLR-TRN,EXIT
2844                                                                          ;IF NO - ERROR - EXIT
2845
2846
2847                                  ;SECOND KEYWORD (FOR SET)
2848  010426                  N110$:  CLI     CLISPA,0,N30$
2849  010432                  N111$:  CLI     CLISTR,SETEXP,N112$,<'EXPECT'>
2850  010450                          CLI     CLIBR,0,N120$
2851  010454                  N112$:  CLI     CLISTR,SETTRN,N30$,<'TRANSMIT'>
2852  010474                          CLI     CLIBR,0,N120$
2853
2854                                  ;GET ADDRESSES FOR DUMP COMMAND
2855  010500                  N50$:   CLI     CLIALP,0,N51$
2856  010504                  N51$:   CLI     CLISPA,0,N52$
2857  010510                  N52$:   CLI     CLIOCT,DMPS,N30$
2858  010514                          CLI     <'->,NOTNUF,N125$
2859  010520                          CLI     CLIOCT,DMPE,N30$
2860  010524                          CLI     <'/>,NOTNUF,N125$
2861  010530                          CLI     <'B>,DMPQ,N30$
2862  010534                          CLI     CLIBR,0,N125$
2863
2864                                  ;QUALIFIERS FOR THE RUN COMMAND
2865  010540                  N115$:  CLI     CLIALP,0,N114$
2866  010544                  N114$:  CLI     <'/>,NOTNUF,N125$
2867  010550                          CLI     CLISTR,NO,N116$,<'NO'>
2868  010562                  N116$:  CLI     <'C>,0,N117$
2869  010566                          CLI     CLISTR,CHECK,N117$,<'HECK'>
2870  010602                          CLI     CLIBR,0,N115$
2871
2872
2873  010606                  N117$:  CLI     CLISTR,STATUS,N118$,<'STATUS'>
2874  010624                          CLI     CLIBR,0,N115$
2875  010630                  N118$:  CLI     CLISTR,ECHO,N119$,<'ECHO'>
2876  010644                          CLI     CLIBR,0,N115$
2877
2878  010650                  N119$:  CLI     <'P>,0,N132$
2879  010654                          CLI     CLISTR,PROTO,N130$,<'ROTOCOL'>
2880  010672                          CLI     CLIBR,0,N115$
2881  010676                  N130$:  CLI     CLISTR,0,N30$,<'ASS'>
2882  010710                          CLI     CLIBR,0,N150$
2883
2884  010714                  N132$:  CLI     CLISTR,MOSC,N131$,<'MODEM'>
2885  010730                          CLI     CLIBR,0,N115$
2886
2887  010734                  N131$:  CLI     CLISTR,0,N30$,<'LOOP'>
```

```
 2888  010750                          CLI     CLIBR,0,N140$
 2889
 2890                                  ;GET MESSAGE TYPE FOR SET MESSAGE COMMANDS
 2891  010754                          N120$:  CLI     <'=>,0,N30$
 2892
 2893                                  ;   LOOK FOR DEFAULT MESSAGE NAME
 2894  010760                          N60$:   CLI     CLISTR,CMSG1,N61$,<'ONES'>
 2895  010774                                  CLI     CLIBR,0,N121$
 2896  011000                          N61$:   CLI     CLISTR,CMSG0,N62$,<'ZEROES'>
 2897  011016                                  CLI     CLIBR,0,N121$
 2898  011022                          N62$:   CLI     CLISTR,CMSG2,N63$,<'1ALT'>
 2899  011036                                  CLI     CLIBR,0,N121$
 2900  011042                          N63$:   CLI     CLISTR,CMSG3,N64$,<'0ALT'>
 2901  011056                                  CLI     CLIBR,0,N121$
 2902  011062                          N64$:   CLI     CLISTR,CMSG5,N65$,<'ITEP'>
 2903  011076                                  CLI     CLIBR,0,N121$
 2904  011102                          N65$:   CLI     CLISTR,CMSG4,N66$,<'CCITT'>
 2905  011116                                  CLI     CLIBR,0,N121$
 2906  011122                          N66$:   CLI     CLISTR,CMSG6,N67$,<'ALPHA'>
 2907  011136                                  CLI     CLIBR,0,N121$
 2908  011142                          N67$:   CLI     CLISTR,SETET,N68$,<'TRANSMIT'>
 2909  011162                                  CLI     CLIBR,0,N125$
 2910                                  ;   LOOK FOR QUOTED MESSAGE
 2911  011166                          N68$:   CLI     <''>,OPRMSG,N30$
 2912  011172                          N70$:   CLI     <''>,ENDQO,N71$
 2913  011176                                  CLI     CLIBR,0,N121$
 2914  011202                          N71$:   CLI     CLISPA,0,N72$
 2915  011206                          N72$:   CLI     CLIALN,0,N73$           ;ONLY A-Z,SP,TAB, OR 0-9 BETWEEN '"S
 2916  011212                                  CLI     CLIBR,0,N70$
 2917  011216                          N73$:   CLI     CLIERR,BADCHR           ;PRINT ERROR IF NONE LEGAL CHAR FOR '"S
 2918
 2919                                  ;GET QUALIFIERS (SIZE OR COPY) FOR SET MESSAGE COMMANDS
 2920  011220                          N121$:  CLI     CLIALP,0,N123$
 2921  011224                          N123$:  CLI     <'/>,NOTNUF,N125$
 2922  011230                                  CLI     CLISTR,SIZE,N122$,<'SIZE'>
 2923  011244                                  CLI     CLIBR,0,N126$
 2924  011250                          N122$:  CLI     CLISTR,QCOPY,N30$,<'COPY'>
 2925  011264                                  CLI     CLIBR,0,N126$
 2926
 2927                                  ;NUMER FOR SIZE OR COPY
 2928  011270                          N126$:  CLI     <'=>,0,N30$
 2929  011274                                  CLI     CLIDEC,NUM,N30$
 2930  011300                                  CLI     CLIBR,0,N121$
 2931
 2932                                  ;GET MAINTENANCE LOOP TYPE FOR RUN "LOOP" QUIALIFIER
 2933  011304                          N140$:  CLI     <'=>,0,N30$
 2934
 2935
 2936  011310                          N141$:  CLI     CLISTR,TTLLOP,N142$,<'INTERNALTTL'>
 2937  011332                                  CLI     CLIBR,0,N115$
 2938  011336                          N142$:  CLI     CLISTR,CBLLOP,N143$,<'CABLE'>
 2939  011352                                  CLI     CLIBR,0,N115$
 2940  011356                          N143$:  CLI     CLISTR,LMDLOP,N144$,<'LOCALMODEM'>
 2941  011400                                  CLI     CLIBR,0,N115$
 2942  011404                          N144$:  CLI     CLISTR,RMDLOP,N30$,<'REMOTEMODEM'>
 2943  011426                                  CLI     CLIBR,0,N115$
```

```
 2944
 2945                                   ;GET LINE NUMBER FOR "PASS" RUN QUALIFIER
 2946  011432                           N150$:  CLI     <'=>,0,N30$
 2947  011436                                   CLI     CLIDEC,PASC,N30$
 2948  011442                                   CLI     CLIBR,0,N115$
 2949
 2950
 2951
 2952                                   ;END-OF-LINE
 2953  011446                           N125$:  CLI     CLIEXI,0
 2954
```

```
                                ;DEVICE DEPENDENT STORAGE LOCATIONS FOR
                                ;  CURRENT DEVICE PARAMTERS

011450  000000                  RXCSR:  .WORD   0               ;RECEIVE STATUS REGISTER
011452  000000                  PARCSR: .WORD   0               ;STATUS REGISTER
011454  000000                  RXDBUF: .WORD   0               ;RECEIVE DATA BUFFER
011456  000000                  TXCSR:  .WORD   0               ;TRANSMIT STATUS REGISTER
011460  000000                  TXDBUF: .WORD   0               ;TRANSMIT DATA BUFFER

011462  000000                  INVEC:  .WORD   0               ;INPUT INTERRUPT VECTOR ADDRESS
011464  000000                  OUTVEC: .WORD   0               ;OUTPUT INTERRUPT VECTOR ADDRESS
011466  000000                  INTPRI: .WORD   0               ;INTERRUPT PRIORITY

011470  100226                  DUPPAR: .WORD   100226          ;THIS WORD IS BROKEN DOWN AS FOLLOWS
                                                                ;BITS 0-7 =SYNC WORD
                                                                ;BIT 9 = CRC ENABLE

                                                                ;BIT 15 = DDCMP MODE

011472  000000                  CMODS:  .WORD   0               ;CURRENT MODEM
011474  000000                  IRXCSR: .WORD   0               ;IMAGE OF RXCSR
011476  000000                  IRXDBUF:.WORD   0               ;IMAGE OF RXDBUF
011500  000000                  MSGPTR: .WORD   0               ;MSG PTR.FOR HEADER OR CONTROL
011502  000000                  MSGCC:  .WORD   0               ;MSG COUNTER OR CC
011504  000000                  SYNCC:  .WORD   0               ;SYNC CHAR COUNT.
011506  000000                  SYNCW:  .WORD   0               ;SYNC WORD.PLUS TSOM BIT.
011510  000000                  RMSGPT: .WORD   0               ;MSG PTR FOR REC
011512  000000                  RMSGCC: .WORD   0               ;CHAR COUNTER FOR REC
011514  000000                  BCCW:   .WORD   0               ;CRC HOLDING LOC.
011516  000000                  MGLCNT: .WORD   0               ;COUNT OF GLITCH ERRORS
011520  000000                  MHRCNT: .WORD   0               ;COUNT OF HARD ERRORS
011522  000000                  RNODE:  .WORD   0               ;1=REMOTE NODE ITEP,0=NON ITEP

                                ;       ERRTBL
```

```
                                .SBTTL  GLOBAL TEXT SECTION

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

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

                                        DEVTYP  <DUP-11>
011524                                                                  L$DVTYP::
011524                                                                          .ASCIZ  /DUP-11/
011524  052504  026520  030461
011532    000
        011534                                                          .EVEN

                                .SBTTL          PROGRAM IDENTIFICATION
                                ; TEST DESCRIPTION
                                ;
                                        DESCRIPT        <DUP-11 DATA COMM LINK TEST >
011534                                                                  L$DESC::
011534                                                                          .ASCIZ  /DUP-11 DATA COM
011534  052504  026520  030461
011542  042040  052101  020101
011550  047503  046515  046040
011556  047111  020113  042524
011564  052123  000040
                                                                        .EVEN
                                .EVEN
```

```
3033                                    .SBTTL          GLOBAL FORMAT STATEMENTS, MESSAGES, AND ASCII INFO
3034
3035  011570  041504  052114  000076    CLI$PM: .ASCIZ  /DCLT>/
      011576  050122  037124     000    CLI$RP: .ASCIZ  /RPT>/
      011603     045  022516  037501    CLIERM: .ASCIZ  /%N%A?ILL CMD-BAD SYNTX?/
      011633     045  022516  037501    CLINUF: .ASCIZ  /%N%A?INCMPLTE CMD?/
      011656  047045  040445  047077    CLINBG: .ASCIZ  /%N%A?NUM TOO BIG?/
      011700  047045  040445  041077    CLIBRX: .ASCIZ  /%N%A?BAD RADIX?/
      011720  047045  040445  021077    CLIBDL: .ASCIZ  /%N%A?"LOOP" VALID ONLY IN ACTIVE?/
      011762  047045  040445  021077    CLINPS: .ASCIZ  /%N%A?"ECHO" VALID ONLY IN PASSIVE?/
      012025     045  022516  037501    CLIBCR: .ASCIZ  /%N%A?ILL CHR- "A-Z,0-9,SP,TAB" ONLY?/
      012072  047045  040445  021077    CLISE0: .ASCIZ  /%N%A?"SIZE=0" NOT VALID?/
      012123     045  022516  037501    CLIPW:  .ASCIZ  /%N%A?TRANSMIT & EXPECT LIST MUST BE IDENTICAL FOR LOOP?/
      012213     045  022516  052101    HLP0:   .ASCIZ  /%N%ATHIS IS DCLT. TYPE "H" OR "?" FOR DETAILS/
      012271     045  022516  000124    HLPF:   .ASCIZ  /%N%T/
      012276  041504  052114  041440    HLP1:   .ASCIZ  /DCLT CMDS:/
      012311     040  046103  040505    HLP2:   .ASCII  / CLEAR OR SHOW  EXPECTLIST OR TRANSMITLIST/<15><12>
      012365     040  051120  047111              .ASCII  / PRINT/<15><12>
      012375     040  054105  052111              .ASCII  / EXIT/<15><12>
      012404  042040  046525  020120              .ASCIZ  ? DUMP START-END/B?
      012426  051440  052105  042440    HLP3:   .ASCIZ  ? SET EXPECTMSG OR TRANSMITMSG=TYPE/SIZE=N OR /COPY=N?
      012513     040  042523  020124    HLP3A:  .ASCIZ  / SET EXPECT=TRANSMIT/
      012540  020040  052040  050131    HLP4:   .ASCIZ  ?   TYPE=ONES,ZEROES,1ALT,0ALT,ITEP,CCITT,ALPHA?
      012617     040  020040  020040    HLP4A:  .ASCIZ  /       OR "OPR SPCD=A-Z,SP,TAB,0-9 IN QUOTES"/
      012675     040  052522  020116    HLP5:   .ASCIZ  ? RUN MODE=MTYP/LOOP=LTYP/CHECK,PROTOCOL,STATUS,ECHO,MODEM,PASS=N?
      012776  020040  046440  054524    HLP6:   .ASCII  /   MTYP=TRAN,REC,ACT,PAS,TAL,LIS,DOWN/<15><12>
      013045     040  020040  052114              .ASCIZ  /   LTYP=INT,CAB,LOC,REM/

      013075     045  022516  052101    RHLP0:  .ASCIZ  /%N%ATYPE "H" OR "?" FOR HELP!/
      013133     104  046103  020124    RHLP1:  .ASCIZ  /DCLT REPORT CMDS:/
      013155     114  043517  026440    RHLP2:  .ASCIZ  /LOG - PRINT DCLT EVENT LOG/
      013210  054105  052111  026440    RHLP3:  .ASCIZ  /EXIT - EXIT REPORT LEVEL/
      013241     110  046105  020120    RHLP4:  .ASCIZ  /HELP - PRINT THIS MESSAGE/
      013273     103  052517  052116    RHLP5:  .ASCIZ  ?COUNTERS/SW - PRINT DDCMP COUNTERS?
      013336  044127  051105  020105    RHLP6:  .ASCIZ  ?WHERE /SW=FULL, /ERRORS, /OFFSET=NN(O)?
      013405     045  022516  047501    RPTIV:  .ASCIZ  /%N%AOFFSET INVALID/
      013430  047045  040445  042104    RPTNV:  .ASCIZ  /%N%ADDCMP COUNTERS VALID ONLY WITH PROTOCOL SELECTED./
      013516  047045  040445  051515    SHMSG:  .ASCIZ  ?%N%AMSG: TYPE=%T%A/SIZE=%D3?
      013552  042532  047522  051505    SHTYP0: .ASCIZ  /ZEROES/
      013561     117  042516  000123    SHTYP1: .ASCIZ  /ONES/
      013566  040461  052114     000    SHTYP2: .ASCIZ  /1ALT/
      013573     060  046101  000124    SHTYP3: .ASCIZ  /0ALT/
      013600  041503  052111  000124    SHTYP4: .ASCIZ  /CCITT/
      013606  052111  050105     000    SHTYP5: .ASCIZ  /ITEP/
      013613     101  050114  040510    SHTYP6: .ASCIZ  /ALPHA/
      013621     117  051120  051440    SHTYP7: .ASCIZ  /OPR SPEC/
      013632  042522  042503  053111    M00:    .ASCIZ  /RECEIVE/
      013642  051124  047101  046523    M01:    .ASCIZ  /TRANSMIT/
      013653     120  051501  044523    M02:    .ASCIZ  /PASSIVE/
      013663     101  052103  053111    M03:    .ASCIZ  /ACTIVE/
      013672  047504  047127  044514    M04:    .ASCIZ  /DOWNLINELOAD/
      013707     124  046101  000113    M05:    .ASCIZ  /TALK/
      013714  044514  052123  047105    M06:    .ASCIZ  /LISTEN/
      013723     000                    LP0:    .ASCIZ  //
      013724  046057  047517  036520    LP00:   .ASCIZ  ?/LOOP=?
      013733     111  052116  051105    LP1:    .ASCIZ  ?INTERNAL?
```

```
013744  040503  046102  000105  LP2:    .ASCIZ  ?CABLE?
013752  047514  040503  046514  LP3:    .ASCIZ  ?LOCALMODEM?
013765     122  046505  052117  LP4:    .ASCIZ  ?REMOTEMODEM?
014001     116     117          PNST:   .ASCII  /NO/
014003     123  040524  052524  PST:    .ASCIZ  /STATUS/
014012  047516                   PNCK:   .ASCII  /NO/
014014  044103  041505  000113  PCK:    .ASCIZ  /CHECK/
014022  047516                   PNEC:   .ASCII  /NO/
014024  041505  047510     000  PEC:    .ASCIZ  /ECHO/
014031     116     117          PNMS:   .ASCII  /NO/
014033     115  042117  046505  PMS:    .ASCIZ  /MODEM/
014041     116     117          PNPR:   .ASCII  /NO/
014043     120  047522  047524  PPR:    .ASCIZ  /PROTOCOL/
014054  047045  040445  044514  LISP:   .ASCIZ  /%N%ALIS>/
014065     124  045514  000076  OPRMM:  .ASCIZ  /TLK>/
014072  044124  051511  040440  L5060:  .ASCIZ  /THIS A 50. OR 60. HZ. LSI-11:/
                                        .EVEN


                                ;
                                ; FORMAT STATEMENTS USED IN PRINT CALLS
                                ;

014130  047045  040445  047504  DLLCM:  .ASCIZ  /%N%ADOWN LINE LOAD NOT SUPPORTED BY THIS DEVICE/


014210  047045  040445  046103  BDCLK:  .ASCIZ  /%N%ACLOCK NOT FOUND/
014234  047045  040445  040502  NOCLK:  .ASCIZ  /%N%ABAD CLOCK - PROGRAM WILL HANG ON ''TIMEOUT''!!/
014315     115  054101  020056  TABEX:  .ASCIZ  /MAX. CHAR. MSG COUNT EXCEEDED -/
014355     102  043125  042506  BUFEX:  .ASCIZ  /BUFFER FULL -/
014373     045  022516  022524  MSGTRN: .ASCIZ  /%N%T%A MSG. NOT BUILT !!/
014424  047045  040445  044103  MSGTRU: .ASCIZ  /%N%ACHAR. COUNT EXCEEDS BUFF LIMIT - MSG TRUNCATED/
014507     045  022516  032523  SHF0:   .ASCIZ  ?%N%S5%AMODE=%T%T%T%A/PASS=%Z5?
014545     045  022516  032523  SHF1:   .ASCIZ  ?%N%S5%S5%S5%A/%T%A/%T%A/%T%A/%T%A/%T?
014612  051445  022465  052101  EFM2:   .ASCIZ  /%S5%ATOTAL MISMATCHES IN MSG = %D5/
014655     045  022516  031523  PCPM:   .ASCIZ  /%N%S3%ACALLED FROM PC=%O6/
014707     045  032523  040445  EFM11:  .ASCIZ  /%S5%ACOMPARE COUNT=%D5%S3%ARECEIVE COUNT=%D5/
014764  047515  042504  020115  MSCMS:  .ASCIZ  /MODEM STATUS CHANGES FOR THIS PASS WERE../
015036  051445  022465  044101  EFM13:  .ASCIZ  /%S5%AHARD CHANGES=%D5%A%S3%AGLITCHES=%D5/

                                ;EVENT DESCRIPTION MESSAGES

015107     124  040522  051516  EDTXQ:  .ASCIZ  /TRANSMIT MSG QUEUED/
015133     124  040522  051516  EDTXC:  .ASCIZ  /TRANSMIT MSG COMPLETED/
015162  042522  042503  053111  EDRXQ:  .ASCIZ  /RECEIVE SPACE QUEUED/
015207     122  041505  044505  EDRXC:  .ASCIZ  /RECEIVE MSG COMPLETED/
015235     104  053105  041511  EDDER:  .ASCIZ  /DEVICE ERROR/
015252  040504  040524  041440  EDDCK:  .ASCIZ  /DATA COMPARISON STARTED/
015302  042504  044526  042503  EDDVI:  .ASCIZ  /DEVICE INIT AND SETUP/
015330  040504  040524  041440  EDDLE:  .ASCIZ  /DATA COMPARISON LENGTH ERROR/
015365     104  052101  020101  EDDDE:  .ASCIZ  /DATA COMPARISON DATA ERROR/
015420  047105  020104  043117  EDEOP:  .ASCIZ  /END OF PASS/
015434  047515  042504  020115  EDMOS:  .ASCIZ  /MODEM STATUS CHANGE/
```

```
015460  041536  040440  047502  EDABO:  .ASCIZ  /^C ABORT/

                                ;EVENT REPORTING MESSAGES
015471     045  031523  047445  BASM3:  .ASCIZ  /%S3%O3/
015500  051445  022463  033117  BASM2:  .ASCIZ  /%S3%O6/
015507     045  022516  033117  BASM1:  .ASCIZ  /%N%O6/
015515     045  022516  052101  NULEVT: .ASCIZ  /%N%ATHE DCLT EVENT LOG IS EMPTY/
015555     045  022516  037101  EVTF0:  .ASCIZ  /%N%A>>> DCLT EVENT LOG ENTRY <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<</
015651     045  022516  032504  EVTF1:  .ASCIZ  /%N%D5%A:%Z2%A:%Z2%S3%T/
015700  047045  051445  022463  EVTF2:  .ASCIZ  /%N%S3%AADDR OF MSG=%O6%S3%ABYTE COUNT=%D5/
015752  047045  051445  022463  EVTF3:  .ASCIZ  /%N%S3%T%N/
015764  051445  022463  033117  EVTF3C: .ASCIZ  /%S3%O6%S3%O6/
016001     045  031523  047445  EVTF3D: .ASCIZ  /%S3%O6%S3%O6%S3%T/
016023     045  022516  031523  EVTF4:  .ASCIZ  /%N%S3%AADDR OF MSG=%O6%S3%ABYTE COUNT=%D5%S3%ANO. OF CMP ERRS=%D5/
016125     045  022516  031523  EVTF4A: .ASCIZ  /%N%S3%AADDR OF MSG=%O6%S3%ARX BYTES=%D5%S3%ACOMPARE BYTES=%D5/

016223     045  022516  031523  EVTF4B: .ASCIZ  /%N%S3%APASS=%D5%S3%AERRORS=%D5%S3%ASTRT-TO=%D5/

016302  051445  022465  041101  EVTF5A: .ASCIZ  /%S5%ABYTE# IN MSG.=%D5%S3%AEXPTD=%O3%S3%ARECVD=%O3/

016365     045  022516  034523  EVMOCG: .ASCIZ  /%N%S9%ACHANGED TO:/

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

016410  047045  051445  022470  EVMOHD: .ASCIZ  /%N%S8%AMODEM STATUS: CTS DSR DCD RTS RI  SRD/
016465     045  022516  034523  EVMOST: .ASCII  /%N%S9%S9%S5%A/
016502     130     040     040  EVMCTS: .BYTE   'X,40,40,40
016506     130     040     040  EVMDSR: .BYTE   'X,40,40,40
016512     130     040     040  EVMDCD: .BYTE   'X,40,40,40
016516     130     040     040  EVMRTS: .BYTE   'X,40,40,40
016522     130     040     040  EVMRI:  .BYTE   'X,40,40,40
016526     130     040     040  EVMSRD: .BYTE   'X,40,40,40
016532     000                          .BYTE   0
        016534                          .EVEN

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

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

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

```
016747    115  051501  042524  DVEM0:  .ASCII  /MASTER RESET DID NOT WORK/
017000 005015  020040  051040          .ASCIZ  <15><12>/   RXCSR     TXCSR  /
017026 047516  041440  042514  DVEM1:  .ASCII  /NO CLEAR TO SEND FROM MODEM /
017062 005015  020040  051040          .ASCIZ  <15><12>/   RXCSR     TXCSR  /
017110 044524  042515  047440  DVEM2:  .ASCII  /TIME OUT WAITING FOR RX OR TX TO COMPLETE/
017161    015  020012  020040          .ASCIZ  <15><12>/   RXCSR     TXCSR/
017205    103  041522  044440  DVEM3:  .ASCII  /CRC IN ERROR/
017221    015  020012  051040          .ASCIZ  <15><12>/  RXDBUF     RXCSR/
017246 042522  042503  053111  DVEM4:  .ASCII  /RECEIVER OVERRUN/
017266 005015  020040  054122          .ASCIZ  <15><12>/  RXDBUF     RXCSR/
017313    124  046511  042105  DVEM5:  .ASCII  /TIMED OUT IN START,STACK,ACK SEQ/
017353    015  020012  020040          .ASCIZ  <15><12>/   RDATA     SDATA/
017377    115  042117  046505  DVEM6:  .ASCII  /MODEM DID NOT RETURN MODEM READY/
017437    015  020012  020040          .ASCIZ  <15><12>/   RXCSR     TXCSR/

                                .EVEN
```

```
                                .SBTTL  GLOBAL ERROR REPORT SECTION

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

017464                                  BGNMSG  ERR1
017464                                                                                  ERR1::
017464                                  PRINTB  #EVTF5A,OFSET,<B,GOOD>,<B,BAD>  ;INDIVIDUAL DATA COMPARE ERROR
017464  005046                                                                                  CLR     -(SP)
017466  153716  006555                                                                          BISB    BAD,(SP)
017472  005046                                                                                  CLR     -(SP)
017474  153716  006554                                                                          BISB    GOOD,(SP)
017500  013746  006532                                                                          MOV     OFSET,-(SP)
017504  012746  016302                                                                          MOV     #EVTF5A,-(SP)
017510  012746  000004                                                                          MOV     #4,-(SP)
017514  010600                                                                                  MOV     SP,R0
017516  104414                                                                                  TRAP    C$PNTB
017520  062706  000012                                                                          ADD     #12,SP
017524                                  ENDMSG
017524                                                                                  L10001:
017524  104423                                                                                  TRAP    C$MSG

017526                                  BGNMSG  ERR2
017526                                                                                  ERR2::
017526                                  PRINTB  #EFM2,TEMP4                     ;TOTAL DATA COMPARE FAILS ERROR
017526  013746  006544                                                                          MOV     TEMP4,-(SP)
017532  012746  014612                                                                          MOV     #EFM2,-(SP)
017536  012746  000002                                                                          MOV     #2,-(SP)
017542  010600                                                                                  MOV     SP,R0
017544  104414                                                                                  TRAP    C$PNTB
017546  062706  000006                                                                          ADD     #6,SP
017552                                  ENDMSG
017552                                                                                  L10002:
017552  104423                                                                                  TRAP    C$MSG

017554                                  BGNMSG  ERR10
017554                                                                                  ERR10::
017554                                  PRINTB  #EFM11,R4,TEMP3                 ;LENGH COMPARISON ERROR
017554  013746  006542                                                                          MOV     TEMP3,-(SP)
017560  010446                                                                                  MOV     R4,-(SP)
017562  012746  014707                                                                          MOV     #EFM11,-(SP)
017566  012746  000003                                                                          MOV     #3,-(SP)
017572  010600                                                                                  MOV     SP,R0
017574  104414                                                                                  TRAP    C$PNTB
017576  062706  000010                                                                          ADD     #10,SP
017602                                  ENDMSG
017602                                                                                  L10003:
017602  104423                                                                                  TRAP    C$MSG
```

```
017604                          BGNMSG  ERR4
017604                                                          ERR4::
017604                          PRINTB  #EFM13,MHRCNT,MGLCNT            ;MODEM STATUS CHANGE
017604  013746  011516                                                          MOV     MGLCNT,-(SP)
017610  013746  011520                                                          MOV     MHRCNT,-(SP)
017614  012746  015036                                                          MOV     #EFM13,-(SP)
017620  012746  000003                                                          MOV     #3,-(SP)
017624  010600                                                                  MOV     SP,R0
017626  104414                                                                  TRAP    C$PNTB
017630  062706  000010                                                          ADD     #10,SP
017634                          ENDMSG
017634                                                          L10004:
017634  104423                                                                  TRAP    C$MSG



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

017636                          BGNMSG  ERR13
017636                                                          ERR13::
017636                          PRINTB  #EVTF3C,TEMP3,TEMP4
017636  013746  006544                                                          MOV     TEMP4,-(SP)
017642  013746  006542                                                          MOV     TEMP3,-(SP)
017646  012746  015764                                                          MOV     #EVTF3C,-(SP)
017652  012746  000003                                                          MOV     #3,-(SP)
017656  010600                                                                  MOV     SP,R0
017660  104414                                                                  TRAP    C$PNTB
017662  062706  000010                                                          ADD     #10,SP
017666                          ENDMSG
017666                                                          L10005:
017666  104423                                                                  TRAP    C$MSG

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

017670                          BGNMSG  ERR14
017670                                                          ERR14::
017670                          PRINTB  #EVTF3D,TEMP3,TEMP4,CONOTM
017670  013746  006550                                                          MOV     CONOTM,-(SP)
017674  013746  006544                                                          MOV     TEMP4,-(SP)
017700  013746  006542                                                          MOV     TEMP3,-(SP)
017704  012746  016001                                                          MOV     #EVTF3D,-(SP)
017710  012746  000004                                                          MOV     #4,-(SP)
017714  010600                                                                  MOV     SP,R0
017716  104414                                                                  TRAP    C$PNTB
017720  062706  000012                                                          ADD     #12,SP
017724                          ENDMSG
017724                                                          L10006:
017724  104423                                                                  TRAP    C$MSG

017726                          EXIT    MSG
017726  000167                                                                  .WORD   J$JMP
```

```
 3136  017730  177772                                          .WORD   L10006-2-.
 3137
 3138
```

```
                                .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
                                ;--

017732                          CLKSET:
017732  012122                         MOV     (R1)+,(R2)+             ;LOAD CLOCK'S CSR ADDR. INTO "CLKCSR"
017734  012112                         MOV     (R1)+,(R2)              ;LOAD CLOCK'S INT. LEVEL INTO "CLKBR"
017736  006312                         ASL     (R2)                    ;ADJUST THE INT. LEVEL FOR LOADING INTO
017740  006312                         ASL     (R2)                    ;    THE PSW WITH A "SETVEC" CALL
017742  006312                         ASL     (R2)
017744  006312                         ASL     (R2)
017746  006322                         ASL     (R2)+
017750  012122                         MOV     (R1)+,(R2)+             ;LOAD CLOCK'S INT. VECTOR INTO "CLKVEC"
017752  012122                         MOV     (R1)+,(R2)+             ;LOAD CLOCK'S HERTZ RATE INTO "CLKHZ"
017754  000207                         RTS     PC
```

```

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

 017756                                         BGNSRV  CLKINT
 017756                                                                                 CLKINT::

 017756 005077  166640                          CLR     @CLKCSR         ;DISABLE THE CLOCK FORM INTERRUPTING
 017762 005337  006640                          DEC     TIMTCK          ;DECREMENT THE # OF TICKS/SEC.
 017766 001015                                  BNE     1$              ;GO CHECK TIMERS (1&2-TICKS, 3-SECONDS)
 017770 013737  006630  006640                  MOV     CLKHZ,TIMTCK    ;RESET THE # OF TICKS/SEC.
 017776 005237  006636                          INC     TIMSEC          ;INC # OF SECS-SINCE-START
 020002 022737  000074  006636                  CMP     #60.,TIMSEC     ;SEE IF WE'VE COUNTED 60 SECS. YET
 020010 001004                                  BNE     1$              ;IF NOT, GO CHECK TIMERS
 020012 005237  006634                          INC     TIMMIN          ; ELSE INC MINUTES-SINCE-START
 020016 005037  006636                          CLR     TIMSEC          ; AND RESTART SECOND COUNTER

 020022 005737  006642                  1$:     TST     TIMER1          ;SEE IF TIMER #1, TIMING ANYTHING
 020026 001402                                  BEQ     2$              ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
 020030 005337  006642                          DEC     TIMER1          ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
 020034 005737  006644                  2$:     TST     TIMER2          ;SEE IF TIMER #2, TIMING ANYTHING
 020040 001402                                  BEQ     3$              ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
 020042 005337  006644                          DEC     TIMER2          ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
 020046 005737  006646                  3$:     TST     TIMERS          ;SEE IF TIMER #3, TIMING ANYTHING
 020052 001406                                  BEQ     4$              ; IF=0, NOTHING BEING TIMED, LEAVE
 020054 023737  006630  006640                  CMP     CLKHZ,TIMTCK    ;SEE IF A SECOND HAS BEEN COUNTED OFF
 020062 001002                                  BNE     4$              ; BR IF NO
 020064 005337  006646                          DEC     TIMERS          ; ELSE DECREMENT THE TIMER VALUE (BY 1 SEC.)
```

```
 3242  020070  013777  006632  166524  4$:     MOV     CLKEN,@CLKCSR   ;REENABLE THE CLOCK TO INTERRUPT
 3243  020076                                  ENDSRV
 3244  020076                                                                          L10007:
 3245  020076  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
                        ;--

020100                  LOGTXQ:
020100  012737  016537  006536          MOV     #STXQ,TEMP1     ;SET UP MSG. TO PRINT
020106  012737  000000  006534          MOV     #TXQ,TEMP       ;SET UP EVENT TYPE
020114  000517                          BR      LOGS1           ;GO LOG EVENT AND TIME

020116                  LOGTXC:
020116  012737  016550  006536          MOV     #STXC,TEMP1     ;SET UP MSG. TO PRINT
020124  012737  000002  006534          MOV     #TXC,TEMP       ;SET UP EVENT TYPE
020132  000510                          BR      LOGS1           ;GO LOG EVENT AND TIME

020134                  LOGRXQ:
020134  012737  016561  006536          MOV     #SRXQ,TEMP1     ;SET UP MSG. TO PRINT
020142  012737  000004  006534          MOV     #RXQ,TEMP       ;SET UP EVENT TYPE
020150  000501                          BR      LOGS1           ;GO LOG EVENT AND TIME

020152                  LOGRXC:
020152  012737  000006  006534          MOV     #RXC,TEMP       ;SET UP EVENT TYPE
020160  000475                          BR      LOGS1           ;GO LOG EVENT AND TIME
020162                  LGDVE:
```

```
3302  020162  012737  016572  006536          MOV     #SDVE,TEMP1     ;SET UP MSG. TO PRINT
3303  020170  012737  000010  006534          MOV     #DER,TEMP       ;SET UP EVENT TYPE
3304  020176  000503                          BR      LOGS3           ;GO LOG EVENT AND TIME
3305
3306  020200                          LOGDVI:
3307  020200  012737  016614  006536          MOV     #SDVI,TEMP1     ;SET UP MSG. TO PRINT
3308  020206  012737  000012  006534          MOV     #DVI,TEMP       ;SET UP EVENT TYPE
3309  020214  113737  006570  006540          MOVB    MODTYP,TEMP2
3310  020222  113737  006572  006541          MOVB    MLTYP,TEMP2+1
3311  020230  013737  006600  006542          MOV     RPASS,TEMP3
3312  020236  013737  006576  006544          MOV     PARAM,TEMP4     ;SET UP EVNT ENTRIES
3313  020244  000460                          BR      LOGS3           ;GO LOG EVENT AND TIME
3314
3315  020246                          LOGCMP:
3316  020246  012737  016603  006536          MOV     #SCM,TEMP1      ;SET UP MSG. TO PRINT
3317  020254  012737  000014  006534          MOV     #DCK,TEMP       ;SET UP EVENT TYPE
3318  020262  000451                          BR      LOGS3
3319  020264                          LOGCML:
3320  020264  012737  016625  006536          MOV     #SCML,TEMP1
3321  020272  012737  000020  006534          MOV     #DLE,TEMP       ;SET UP MSG. AND TYPE
3322  020300  000442                          BR      LOGS3           ;GO LOG EVENT AND TIME
3323  020302                          LOGCMD:
3324  020302  012737  016636  006536          MOV     #SCMD,TEMP1
3325  020310  012737  000022  006534          MOV     #DDE,TEMP
3326  020316  000433                          BR      LOGS3           ;GO LOG MSG TYPE AND TIME
3327  020320                          LOGEOP:
3328  020320  012737  016647  006536          MOV     #SEOP,TEMP1
3329  020326  012737  000024  006534          MOV     #EOP,TEMP
3330  020334  000424                          BR      LOGS3           ;GO LOG MSG TYPE AND TIME
3331
3332
3333  020336                          LOGMSC:
3334  020336  012737  016660  006536          MOV     #SMSC,TEMP1
3335  020344  012737  000016  006534          MOV     #MSC,TEMP
3336  020352  000415                          BR      LOGS3
3337
3338
3339  020354  013746  006506          LOGS1:  MOV     ERRCNT,-(SP)    ;SAVE CURRENT ERROR COUNT
3340  020360  004737  035152                  JSR     PC,DVMODS       ; GO GET MODEM STATUS
3341  020364  012604                          MOV     (SP)+,R4        ;GET SAVED ERRCNT VALUE
3342  020366  020437  006506                  CMP     R4,ERRCNT       ;WHERE ANY ERRORS FOUND
3343  020372  001402                          BEQ     1$              ; BR IF NONE
3344  020374  000137  020610                  JMP     LOGEX           ; ELSE, LEAVE WITHOUT LOGGING ANYTHING
3345                                                                  ;   BUT THE DEVICE ERROR FROM "DVMODS"
3346  020400  013737  007556  006544  1$:     MOV     MODS,TEMP4      ;AND PUT IT IN TEMP4
3347
3348  020406                          LOGS3:
3349  020406  022737  000006  006534          CMP     #RXC,TEMP
3350  020414  001434                          BEQ     LOGS5           ;IF RXC DONT PRINT
3351  020416  032737  000001  006576          BIT     #STATB,PARAM
3352  020424  001430                          BEQ     LOGS5           ;IF NO STATUS SELECTED
3353                                                                  ;GO TO 5
3354
3355  020426  022737  000010  006500          CMP     #10,LNCNT       ;HAVE WE DONE 10?
3356  020434  001012                          BNE     LOGS4           ;IF NOT GO TO 4
3357  020436  005037  006500                  CLR     LNCNT           ;ESLE CLEAR IT
```

```
3358
3359  020442
3360  020442  012746  016534                PRINTF  #CR              ;ELSE PRINT CR
3361  020446  012746  000001                                                MOV     #CR,-(SP)
3362  020452  010600                                                       MOV     #1,-(SP)
3363  020454  104417                                                       MOV     SP,R0
3364  020456  062706  000004                                               TRAP    C$PNTF
3365  020462                                                                ADD     #4,SP
3366  020462  005237  006500        LOGS4:  INC     LNCNT           ;INC COUNTER OF # OF AWAKE MSGS
3367  020466                                PRINTF  TEMP1           ;PRINT OPERATOR AWAKE MSG.
3368  020466  013746  006536                                                MOV     TEMP1,-(SP)
3369  020472  012746  000001                                                MOV     #1,-(SP)
3370  020476  010600                                                       MOV     SP,R0
3371  020500  104417                                                       TRAP    C$PNTF
3372  020502  062706  000004                                               ADD     #4,SP
3373  020506  010346                LOGS5:  MOV     R3,-(SP)        ;SAVE R3 ON THE STACK
3374  020510  013703  006650                MOV     EVTPTR,R3
3375  020514  113723  006534                MOVB    TEMP,(R3)+      ;LOG EVENT
3376  020520  013737  006630  006534        MOV     CLKHZ,TEMP
3377  020526  163737  006640  006534        SUB     TIMTCK,TEMP
3378  020534  113723  006534                MOVB    TEMP,(R3)+      ;LOG TIME SINCE START
3379  020540  113723  006636                MOVB    TIMSEC,(R3)+
3380  020544  113723  006634                MOVB    TIMMIN,(R3)+    ;TICKS,SECS AND MINS.
3381  020550  013723  006540                MOV     TEMP2,(R3)+     ;LOG EVNT ENTRY 3
3382  020554  013723  006542                MOV     TEMP3,(R3)+     ;LOG EVNT ENTRY 4
3383  020560  013723  006544                MOV     TEMP4,(R3)+     ;LOG EVNT ENTRY 5
3384  020564  020327  007554                CMP     R3,#EVTEND
3385  020570  103404                        BLO     LOGS2           ;IF EVENT LOG FULL GO
3386                                                                ;CONTINUE;ELSE GO TO 2
3387  020572  012713  177777                MOV     #-1,(R3)        ;LOG A TABLE END
3388  020576  012703  006652                MOV     #EVTLOG,R3      ;PUT R3 TO START OF TABLE
3389  020602  010337  006650        LOGS2:  MOV     R3,EVTPTR       ;RESTORE POINTER
3390  020606  012603                        MOV     (SP)+,R3        ;RESTORE R3
3391  020610  000207                LOGEX:  RTS     PC
3392
```

```
                        .SBTTL  REPORT EVENT LOG

                        ;;RPT> LOG
                        ;;     HELP
                        ;;     EXIT
                        ;;     COUNTER/FULL,ERROR,OFFSET=NN(O)

020612  010246          REPORT: MOV     R2,-(SP)        ;SAVE R2,R3,R4 ON THE STACK
020614  010346                  MOV     R3,-(SP)
020616  010446                  MOV     R4,-(SP)

                        ;PRINT HELP MESSAGE
020620                  PRINTF  #RHLP0          ;BASIC HELP MESSAGE
020620  012746  013075                                    MOV     #RHLP0,-(SP)
020624  012746  000001                                    MOV     #1,-(SP)
020630  010600                                            MOV     SP,R0
020632  104417                                            TRAP    C$PNTF
020634  062706  000004                                    ADD     #4,SP

020640  105037  003147  GETRCL: CLRB    P$GDBD          ;INIT GOOD/BAD FLAG  -1=BAD INPUT
020644  105037  003146          CLRB    P$NNUF          ;INIT MORE COMMAND LINE INPUT NEEDED

                        ;PRINT PROMPT 'RPT>'
020650                  GMANID  CLI$RP,CMDBUF,A,-1,1,72.,NO
020650  104443                                            TRAP    C$GMAN
020652  000406                                            BR      10000$
020654  002666                                            .WORD   CMDBUF
020656  000142                                            .WORD   T$CODE
020660  011576                                            .WORD   CLI$RP
020662  177777                                            .WORD   -1
020664  000001                                            .WORD   T$LOLIM
020666  000110                                            .WORD   T$HILIM
020670                                                  10000$:
020670  012737  002666  003132          MOV     #CMDBUF,P$BUFA  ;INPUT BUFFER
020676  012737  021032  003134          MOV     #CLIRT,P$TREE   ;REPORT CLI TREE
020704  012737  021234  003136          MOV     #CLIRAC,P$ACT   ;ACTION ROUTINES
020712  005037  003012                  CLR     QUALFG
020716  004737  024246                  JSR     PC,P$TRV        ;GO PARSE COMMAND LINE
020722  105737  003147                  TSTB    P$GDBD          ;COMMAND OK ?
020726  001412                          BEQ     1$              ;YES,BRANCH
020730                                  PRINTF  #CLIERM         ;PRINT INVALID INPUT MESSAGE
020730  012746  011603                                            MOV     #CLIERM,-(SP)
020734  012746  000001                                            MOV     #1,-(SP)
020740  010600                                                    MOV     SP,R0
020742  104417                                                    TRAP    C$PNTF
020744  062706  000004                                            ADD     #4,SP
020750  000137  020640                  JMP     GETRCL          ;TRY AGAIN

020754  105737  003146          1$:     TSTB    P$NNUF          ;MORE COMMAND NEEDED ?
020760  001412                          BEQ     10$             ;NO,BRANCH
020762                                  PRINTF  #CLINUF         ;INCOMPLETE MESSAGE
020762  012746  011633                                            MOV     #CLINUF,-(SP)
020766  012746  000001                                            MOV     #1,-(SP)
020772  010600                                                    MOV     SP,R0
020774  104417                                                    TRAP    C$PNTF
020776  062706  000004                                            ADD     #4,SP
```

```
 3449  021002  000137  020640                    JMP     GETRCL                  ;TRY AGAIN
 3450
 3451  021006  023727  003010  000002  10$:      CMP     KEYWD1,#RPEXT           ;EXIT COMMAND ?
 3452  021014  001402                            BEQ     20$                     ;YES,BRANCH
 3453  021016  000137  020640                    JMP     GETRCL                  ;GET ANOTHER COMMAND
 3454  021022  012604                  20$:      MOV     (SP)+,R4                ;RESTORE R4
 3455  021024  012603                            MOV     (SP)+,R3                ;RESTORE R3
 3456  021026  012602                            MOV     (SP)+,R2                ;RESTORE R2
 3457  021030  000207                            RTS     PC                      ;RETURN
```

```
 3458
 3459  021032                .SBTTL  COMMAND LINE PARSING TREE FOR REPORT
 3460  021036  CLIRT:   CLI CLISPA,0,R10$                ;SKIP SPACES IN COMMAND LINE
 3461  021042  R10$:    CLI <'?>,RPHLP,R11$             ;IF INPUT = ? THEN PRINT HELP MESSAGE
 3462  021044           CLI CLIEXI,0                    ;;AND EXIT PARSER
 3463  021060  R11$:    CLI CLISTR,RPHLP,R12$,<'HELP'>  ;IF INPUT = 'HELP' THEN PRINT HELP
 3464  021062           CLI CLIEXI,0                    ;;MESSAGE AND EXIT PARSER
 3465  021076  R12$:    CLI CLISTR,RPEXT,R13$,<'EXIT'>  ;IF INPUT = 'EXIT' THEN SET KEYWORD =
 3466  021100           CLI CLIEXI,0                    ;;RPEXT AND EXIT PARSER
 3467  021112  R13$:    CLI CLISTR,RPLOG,R14$,<'LOG'>   ;IF INPUT = 'LOG' THEN GO PRINT EVENT
 3468  021114           CLI CLIEXI,0                    ;;LOG AND EXIT PARSER
 3469  021134  R14$:    CLI CLISTR,RNOTNF,R30$,<'COUNTERS'>;IF INPUT = 'COUNTERS'
 3470  021140           CLI CLIBR,0,R20$                ;;THEN GET SWITCH
 3471  021144  R20$:    CLI <'/>,RNOTNF,R30$
 3472  021160           CLI CLISTR,RPERR,R21$,<'ERROR'> ; REPORT ERROR COUNTERS
 3473  021162           CLI CLIEXI,0
 3474  021176  R21$:    CLI CLISTR,RPFUL,R22$,<'FULL'>  ; REPORT ALL STATUS
 3475  021200           CLI CLIEXI,0
 3476  021216  R22$:    CLI CLISTR,RNOTNF,R30$,<'OFFSET'> ; REPORT ONE LOCATION
 3477  021222           CLI <'=>,0,R30$
 3478  021226           CLI CLIOCT,RPSWO,R30$
 3479  021230           CLI CLIEXI,0
 3480  021232  R30$:    CLI CLIERR,0
                R125$:   CLI CLIEXI,0
```

```
3481
3482  021234  006302                  .SBTTL  CLI ACTION DISPATCHER AND ROUTINES
3483  021236  016202  021252          CLIRAC: ASL     R2              ;SET UP INDEX
3484  021242  062702  021252                  MOV     10$(R2),R2      ;
3485  021246  004712                          ADD     #10$,R2         ;
3486  021250  000207                          JSR     PC,(R2)         ;GO DO ACTION
3487  021252  000026                          RTS     PC              ;RETURN
3488  021254  000030                  10$:    .WORD   ACTRNL-10$      ;NULL
3489  021256  000074                          .WORD   ACTRHL-10$      ;HELP ROUTINE
3490  021260  000104                          .WORD   ACTREX-10$      ;EXIT ROUTINE
3491  021262  000142                          .WORD   ACTRLG-10$      ;REPORT EVENT LOG ROUTINE
3492  021264  000120                          .WORD   ACTERR-10$      ;REPORT ONLY ERROR COUNTERS
3493  021266  000020                          .WORD   ACTFUL-10$      ;REPORT ALL COUNTERS
3494  021270  000164                          .WORD   ACTRNF-10$      ;MORE COMMAND NEEDED
3495                                          .WORD   ACTRSO-10$      ;VALIDATE OFFSET
3496
3497  021272  113737  177777  003146  ;;;;;ACTION ROUTINES FOR REPORT;;;;;
3498  021300  000207  ACTRNF: MOVB    -1,P$NNUF       ;SET'MORE COMMAND NEEDED' FLAG
3499                  ACTRNL: RTS     PC              ;NULL
3500
3501  021302  012702  003036          ;PRINT HELP MESSAGE
3502  021306          ACTRHL: MOV     #RHLPTB,R2      ;INDEX FOR HELP MESSAGES
3503  021306  012246  1$:     PRINTF  #HLPF,(R2)+     ;PRINT IT
3504  021310  012746  012271                                          MOV     (R2)+,-(SP)
3505  021314  012746  000002                                          MOV     #HLPF,-(SP)
3506  021320  010600                                                  MOV     #2,-(SP)
3507  021322  104417                                                  MOV     SP,R0
3508  021324  062706  000006                                          TRAP    C$PNTF
3509  021330  020227  003052                                          ADD     #6,SP
3510  021334  001364                  CMP     R2,#RHLPEN      ;LAST MESSAGE ?
3511  021336  012737  000001  003010  BNE     1$              ;NO,BRANCH
3512  021344  000207                  MOV     #RPHLP,KEYWD1   ;SET KEYWORD
3513                                  RTS     PC              ;RETURN
3514
3515                  ;EXIT REPORT LEVEL
3516  021346  012737  000002  003010  ACTREX: MOV     #RPEXT,KEYWD1   ;SET KEYWORD AND RETURN
3517  021354  000207                  RTS     PC
3518
3519                  ;PRINT EVENT LOG
3520  021356  004737  022060          ACTRLG: JSR     PC,REPLOG       ;GO PRINT EVENT LOG
3521  021362  012737  000003  003010  MOV     #RPLOG,KEYWD1   ;SET KEYWORD
3522  021370  000207                  RTS     PC              ;RETURN
3523
3524                  ;;REPORT ALL MESSAGE AND ERROR COUNTERS
3525  021372  012737  000000  037254  ACTFUL: MOV     #0,FIR          ;STARTING INDEX
3526  021400  012737  000036  037252  MOV     #36,LAST        ;LAST INDEX
3527  021406  004737  021542          JSR     PC,STAPRI       ;GO PRINT IT
3528  021412  000207                  RETURN
3529
3530                  ;;PRINT ONLY DDCMP ERROR COUNTERS
3531  021414  012737  000014  037254  ACTERR: MOV     #14,FIR         ;FIRST ERROR
3532  021422  012737  000036  037252  MOV     #36,LAST        ;LAST ERROR
3533  021430  004737  021542          JSR     PC,STAPRI       ;GO PRINT IT
3534  021434  000207                  RETURN
3535
3536
```

```
3537                                    ;;VERIFY OFFSET VALUE
3538  021436  105037  003146  ACTRSO: CLRB    P$NNUF          ;CLEAR 'NOT ENOUGH FLAG'
3539  021442  032737  000001  003142         BIT     #BIT0,P$NUM     ;IS IT ODD ?
3540  021450  001020                         BNE     20$             ;YES,BRANCH
3541  021452  005737  003142                 TST     P$NUM           ;NEGATIVE # ?
3542  021456  100415                         BMI     20$             ;YES,BRANCH
3543  021460  023727  003142  000036         CMP     P$NUM,#36       ;INDEX LARGER THEN 36 ?
3544  021466  003011                         BGT     20$             ;YES,BRANCH
3545  021470  013737  003142  037254         MOV     P$NUM,FIR       ;STARTING INDEX
3546  021476  013737  003142  037252         MOV     P$NUM,LAST      ;LAST LOCATION
3547  021504  004737  021542                 JSR     PC,STAPRI       ;PRINT SINGLE LOCATION
3548  021510  000413                         BR      30$             ;EXIT
3549  021512                          20$:    PRINTS  #RPTIV          ;INVALID
3550  021512  012746  013405                                                    MOV     #RPTIV,-(SP)
3551  021516  012746  000001                                                    MOV     #1,-(SP)
3552  021522  010600                                                            MOV     SP,R0
3553  021524  104416                                                            TRAP    C$PNTS
3554  021526  062706  000004                                                    ADD     #4,SP
3555  021532  112737  177777  003147         MOVB    #-1,P$GDBD      ;SET BAD DATA FLAG
3556  021540  000207                  30$:    RETURN                  ;OFFSET OK - EXIT
3557
3558
3559                                    ;; PRINT ROUTINES
3560  021542  010146                  STAPRI: MOV     R1,-(SP)        ;SAVE R1
3561  021544  032737  000040  006576         BIT     #PROTOB,PARAM   ;"/PROTOCOL"?
3562  021552  001011                         BNE     5$              ;YES,BRANCH
3563  021554                                 PRINTF  #RPTNV          ;"COUNTERS VALID ONLY WITH PROTOCOL SELECTED"
3564  021554  012746  013430                                                    MOV     #RPTNV,-(SP)
3565  021560  012746  000001                                                    MOV     #1,-(SP)
3566  021564  010600                                                            MOV     SP,R0
3567  021566  104417                                                            TRAP    C$PNTF
3568  021570  062706  000004                                                    ADD     #4,SP
3569  021574  000420                         BR      20$             ;EXIT
3570  021576  013701  037254          5$:    MOV     FIR,R1  ;FIRST INDEX
3571  021602  016137  037152  037256 10$:    MOV     STALST(R1),MES  ;MESSAGE ADDRESS
3572  021610  016137  037052  037260         MOV     PRSTAT(R1),MESDATA ;MESSAGE DATA
3573  021616  004771  037212                 JSR     PC,@STAIND(R1)  ;JUMP TO PROPER PRINT ROUTINE
3574  021622  062701  000002                 ADD     #2,R1           ;BUMP INDEX
3575  021626  020137  037252                 CMP     R1,LAST         ;ALL MESSAGES PRINTED
3576  021632  003001                         BGT     20$             ;YES,BRANCH
3577  021634  000762                         BR      10$             ;PRINT NEXT MESSAGE
3578  021636  012601                  20$:    MOV     (SP)+,R1        ;RESTORE R1
3579  021640  000207                         RETURN                  ;EXIT
3580
3581
3582                                    ;; PRINT WORD LOCATION
3583  021642                          PRIW:   PRINTS  MES,MESDATA     ;PRINT WORD LOCATION
3584  021642  013746  037260                                                    MOV     MESDATA,-(SP)
3585  021646  013746  037256                                                    MOV     MES,-(SP)
3586  021652  012746  000002                                                    MOV     #2,-(SP)
3587  021656  010600                                                            MOV     SP,R0
3588  021660  104416                                                            TRAP    C$PNTS
3589  021662  062706  000006                                                    ADD     #6,SP
3590  021666  000207                         RETURN
3591
3592
```

```
3593
3594  021670
3595  021670  005046                    ;; PRINT TWO BYTES OF DATA
3596  021672  153716  037261            PRIBB:  PRINTS  MES,<B,MESDATA>,<B,MESDATA+1>
3597  021676  005046                                    CLR     -(SP)
3598  021700  153716  037260                            BISB    MESDATA+1,(SP)
3599  021704  013746  037256                            CLR     -(SP)
3600  021710  012746  000003                            BISB    MESDATA,(SP)
3601  021714  010600                                    MOV     MES,-(SP)
3602  021716  104416                                    MOV     #3,-(SP)
3603  021720  062706  000010                            MOV     SP,R0
3604  021724  000207                                    TRAP    C$PNTS
3605                                                    ADD     #10,SP
3606                                  RETURN
3607  021726  005037  006536            ;; PRINT SPECIAL BYTE MASK
3608  021732  005037  006540            PRIBS:  CLR     TEMP1
3609  021736  005037  006542                    CLR     TEMP2
3610  021742  132737  000001  037261            CLR     TEMP3
3611  021750  001402                            BITB    #BIT0,MESDATA+1 ;BIT 0 = 1 ?
3612  021752  005237  006536                    BEQ     10$             ;NO,BRANCH
3613  021756  132737  000002  037261            INC     TEMP1           ;SET IT
3614  021764  001402                    10$:    BITB    #BIT1,MESDATA+1 ;BIT 1 =1 ?
3615  021766  005237  006540                    BEQ     20$             ;NO,BRANCH
3616  021772  132737  000004  037261            INC     TEMP2           ;SET IT
3617  022000  001402                    20$:    BITB    #BIT2,MESDATA+1 ;BIT2 = 1 ?
3618  022002  005237  006542                    BEQ     30$             ;NO,BRANCH
3619  022006                                    INC     TEMP3           ;SET IT
3620  022006  005046                    30$:    PRINTS  MES,<B,MESDATA>,<B,TEMP1>,<B,TEMP2>,<B,TEMP3>
3621  022010  153716  006542                                    CLR     -(SP)
3622  022014  005046                                            BISB    TEMP3,(SP)
3623  022016  153716  006540                                    CLR     -(SP)
3624  022022  005046                                            BISB    TEMP2,(SP)
3625  022024  153716  006536                                    CLR     -(SP)
3626  022030  005046                                            BISB    TEMP1,(SP)
3627  022032  153716  037260                                    CLR     -(SP)
3628  022036  013746  037256                                    BISB    MESDATA,(SP)
3629  022042  012746  000005                                    MOV     MES,-(SP)
3630  022046  010600                                            MOV     #5,-(SP)
3631  022050  104416                                            MOV     SP,R0
3632  022052  062706  000014                                    TRAP    C$PNTS
3633  022056  000207                                            ADD     #14,SP
                                  RETURN
3634
```

```
3635                                    .SBTTL          DUMP EVENT LOG
3636
3637
3638  022060  010246                    REPLOG: MOV     R2,-(SP)        ;SAVE R2,R3,R4 ON THE STACK
3639  022062  010346                            MOV     R3,-(SP)
3640  022064  010446                            MOV     R4,-(SP)
3641
3642
3643
3644
3645  022066  013702  006650                    MOV     EVTPTR,R2       ;MAKE R2 A POINTER TO EVENT TABLE
3646  022072  023727  006652  177777            CMP     EVTLOG,#-1      ;SEE IF EVENT TABLE IS EMPTY
3647  022100  001034                            BNE     RPT0            ;BR IF NO
3648  022102                                    PRINTS  #NULEVT         ;IF EMPTY TELL OPRERATOR.
3649  022102  012746  015515                                                    MOV     #NULEVT,-(SP)
3650  022106  012746  000001                                                    MOV     #1,-(SP)
3651  022112  010600                                                            MOV     SP,R0
3652  022114  104416                                                            TRAP    C$PNTS
3653  022116  062706  000004                                                    ADD     #4,SP
3654  022122  000137  023006                    JMP     ENDEVT          ;AND END
3655
3656  022126  162702  000012          RPT:      SUB     #12,R2          ;NOW POINT BACK TO TOP OF ENTRY U
3657                                                                    ;JUST PRINTED
3658
3659  022132  020227  006652                    CMP     R2,#EVTLOG      ;POINTING TO TOP OF EVNT LOG QUEUE?
3660  022136  001010                            BNE     RPT1            ; BR IF NO
3661  022140  012702  007554                    MOV     #EVTEND,R2      ;SET R2 TO POINT TO BOTTOM OF LOG
3662  022144  026227  177776  177777            CMP     -2(R2),#-1
3663  022152  001007                            BNE     RPT0            ;IF END OF LOG IS NOT EMPTY
3664  022154  000137  023006                    JMP     ENDEVT          ;CONTINUE...ELSE EXIT
3665
3666  022160  020237  006650          RPT1:     CMP     R2,EVTPTR       ;ARE WE BACK TO POINTER?
3667  022164  001002                            BNE     RPT0            ;IF NOT CONTINUE
3668  022166  000137  023006                    JMP     ENDEVT          ;IF SO EXIT....
3669
3670  022172  162702  000012          RPT0:     SUB     #12,R2          ;POINT R2 TO START OF ENTRY
3671  022176                          RPTAA:    PRINTS  #EVTF0          ;PRINT EVENT ENTRY HEADER
3672  022176  012746  015555                                                    MOV     #EVTF0,-(SP)
3673  022202  012746  000001                                                    MOV     #1,-(SP)
3674  022206  010600                                                            MOV     SP,R0
3675  022210  104416                                                            TRAP    C$PNTS
3676  022212  062706  000004                                                    ADD     #4,SP
3677  022216  112203                            MOVB    (R2)+,R3        ;PUT EVENT TYPE INTO R3
3678  022220  112237  007644                    MOVB    (R2)+,EVTTCK
3679  022224  112237  007640                    MOVB    (R2)+,EVTSEC    ;PUT EVENT TIME (TICKS,SECS,MINS IN TEMP LOC.S)
3680  022230  112237  007642                    MOVB    (R2)+,EVTMIN
3681  022234                                    PRINTS  #EVTF1,EVTMIN,EVTSEC,EVTTCK,EVTLST(R3)  ;PRINT EVENT TIME AND DESCRIPT.
3682  022234  016346  007610                                                    MOV     EVTLST(R3),-(SP)
3683  022240  013746  007644                                                    MOV     EVTTCK,-(SP)
3684  022244  013746  007640                                                    MOV     EVTSEC,-(SP)
3685  022250  013746  007642                                                    MOV     EVTMIN,-(SP)
3686  022254  012746  015651                                                    MOV     #EVTF1,-(SP)
3687  022260  012746  000005                                                    MOV     #5,-(SP)
3688  022264  010600                                                            MOV     SP,R0
3689  022266  104416                                                            TRAP    C$PNTS
3690  022270  062706  000014                                                    ADD     #14,SP
```

```
3691  022274  000173  007654                  JMP     @RPTDSP(R3)     ;DISPATCH TO DECODING SECTION FOR SPECIFIC TYPE
3692
3693  022300  012237  007646  RPTTXQ: MOV     (R2)+,EVTADD    ;STORE MESSAGE ADDRESS FOR PRINTING
3694  022304  012237  007650          MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT FOR PRINTING
3695  022310  012203                  MOV     (R2)+,R3        ;STORE MODEM STATUS FOR PRINTING
3696  022312                          PRINTS  #EVTF2,EVTADD,EVTBCT    ;PRINT ADDR,BYTE CNT
3697  022312  013746  007650                                                  MOV     EVTBCT,-(SP)
3698  022316  013746  007646                                                  MOV     EVTADD,-(SP)
3699  022322  012746  015700                                                  MOV     #EVTF2,-(SP)
3700  022326  012746  000003                                                  MOV     #3,-(SP)
3701  022332  010600                                                          MOV     SP,R0
3702  022334  104416                                                          TRAP    C$PNTS
3703  022336  062706  000010                                                  ADD     #10,SP
3704  022342  004737  023016                  JSR     PC,RPTMSB       ;GO PRINT MODEM STATUS
3705  022346  000137  022126                  JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
3706
3707  022352  012237  007652  RPTDER: MOV     (R2)+,EVTTMP    ;GET ADDRESS OF DEVICE INFO MESSAGE
3708  022356  012237  007704          MOV     (R2)+,DEV1      ;STORE DEVICE REG CONTENTS FOR PRINTING
3709  022362  012237  007706          MOV     (R2)+,DEV2
3710  022366                          PRINTS  #EVTF3,EVTTMP   ;PRINT DEVICE REG CONTENTS.
3711  022366  013746  007652                                                  MOV     EVTTMP,-(SP)
3712  022372  012746  015752                                                  MOV     #EVTF3,-(SP)
3713  022376  012746  000002                                                  MOV     #2,-(SP)
3714  022402  010600                                                          MOV     SP,R0
3715  022404  104416                                                          TRAP    C$PNTS
3716  022406  062706  000006                                                  ADD     #6,SP
3717  022412                          PRINTS  #EVTF3C,DEV1,DEV2
3718  022412  013746  007706                                                  MOV     DEV2,-(SP)
3719  022416  013746  007704                                                  MOV     DEV1,-(SP)
3720  022422  012746  015764                                                  MOV     #EVTF3C,-(SP)
3721  022426  012746  000003                                                  MOV     #3,-(SP)
3722  022432  010600                                                          MOV     SP,R0
3723  022434  104416                                                          TRAP    C$PNTS
3724  022436  062706  000010                                                  ADD     #10,SP
3725  022442  000137  022126                  JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
3726
3727  022446  005037  007704  RPTDVI: CLR     DEV1
3728  022452  005037  007706          CLR     DEV2            ;CLEAR UPPER BYTES OF DEV1 & DEV2 BEFORE USE
3729  022456  112237  007704          MOVB    (R2)+,DEV1      ;STORE SETUP OPERATION PARAMETERS FOR PRINTING
3730  022462  112237  007706          MOVB    (R2)+,DEV2
3731  022466  012237  007710          MOV     (R2)+,DEV3
3732  022472  012237  007712          MOV     (R2)+,DEV4
3733  022476  010246                  MOV     R2,-(SP)        ;SAVE R2 ON THE STACK
3734  022500  004737  023714          JSR     PC,SHWOP        ;GO PRINT MODE, MAINT-LOOP TYPE, PARAMTERS.
3735  022504  012602                  MOV     (SP)+,R2        ;RESTORE R2
3736  022506  000137  022126          JMP     RPT             ;GO BACK FOR NEXT EVENT ENTRY
3737
3738
3739  022512          ;;REPORT END OF PASS OR ^C ABORT
                      RPTABO:
3740  022512  012237  007646  RPTEOP: MOV     (R2)+,EVTADD    ;PASSES
3741  022516  012237  007650          MOV     (R2)+,EVTBCT    ;ERRORS
3742  022522  012237  007652          MOV     (R2)+,EVTTMP    ;START TIME OUTS
3743  022526                          PRINTS  #EVTF4B,EVTADD,EVTBCT,EVTTMP    ;PRINT ADDR,RXBYTES,CMPBYTES.
3744  022526  013746  007652                                                  MOV     EVTTMP,-(SP)
3745  022532  013746  007650                                                  MOV     EVTBCT,-(SP)
3746  022536  013746  007646                                                  MOV     EVTADD,-(SP)
```

```
                                                                                        MOV     #EVTF4B,-(SP)
                                                                                        MOV     #4,-(SP)
                                                                                        MOV     SP,R0
                                                                                        TRAP    C$PNTS
                                                                                        ADD     #12,SP
                        JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY

RPTDDE:                 MOV     (R2)+,EVTADD    ;STORE MESSAGE ADDRESS FOR PRINTING
                        MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT FOR PRINTING
                        MOV     (R2)+,EVTTMP    ;STORE TOTAL # OF CMP ERRORS
                        PRINTS  #EVTF4,EVTADD,EVTBCT,EVTTMP     ;PRINT ADDR, BYTE CNT, # CMP ERRS
                                                                                        MOV     EVTTMP,-(SP)
                                                                                        MOV     EVTBCT,-(SP)
                                                                                        MOV     EVTADD,-(SP)
                                                                                        MOV     #EVTF4,-(SP)
                                                                                        MOV     #4,-(SP)
                                                                                        MOV     SP,R0
                                                                                        TRAP    C$PNTS
                                                                                        ADD     #12,SP
                        JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY

RPTDLE:
RPTDCK:                 MOV     (R2)+,EVTADD    ;STORE MSG ADDR FOR PRINT
                        MOV     (R2)+,EVTBCT    ;STORE BYTE COUNT
                        MOV     (R2)+,EVTTMP    ;STORE BYTE COUNT COMP
                        PRINTS  #EVTF4A,EVTADD,EVTBCT,EVTTMP    ;PRINT ADDR,RXBYTES,CMPBYTES.
                                                                                        MOV     EVTTMP,-(SP)
                                                                                        MOV     EVTBCT,-(SP)
                                                                                        MOV     EVTADD,-(SP)
                                                                                        MOV     #EVTF4A,-(SP)
                                                                                        MOV     #4,-(SP)
                                                                                        MOV     SP,R0
                                                                                        TRAP    C$PNTS
                                                                                        ADD     #12,SP

                        JMP     RPT             ;THEN GO GET NEXT EVENT ENTRY




RPTMSC:                 MOV     (R2)+,EVTTMP
                        PRINTS  #EVTF3,EVTTMP   ;PRINT CHANGE TYPE
                                                                                        MOV     EVTTMP,-(SP)
                                                                                        MOV     #EVTF3,-(SP)
                                                                                        MOV     #2,-(SP)
                                                                                        MOV     SP,R0
                                                                                        TRAP    C$PNTS
                                                                                        ADD     #6,SP
                        MOV     (R2)+,R3        ;PUT OLD MODEM STATUS IN R3 FOR PRINTING
                        JSR     PC,RPTMSB       ;GO PRINT OLD MODEM STATUS
                        PRINTS  #EVMOCG         ;GO PRINT "CHANGED TO:"
                                                                                        MOV     #EVMOCG,-(SP)
                                                                                        MOV     #1,-(SP)
                                                                                        MOV     SP,R0
                                                                                        TRAP    C$PNTS
```

```
3803  022770  062706  000004                                                  ADD     #4,SP
3804  022774  012203                  MOV     (R2)+,R3        ;PUT NEW MODEM STATUS IN R3 FOR PRINTING
3805  022776  004737  023016          JSR     PC,RPTMSB       ;GO PRINT NEW MODEM STATUS
3806  023002  000137  022126  RPTMSE: JMP     RPT             ;THEN GO GET NEXT EVENT
3807
3808
3809  023006  012604          ENDEVT: MOV     (SP)+,R4        ;RESTORE R4,R3,R2
3810  023010  012603                  MOV     (SP)+,R3
3811  023012  012602                  MOV     (SP)+,R2
3812  023014  000207                  RTS     PC              ;RETURN TO CALLING ROUTINE
3813
3814
3815                          ;REPORT MODEM STATUS SUBROUTINE
3816                          ;       PART OF STATISICAL REPORTING (DUMPING EVENT LOG)
3817
3818  023016          RPTMSB: PRINTS  #EVMOHD                 ;PRINT MODEM STATUS HEADER
3819  023016  012746  016410                                                  MOV     #EVMOHD,-(SP)
3820  023022  012746  000001                                                  MOV     #1,-(SP)
3821  023026  010600                                                          MOV     SP,R0
3822  023030  104416                                                          TRAP    C$PNTS
3823  023032  062706  000004                                                  ADD     #4,SP
3824  023036  012704  007560          MOV     #MOBITS,R4      ;MAKE R4 A POINTER TO MODEM SIG. BIT DEF. TABLE
3825  023042  012705  007574          MOV     #MOMSGS,R5      ;MAKE R5 A POINTER TO MODEM MSG. POSITION TABLE
3826  023046  005714          6$:     TST     (R4)            ;SEE IF BIT AVAIABLE FROM DEVICE
3827  023050  001004                  BNE     7$              ;BR IF THAT MODEM SIG. AVAIABLE
3828  023052  112735  000130          MOVB    #'X,@(R5)+      ;ELSE PUT 'X" IN REPORT IF SIGNAL NOT AVAILABLE
3829  023056  005724                  TST     (R4)+           ;BUMP R4 TO POINT TO NEXT BIT DEFINITION
3830  023060  000407                  BR      9$              ;GO SEE IF CHECKED ALL MODEM SIGNALS
3831  023062  032403          7$:     BIT     (R4)+,R3        ;IF THERE, SEE IF THAT BIT IN DEVICE'S ENTRY=1
3832  023064  001403                  BEQ     8$              ;BR IF BIT (SIGNAL) VALUE =0
3833  023066  112735  000061          MOVB    #'1,@(R5)+      ;IF=1, PUT "1" IN REPORT MESSAGE
3834  023072  000402                  BR      9$              ;GO SEE IF ALL MODEM SIGNALS CHECKED
3835  023074  112735  000060  8$:     MOVB    #'0,@(R5)+      ;IF BIT(SIGNAL)=0, PUT "0" IN REPORT MESSAGE
3836  023100  020427  007574  9$:     CMP     R4,#MOBITE      ;SEE IF ALL BITS(SIGNALS) CHECKED
3837  023104  002760                  BLT     6$              ;LOOP UNTIL ALL SIGNALS(BITS) CHECKED
3838  023106                          PRINTS  #EVMOST         ;THEN PRINT MODEM SIGNAL VALUE MESSAGE
3839  023106  012746  016465                                                  MOV     #EVMOST,-(SP)
3840  023112  012746  000001                                                  MOV     #1,-(SP)
3841  023116  010600                                                          MOV     SP,R0
3842  023120  104416                                                          TRAP    C$PNTS
3843  023122  062706  000004                                                  ADD     #4,SP
3844  023126  000207                  RTS     PC              ;RETURN TO EVENT DECODING
3845
3846
```

```
                        .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
                        ;
                        ;--

023130  013702  006510  DUMPSR: MOV     STADD,R2        ;SET R2 UP TO STARTING ADDR.
023134  005003          DUM4:   CLR     R3              ;CLEAR R3
023136                          PRINTF  #BASM1,R2       ;PRINT ADDRESS
023136  010246                                                  MOV     R2,-(SP)
023140  012746  015507                                          MOV     #BASM1,-(SP)
023144  012746  000002                                          MOV     #2,-(SP)
023150  010600                                                  MOV     SP,R0
023152  104417                                                  TRAP    C$PNTF
023154  062706  000006                                          ADD     #6,SP
023160  005737  006514  DUM3:   TST     BYTBIT          ;IS THIS BYTE OR WORD
023164  001416                  BEQ     DUM1            ;BR IF WORD
023166  112237  006534          MOVB    (R2)+,TEMP      ;MOV BYTE TO TEMP
023172                          PRINTF  #BASM3,<B,TEMP> ;PRINT BYTE
023172  005046                                                  CLR     -(SP)
023174  153716  006534                                          BISB    TEMP,(SP)
023200  012746  015471                                          MOV     #BASM3,-(SP)
023204  012746  000002                                          MOV     #2,-(SP)
023210  010600                                                  MOV     SP,R0
023212  104417                                                  TRAP    C$PNTF
023214  062706  000006                                          ADD     #6,SP
023220  000411                  BR      DUM2
023222          DUM1:   PRINTF  #BASM2,(R2)+    ;PRINT WORD
023222  012246                                                  MOV     (R2)+,-(SP)
023224  012746  015500                                          MOV     #BASM2,-(SP)
023230  012746  000002                                          MOV     #2,-(SP)
023234  010600                                                  MOV     SP,R0
023236  104417                                                  TRAP    C$PNTF
023240  062706  000006                                          ADD     #6,SP
023244  020237  006512  DUM2:   CMP     R2,ENADD        ;COMPARE FOR LAST ADD
```

```
 3903  023250  003005                    BGT     DUMEX           ;IF DONE EXIT
 3904  023252  005203                    INC     R3              ;ELSE BUMP R3
 3905  023254  022703  000010            CMP     #8.,R3          ;HAVE WE PRINTED 8 ACCROSS
 3906  023260  001725                    BEQ     DUM4            ;IF SO GO BACK TO 4
 3907  023262  000736                    BR      DUM3            ;ELSE GO BACK AND PRINT ANOTHER
 3908                                                            ;BYTE OR WORD
 3909  023264  000207            DUMEX:  RTS     PC              ;RETURN TO CALLER
 3910
```

```
                                .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
                                ;--

023266  063737  006520  006530  ADDCC:  ADD     CURCC,TOTCC     ;ADD CURRENT TO TOTAL
023274  022737  001000  006530           CMP     #BUFLIM,TOTCC   ; COMPARE TO 'BUFLIM'
023302  103027                           BHIS    ADDC1           ;IF NOT MORE THEN 'BUFLIM' EXIT

                                        ; PRINT MESSAGE AND TRUNCATE COUNT

023304                                  PRINTF  #MSGTRU
023304  012746  014424                                                  MOV     #MSGTRU,-(SP)
023310  012746  000001                                                  MOV     #1,-(SP)
023314  010600                                                          MOV     SP,R0
023316  104417                                                          TRAP    C$PNTF
023320  062706  000004                                                  ADD     #4,SP
023324  163737  006520  006530           SUB     CURCC,TOTCC     ;SUB CURRENT FROM TOTAL
023332  012737  001000  006534           MOV     #BUFLIM,TEMP    ;MOV 'BUFLIM' TO TEMP
023340  163737  006530  006534           SUB     TOTCC,TEMP      ;SUB TOTAL FROM 'BUFLIM'
023346  013737  006534  006520           MOV     TEMP,CURCC      ;AND ESTABLISH NEW CURRENT
023354  063737  006520  006530           ADD     CURCC,TOTCC     ;ADD 'ADJUSTED CURRENT' TO TOTAL CHAR. CNT.
023362  000207                  ADDC1:   RTS     PC              ;RETURN TO CALLER
```

```
                                .SBTTL          BUILD MESSAGE BUFFERS SUBROUTINE

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

023364                          BLDBUF:
023364  010246                          MOV     R2,-(SP)        ;SAVE R2 AND R3 ON THE STACK
023366  010346                          MOV     R3,-(SP)
023370  013702  006524                  MOV     CPTR,R2

023374  013722  006526         BLDB1:   MOV     CURADD,(R2)+    ;PUT CURRENT ADD ON POINTER TAB
023400  013722  006520                  MOV     CURCC,(R2)+     ;PUT CURRENT CC ON POINTER TAB
023404  010237  006524                  MOV     R2,CPTR         ;PUT UPDATED R2 BACK TO CURRENT POINT
023410  013702  006516                  MOV     MSGTYP,R2       ;GET MESSAGE TYPE TO USE AS INDEX
023414  006302                          ASL     R2              ;DOUBLE FOR WORD INDEX
023416  013737  006526  006534         MOV     CURADD,TEMP     ;MOVE CURRENT ADD TO TEMP
023424  063737  006520  006534         ADD     CURCC,TEMP      ;ADD CHAR COUNT TO IT TO GET END
023432  013703  006526                  MOV     CURADD,R3       ;SET R3 TO CURRENT START ADD
023436  016237  002150  006540 BLDB2:  MOV     DMSGCT(R2),TEMP2        ;GET BYTE COUNT
023444  016204  002172                  MOV     DMSGAD(R2),R4   ;PUT STARTING FROM ADD IN R4
023450  060437  006540                  ADD     R4,TEMP2        ;ADD IT TO TEMP2 TO GET END OF FROM
023454  112423                 BLDB3:   MOVB    (R4)+,(R3)+     ;MOV BYTE FROM PATTERN TO BUFFER
023456  020337  006534                  CMP     R3,TEMP         ;ALL DONE?
023462  001404                          BEQ     BLDBEX          ;IF SO EXIT
023464  020437  006540                  CMP     R4,TEMP2        ;IS PATTERN COUNT EXPIRED
023470  001762                          BEQ     BLDB2           ;IF SO GO START AGAIN
023472  000770                          BR      BLDB3           ;IF NOT GET ANOTHER BYTE
023474  063737  006520  006526 BLDBEX: ADD     CURCC,CURADD    ;BUMP CURADD
023502  012603                          MOV     (SP)+,R3        ;RESTORE R3 AND R2
023504  012602                          MOV     (SP)+,R2
023506  000207                          RTS     PC              ;RETURN TO CALLER
```

```
                        .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 = STORAGE ADDRESS OF MESSAGE IN CMPBUF
                        ;                    MSGLIN = MAXIMUM NUMBER OF MESSAGES THAT CAN BE STORED
                        ;
                        ;       BEGIN FACSIMILE ROUTINE
                        ;       (*COPY TXBUF ==> CMPBUF*)
                        ;       ..SAVE R1
                        ;       ..INIT R1
                        ;       ..REPEAT
                        ;       ....[CMPBUF]R1=[TXBUF]R1
                        ;       ....R1=R1+1
                        ;       ..UNTIL R1 = BUFLIM
                        ;
                        ;       (*NOW CALCULATE EXPECT LIST MESSAGE POINTER*)
                        ;       ..CMPPTR = PTRTAB + (2 * MSGLIM)
                        ;
                        ;       (*NOW PRIME THE WHILE - DO LOOP*)
                        ;       ..TXPTR = PTRTAB
                        ;       ..CCURAD = CMPBUF
                        ;       ..TXPTR = TXPTR + 2
                        ;       ..CTOTCC = [TXPTR]
                        ;       ..CMPTOT = 0
                        ;       ..WHILE TXMTOT <> CMPTOT DO
                        ;       ....[CMPPTR] = CCURAD
                        ;       ....CMPPTR = CMPPRT + 2
                        ;       ....[CMPPTR] = CTOTCC
                        ;       ....TXPTR = TXPTR + 4
                        ;       ....CCURAD = CCURAD + CTOTCC
                        ;       ....CTOTCC = [TXPTR]
                        ;       ....CMPPTR = CMPPTR + 2
                        ;       ....CMPTOT = CMPTOT + 1
                        ;       ..END WHILE DO
                        ;       ..CTOTCC = TTOTCC
                        ;       END FACSIMILE ROUTINE
```

```
 4060  023510                                  FACSIMILE:
 4061
 4062  023510  010146                                  MOV     R1,-(SP)                ;SAVE R1
 4063  023512  005001                                  CLR     R1                      ;INIT R1
 4064  023514  116161  003150  005150  10$:           MOVB    TXBUF(R1),CMPBUF(R1)    ;COPY TX BUFFER TO EXPECTED BUFFER
 4065  023522  005201                                  INC     R1                      ;BUMP INDEX
 4066  023524  020127  001000                          CMP     R1,#BUFLIM              ;ALL DATA COPIED ?
 4067  023530  001371                                  BNE     10$                     ;NO,BRANCH
 4068
 4069  023532  012701  000017          20$:           MOV     #MSGLIM,R1              ;MESSAGE LIMIT
 4070  023536  006301                                  ASL     R1                      ;MULTIPLY BY 2
 4071  023540  006301                                  ASL     R1                      ;MULTIPLY BY 2
 4072  023542  012737  006150  006444                  MOV     #PTRTAB,CMPPTR          ;TOP OF POINTER TABLE
 4073  023550  060137  006444                          ADD     R1,CMPPTR               ;START OF EXPECTED POINTER TABLE
 4074  023554  005001                                  CLR     R1                      ;INIT R1
 4075
 4076                                                  ;SET UP WHILE - DO LOOP
 4077  023556  012737  006150  006442                  MOV     #PTRTAB,TXPTR           ;TX POINTER NOW AT TOP OF TABLE
 4078  023564  012737  005150  006452                  MOV     #CMPBUF,CCURAD          ;TRANSFER ADDRESS OF 1ST MESSAGE
 4079  023572  062737  000002  006442                  ADD     #2,TXPTR                ;BUMP POINTER
 4080  023600  017737  162636  006450                  MOV     @TXPTR,CTOTCC           ;BYTE COUNTER 1ST MESSAGE
 4081  023606  005037  006446                          CLR     CMPTOT                  ;INIT EXPECTED MESSAGE COUNT
 4082
 4083                                                  ;WHILE TX MESSAGE TOTAL <> EXPECTED MESSAGE TOTAL DO
 4084  023612  023737  006462  006446  30$:           CMP     TXMTOT,CMPTOT           ;ALL MESSAGES COPIED ?
 4085  023620  001430                                  BEQ     40$                     ;YES,BRANCH
 4086  023622  013777  006452  162614                  MOV     CCURAD,@CMPPTR          ;TRANSFER ADDRESS OF MESSAGE
 4087  023630  062737  000002  006444                  ADD     #2,CMPPTR               ;BUMP POINTER
 4088  023636  013777  006450  162600                  MOV     CTOTCC,@CMPPTR          ;BYTE COUNT OF MESSAGE
 4089  023644  062737  000004  006442                  ADD     #4,TXPTR                ;BUMP TX MESSAGE POINTER
 4090  023652  063737  006450  006452                  ADD     CTOTCC,CCURAD           ;CALC. TRANSFER ADDRESS
 4091  023660  017737  162556  006450                  MOV     @TXPTR,CTOTCC           ;BYTE COUNT NEXT MESSAGE
 4092  023666  062737  000002  006444                  ADD     #2,CMPPTR               ;BUMP POINTER
 4093  023674  005237  006446                          INC     CMPTOT                  ;INCREMENT MESSAGE COUNT
 4094  023700  000744                                  BR      30$                     ;DO IT AGAIN
 4095                                                  ;END WHILE - DO
 4096  023702  013737  006464  006450  40$:           MOV     TTOTCC,CTOTCC           ;COPY TOTAL CHARACTER COUNT
 4097
 4098                                                  ;END ROUTINE
 4099  023710  012601                                  MOV     (SP)+,R1                ;RESTORE R1
 4100  023712  000207                                  RTS     PC                      ;RETURN
 4101
 4102
```

```
                        .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
                        ;--

023714  013702  007704          SHWOP:  MOV     DEV1,R2         ;GET THE MODE TYPE IN R2
023720  006302                  ASL     R2                      ;MAKE IT A WORD TABLE OFFSET
023722  016237  003102  006534          MOV     MODES(R2),TEMP  ;GET ADDRESS OF MODE-IN-ASCII
023730  013702  007706                  MOV     DEV2,R2         ;GET MAINTENANCE LOOP TYPE
023734  006302                  ASL     R2
023736  012737  013724  006542          MOV     #LP00,TEMP3     ;LOAD TEMP3 TO POINT TO "/LOOP="
023744  005702                  TST     R2                      ;SEE IF /LOOP=XXXXX OR NONE
023746  001003                  BNE     10$                     ; BR IF /LOOP= OF SOME KIND
023750  012737  013723  006542          MOV     #LP0,TEMP3      ;IF NO LOOP THEN DON'T PRINT "/LOOP="
023756  016237  003120  006536  10$:    MOV     LOOPS(R2),TEMP1 ;GET ADDRESS OF LOOP-IN-ASCII
023764  013737  007710  006540          MOV     DEV3,TEMP2      ;GET NUMBER OF PASSES
023772                          PRINTS  #SHF0,TEMP,TEMP3,TEMP1,TEMP2
023772  013746  006540                                                  MOV     TEMP2,-(SP)
023776  013746  006536                                                  MOV     TEMP1,-(SP)
024002  013746  006542                                                  MOV     TEMP3,-(SP)
024006  013746  006534                                                  MOV     TEMP,-(SP)
024012  012746  014507                                                  MOV     #SHF0,-(SP)
024016  012746  000005                                                  MOV     #5,-(SP)
024022  010600                                                          MOV     SP,R0
024024  104416                                                          TRAP    C$PNTS
024026  062706  000014                                                  ADD     #14,SP

024032  005002                  CLR     R2                      ;NOW SET UP FOR QUALIFIERS IN ASCII
024034  012737  014003  006534          MOV     #PST,TEMP
024042  032737  000001  007712          BIT     #STATB,DEV4     ;SEE IF /STATUS OR /NOSTATUS
024050  001003                  BNE     1$                      ;BR IF /STATUS
024052  012737  014001  006534          MOV     #PNST,TEMP
024060  012737  014014  006536  1$:     MOV     #PCK,TEMP1
024066  032737  000002  007712          BIT     #DATCKB,DEV4    ;SEE IF /CHECK OR /NOCHECK
024074  001003                  BNE     2$                      ;BR IF /CHECK
024076  012737  014012  006536          MOV     #PNCK,TEMP1
024104  012737  014024  006540  2$:     MOV     #PEC,TEMP2
024112  032737  000004  007712          BIT     #ECHOB,DEV4     ;SEE IF /ECHO OR /NOECHO
024120  001003                  BNE     4$                      ;BR IF /ECHO
024122  012737  014022  006540          MOV     #PNEC,TEMP2
```

```
 4159
 4160  024130  012737  014043  006544  4$:     MOV     #PPR,TEMP4
 4161  024136  032737  000040  007712          BIT     #PROTOB,DEV4    ;SEE OF /PROTOCOL OR /NOPROTOCOL
 4162  024144  001003                          BNE     3$              ;BR IF /PROTOCOL
 4163  024146  012737  014041  006544          MOV     #PNPR,TEMP4
 4164
 4165  024154  012737  014033  006546  3$:     MOV     #PMS,TEMP5
 4166  024162  032737  000010  007712          BIT     #MOCHK,DEV4     ;SEE IF /MODEM OR /NOMODEM
 4167  024170  001003                          BNE     5$              ;BR IF MODEM
 4168  024172  012737  014031  006546          MOV     #PNMS,TEMP5
 4169
 4170
 4171  024200                          5$:     PRINTS  #SHF1,TEMP,TEMP1,TEMP2,TEMP5,TEMP4 :,TEMP3
 4172  024200  013746  006544                                                          MOV     TEMP4,-(SP)
 4173  024204  013746  006546                                                          MOV     TEMP5,-(SP)
 4174  024210  013746  006540                                                          MOV     TEMP2,-(SP)
 4175  024214  013746  006536                                                          MOV     TEMP1,-(SP)
 4176  024220  013746  006534                                                          MOV     TEMP,-(SP)
 4177  024224  012746  014545                                                          MOV     #SHF1,-(SP)
 4178  024230  012746  000006                                                          MOV     #6,-(SP)
 4179  024234  010600                                                                  MOV     SP,R0
 4180  024236  104416                                                                  TRAP    C$PNTS
 4181  024240  062706  000016                                                          ADD     #16,SP
 4182  024244  000207                          RTS     PC              ;RETURN
 4183
 4184
```

```
                                        .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
                                        ;--

024246                                  P$TRV:
024246  013704  003132                          MOV     P$BUFA,R4
024252  013703  003134                          MOV     P$TREE,R3
024256  105714                         P$TR5:   TSTB    (R4)                    ;SEE IF ANY CHARS LEFT IN INPUT STRING
024260  001441                                 BEQ     P$EXIT                  ;BR IF NO
024262  121327  000013                          CMPB    (R3),#11.               ;SEE IF SPECIAL CLI CHAR CODE OR ASCII
024266  003023                                 BGT     20$                     ;BR IF REGULAR ASCII CHAR.
024270  111305                                 MOVB    (R3),R5                 ;GET SPECIAL CHAR CODE INTO R5
024272  006305                                 ASL     R5
024274  016505  024310                        MOV     10$(R5),R5              ;BUILD TRAVERSE ROUTINE ADDRESS
024300  062705  024310                        ADD     #10$,R5
024304  004715                                 JSR     PC,(R5)                 ;JSR TO SPECIAL CLI TRAVERSE ROUTINE
024306  000763                                 BR      P$TR5                   ;GO SEE IF MORE OF STRING LEFT

024310  000114                         10$:     .WORD   TRVERR-10$              ;TRAVERSE TABLE FOR "CLI FUNTIONS"
024312  000134                                 .WORD   TRVEXI-10$              ;1
024314  000152                                 .WORD   TRVBR-10$               ;2
024316  000162                                 .WORD   TRVBIF-10$              ;3
024320  000204                                 .WORD   TRVSPA-10$              ;4
024322  000270                                 .WORD   TRVNUM-10$              ;5
024324  000604                                 .WORD   TRVALP-10$              ;6
024326  000650                                 .WORD   TRVALN-10$              ;7
024330  000270                                 .WORD   TRVOCT-10$              ;8
024332  000256                                 .WORD   TRVDEC-10$              ;9
024334  000736                                 .WORD   TRVSTR-10$              ;10

                                        ;NOT A SPECIAL CODE

024336  121314                         20$:     CMPB    (R3),(R4)               ;SEE IF FIRST CHAR OF STRING IS A MATCH
024340  001403                                 BEQ     22$                     ;BR IF A MATCH
024342  004737  024406                        JSR     PC,TRVBRC               ;IF NOT A MATCH, GO TAKE MISS BRANCH
024346  000743                                 BR      P$TR5                   ; THEN GO BACK PT'G TO MISS NODE
024350  004737  024366                 22$:    JSR     PC,TRVACT               ;IF A MATCH, GO DO ACTION DEFINED BY
024354  062703  000004                        ADD     #4,R3                   ; ACTION CODE IN CLI NODE, THEN
                                                                                ; ADJUST PTR TO NEXT CLI NODE
024360  005204                                 INC     R4                      ;ADJUST BUF PTR TO NEXT CHAR IF MATCH
024362  000735                                 BR      P$TR5
```

```
4241
4242  024364  000207                  P$EXIT: RTS     PC                      ;RETURN FROM PARSER
4243
4244                                  ;-----------------------------------------------------------------------
4245
4246                                  ;GOTO USER ACTION ROUTINE
4247  024366  116302  000001          TRVACT: MOVB    1(R3),R2                ;GET ACTION CODE FROM CLI NODE
4248  024372  042702  177400                  BIC     #177400,R2              ;CLEAR ANY SIGN EXTENSION
4249  024376  013705  003136                  MOV     P$ACT,R5                ;GET ADDRESS OF CLI ACTION ROUTINE
4250  024402  004715                          JSR     PC,(R5)                 ;GO DO ACTION DEFINED BY CODE
4251  024404  000207                          RTS     PC                      ;RETURN TO CALLING CODE
4252
4253                                  ;TAKE BRANCH IN TREE
4254  024406  016305  000002          TRVBRC: MOV     2(R3),R5                ;GET BRANCH DISPLACEMENT FROM TREE
4255  024412  060503                          ADD     R5,R3                   ; AND POINT R3 TO THE 'MISS" NODE
4256  024414  000207                          RTS     PC                      ; RETURN TO P$TRV
4257
4258                                  ;NO BRANCH TAKEN
4259  024416  062703  000004          TRVNOB: ADD     #4,R3                   ;THINGS OK, UPDATE R3 TO POINT TO NEXT
4260  024422  000207                          RTS     PC                      ; NODE AND RETURN TO P$TRV
4261
4262                                  ;-----------------------------------------------------------------------
4263  024424  004737  024366          TRVERR: JSR     PC,TRVACT               ;TAKE ERROR ACTION
4264  024430  112737  177777  003147          MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
4265  024436  005726                          TST     (SP)+                   ;GET RID OF "JSR PUSH TO TRVERR"
4266  024440  000137  024364                  JMP     P$EXIT                  ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
4267
4268  024444  004737  024366          TRVEXI: JSR     PC,TRVACT               ;TAKE EXIT ACTION
4269  024450  105037  003147                  CLRB    P$GDBD                  ;SET GOOD/BAD FLAG TO "SUCCESS (0)"
4270  024454  005726                          TST     (SP)+                   ;GET RID OF "JSR PUSH TO TRVEXI"
4271  024456  000137  024364                  JMP     P$EXIT                  ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
4272
4273  024462  004737  024366          TRVBR:  JSR     PC,TRVACT               ;GO TAKE BRANCH ACTION
4274  024466  000137  024406                  JMP     TRVBRC
4275
4276  024472  004737  024366          TRVBIF: JSR     PC,TRVACT
4277  024476  105737  003147                  TSTB    P$GDBD                  ;SEE IF P$GDBD SET OR CLEARED BY ACTION
4278  024502  001402                          BEQ     1$                      ;IF CLEAR FALL THRU TO NEXT NODE
4279  024504  000137  024406                  JMP     TRVBRC                  ;ELSE TAKE THE "MISS" BRANCH
4280  024510  000137  024416          1$:     JMP     TRVNOB                  ;JUST UPDATE TO NEXT NODE IF THINGS OK
4281
4282  024514  005005                  TRVSPA: CLR     R5                      ;CLEAR "SPACE OR TAB FOUND" FLAG
4283  024516  121427  000011          1$:     CMPB    (R4),#11                ;SEE IF CHAR. IN CMD LINE= TAB
4284  024522  001003                          BNE     2$                      ;BR IF NO, NOT A TAB
4285  024524  005204                          INC     R4                      ;INC INPUT STRING POINTER
4286  024526  005205                          INC     R5                      ;INDICATE A TAB FOUND
4287  024530  000772                          BR      1$                      ;GO CHECK NEXT CHAR
4288
4289  024532  121427  000040          2$:     CMPB    (R4),#40                ;SEE IF CHAR. IN CMD LINE= SPACE
4290  024536  001003                          BNE     10$                     ;BR IF NO, NON-SPACE OR NON-TAB CHAR.
4291  024540  005204                          INC     R4                      ;INC INPUT STRING POINTER
4292  024542  005205                          INC     R5                      ;INDICATE A SPACE FOUND
4293  024544  000764                          BR      1$                      ;GO CHECK NEXT CHAR
4294  024546  005705                  10$:    TST     R5                      ;SEE IF ANY SPACES OR TABS FOUND
4295  024550  001404                          BEQ     15$                     ;BR IF NO, TAKE NO ACTION
4296  024552  004737  024366                  JSR     PC,TRVACT               ;GO TAKE ACTION IF ANY FOUND
```

```
 4297  024556  000137  024416                 JMP     TRVNOB                  ;JUST GO UPDATE R3 TO NEXT NODE IF OK
 4298  024562  000137  024406          15$:   JMP     TRVBRC                  ;TAKE BRANCH (MISS) IF NONE FOUND
 4299
 4300
 4301  024566  012737  000012  003144  TRVDEC: MOV     #10.,P$RADX             ;USE DECIMAL AS RADIX AND ASSUME +
 4302  024574  000137  024606                 JMP     TRVNMA
 4303  024600                          TRVOCT: ;(SAME AS TRVNUM SINCE DEFAULT RADIX IS OCTAL)
 4304  024600  012737  000010  003144  TRVNUM: MOV     #8.,P$RADX              ;USE OCTAL AS RADIX AND ASSUME +
 4305  024606  005005                  TRVNMA: CLR     R5                      ;CLEAR DIGIT COUNTER
 4306  024610  121427  000053                 CMPB    (R4),#'+                ;SEE IF THERE'S A + SIGN THERE
 4307  024614  001001                         BNE     10$                     ; BR IF NO
 4308  024616  000406                         BR      11$                     ; ELSE P$RADX ALREADY SAYS +, JUST BR
 4309  024620  121427  000055          10$:   CMPB    (R4),#'-                ;SEE IF THERE'S A - SIGN THERE
 4310  024624  001004                         BNE     1$                      ; BR IF NO
 4311  024626  112737  177777  003145         MOVB    #-1,P$RADX+1            ;SET 'MINUS FLAG" (HI BYTE OF P$RADX)
 4312  024634  005204                  11$:   INC     R4                      ;BUMP R4 TO POINT TO FIRST CHAR
 4313
 4314  024636  121427  000060          1$:    CMPB    (R4),#60                ;SEE IF CHAR. LESS THAN A "0"
 4315  024642  002434                         BLT     2$                      ;BR IF YES (NOT NUMERIC)
 4316  024644  121427  000067                 CMPB    (R4),#67                ;SEE IF CHAR. GREATER THAN A "7"
 4317  024650  003426                         BLE     13$                     ; BR IF YES
 4318  024652  123727  003144  000012         CMPB    P$RADX,#10.             ;SEE IF IN DECIMAL MODE
 4319  024660  001417                         BEQ     12$                     ; BR IF YES (CAN USE HIGHER LIMIT)
 4320  024662  121427  000071                 CMPB    (R4),#71                ;SEE IF DIGIT WAS A 8 OR 9
 4321  024666  003022                         BGT     2$                      ;BR IF NON-NUMERIC
 4322  024670                                 PRINTF  #CLIBRX                 ;ELSE WAS A 8 OR 9 WHEN IN OCTAL RADIX
 4323  024670  012746  011700                                                                 MOV     #CLIBRX,-(SP)
 4324  024674  012746  000001                                                                 MOV     #1,-(SP)
 4325  024700  010600                                                                         MOV     SP,R0
 4326  024702  104417                                                                         TRAP    C$PNTF
 4327  024704  062706  000004                                                                 ADD     #4,SP
 4328  024710  112737  177777  003147         MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
 4329  024716  000474                         BR      5$                      ; PRINT ERROR AND TAKE MISS
 4330
 4331  024720  121427  000071          12$:   CMPB    (R4),#71                ;SEE IF CHAR. GREATER THAN A "9"
 4332  024724  003003                         BGT     2$                      ;BR IF YES (NOT NUMERIC)
 4333  024726  005204                  13$:   INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR.
 4334  024730  005205                         INC     R5                      ;INDICATE A NUMERIC FOUND
 4335  024732  000741                         BR      1$                      ;GO LOOK AT NEXT CHAR.
 4336
 4337  024734  005705                  2$:    TST     R5                      ;SEE IF FOUND ANY NUMERICS
 4338  024736  001464                         BEQ     5$                      ;BR IF NO, TAKE "MISS" BRANCH
 4339  024740  010401                         MOV     R4,R1                   ;GET POINTER TO START OF NUMERIC STRING
 4340  024742  160501                         SUB     R5,R1
 4341  024744  005037  003142                 CLR     P$NUM                   ;CLEAR LOC. WHERE VALUE WILL BE STORED
 4342  024750  112102                  3$:    MOVB    (R1)+,R2                ;GET ASCII CHAR AND CONVERT IT TO A #
 4343  024752  162702  000060                 SUB     #60,R2
 4344  024756  006337  003142                 ASL     P$NUM                   ;SHIFT CURRENT VALUE TO MAKE ROOM
 4345  024762  103437                         BCS     7$                      ;ERROR IF NUMBER TOO BIG
 4346  024764  013737  003142  003140         MOV     P$NUM,P$CNT             ;SAVE FOR LATER IN CASE DECIMAL RADIX
 4347  024772  006337  003142                 ASL     P$NUM
 4348  024776  103431                         BCS     7$                      ;ERROR IF NUMBER TOO BIG
 4349  025000  006337  003142                 ASL     P$NUM
 4350  025004  103426                         BCS     7$                      ;ERROR IF NUMBER TOO BIG
 4351  025006  123727  003144  000012         CMPB    P$RADX,#10.             ;SEE IF DECIMAL RADIX
 4352  025014  001004                         BNE     4$                      ;BR IF NOT EQUAL
```

```
 4353  025016  063737  003140  003142          ADD     P$CNT,P$NUM
 4354  025024  103416                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
 4355  025026  060237  003142          4$:     ADD     R2,P$NUM
 4356  025032  103413                          BCS     7$                      ;ERROR IF NUMBER TOO BIG
 4357  025034  005305                          DEC     R5
 4358  025036  001344                          BNE     3$
 4359  025040  105737  003145                  TSTB    P$RADX+1                ;SEE IF NUM WAS PRECEDED BY A - SIGN
 4360  025044  001402                          BEQ     15$                     ; BR IF NO
 4361  025046  005437  003142                  NEG     P$NUM                   ; ELSE NEGATE THE NUMBER BEFORE LEAVING
 4362  025052  004737  024366          15$:    JSR     PC,TRVACT               ;SINCE NUMERIC FOUND, GO TAKE ACTION
 4363  025056  000137  024416                  JMP     TRVNOB                  ;GO POINT R3 TO NEXT NODE
 4364
 4365  025062                          7$:     PRINTF  #CLINBG                 ;PRINT NUMBER TOO BIG ERROR
 4366  025062  012746  011656                                                          MOV     #CLINBG,-(SP)
 4367  025066  012746  000001                                                          MOV     #1,-(SP)
 4368  025072  010600                                                                  MOV     SP,R0
 4369  025074  104417                                                                  TRAP    C$PNTF
 4370  025076  062706  000004                                                          ADD     #4,SP
 4371  025102  112737  177777  003147          MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
 4372  025110  000137  024406          5$:     JMP     TRVBRC                  ;TAKE "MISS" BRANCH
 4373
 4374
 4375  025114  005005                  TRVALP: CLR     R5                      ;CLEAR ALPHA FOUND FLAG
 4376  025116  121427  000101          1$:     CMPB    (R4),#101               ;SEE IF CHAR. LESS THAN A "A"
 4377  025122  002406                          BLT     2$                      ;BR IF YES (NOT ALPHA)
 4378  025124  121427  000132                  CMPB    (R4),#132               ;SEE IF CHAR. GREATER THAN A "Z"
 4379  025130  003003                          BGT     2$                      ;BR IF YES (NOT ALPHA)
 4380  025132  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR
 4381  025134  005205                          INC     R5                      ;INDICATE AN ALPHA WAS FOUND
 4382  025136  000767                          BR      1$                      ;GO LOOK AT NEXT CHAR.
 4383  025140  005705                  2$:     TST     R5                      ;SEE IF ANY ALPHA'S WERE FOUND
 4384  025142  001404                          BEQ     3$                      ;BR IF NO
 4385  025144  004737  024366                  JSR     PC,TRVACT               ;IF ANY FOUND TAKE ACTION
 4386  025150  000137  024416                  JMP     TRVNOB                  ;THEN UPDATE R3 TO NEXT NODE -NO BRANCH
 4387  025154  000137  024406          3$:     JMP     TRVBRC                  ;NONE FOUND, TAKE MISS BRANCH
 4388
 4389  025160  005005                  TRVALN: CLR     R5                      ;CLEAR ALPHANUM FOUND FLAG
 4390  025162  121427  000060          10$:    CMPB    (R4),#60                ;SEE IF CHAR. LESS THAN A "0"
 4391  025166  002417                          BLT     2$                      ;BR IF YES (NOT NUMERIC OR ALPHA)
 4392  025170  121427  000072                  CMPB    (R4),#72                ;SEE IF CHAR. GREATER THAN A "9"
 4393  025174  003003                          BGT     1$                      ;BR IF YES (NOT NUMERIC)
 4394  025176  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR.
 4395  025200  005205                          INC     R5                      ;INDICATE A NUMERIC FOUND
 4396  025202  000767                          BR      10$                     ;GO LOOK AT NEXT CHAR.
 4397  025204  121427  000101          1$:     CMPB    (R4),#101               ;SEE IF CHAR. LESS THAN A "A"
 4398  025210  002406                          BLT     2$                      ;BR IF YES (NOT ALPHA)
 4399  025212  121427  000132                  CMPB    (R4),#132               ;SEE IF CHAR. GREATER THAN A "Z"
 4400  025216  003003                          BGT     2$                      ;BR IF YES (NOT ALPHA)
 4401  025220  005204                          INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR
 4402  025222  005205                          INC     R5                      ;INDICATE AN ALPHA FOUND
 4403  025224  000756                          BR      10$                     ;GO LOOK AT NEXT CHAR.
 4404  025226  005705                  2$:     TST     R5                      ;SEE IF ANY ALPHANUM'S WERE FOUND
 4405  025230  001404                          BEQ     3$                      ;BR IF NO
 4406  025232  004737  024366                  JSR     PC,TRVACT               ;IF ANY FOUND TAKE ACTION
 4407  025236  000137  024416                  JMP     TRVNOB                  ;THEN UPDATE R3 TO NEXT NODE -NO BRANCH
 4408  025242  000137  024406          3$:     JMP     TRVBRC                  ;NONE FOUND, TAKE MISS BRANCH
```

```
 4409
 4410
 4411
 4412  025246  010401              TRVSTR: MOV     R4,R1                   ;POINT R1 TO CMD STRING
 4413  025250  010305                      MOV     R3,R5
 4414  025252  062705  000006              ADD     #6,R5                   ;POUNT R5 TO MATCH STRING FROM CLI NODE
 4415  025256  005037  003140              CLR     P$CNT                   ;CLEAR CHAR MATCH COUNT
 4416  025262  105715              2$:     TSTB    (R5)                    ;SEE IF END OF MATCH STRING YET
 4417  025264  001411                      BEQ     10$                     ;BR IF YES
 4418  025266  105711                      TSTB    (R1)                    ;SEE IF END OF CMD LINE YET
 4419  025270  001407                      BEQ     10$                     ;BR IF YES
 4420  025272  121115                      CMPB    (R1),(R5)               ;SEE IF CHARACTERS MATCH
 4421  025274  001005                      BNE     10$                     ;BR IF NO
 4422  025276  005237  003140              INC     P$CNT                   ;MATCH -INCREMENT MATCH COUNT
 4423  025302  005201                      INC     R1                      ;UPDATE STRING POINTERS
 4424  025304  005205                      INC     R5
 4425  025306  000765                      BR      2$                      ;BR TO CONTINUE CHECKING CHARS.
 4426
 4427  025310  005737  003140      10$:    TST     P$CNT                   ;WHEN DONE SEE IF ANY MATCHES FOUND
 4428  025314  001406                      BEQ     15$                     ;BR IF NO, GO TAKE THE MISS BRANCH
 4429  025316  010104                      MOV     R1,R4                   ;POINT CMD POINTER TO END OF STRING &
 4430  025320  004737  024366              JSR     PC,TRVACT               ;IF A MATCH FOUND, GO DO MATCH ACTION
 4431  025324  066303  000004              ADD     4(R3),R3                ;UPDATE R3 TO NEXT NODE (NO BRANCH)
 4432  025330  000207                      RTS     PC                      ; (NO RETURN THRU TRVNOB SINCE DIFFERNT
 4433                                                                      ;  DISPLACEMENT DUE TO MATCH STRING)
 4434  025332  000137  024406      15$:    JMP     TRVBRC                  ; GO TAKE BRANCH
 4435
 4436                                                                      ; (PARSED OK), -1 IF ILL CMD.....
 4437                              ;-----------------------------------------------------------------------
 4438
```

```
 4439                                   .SBTTL  REPORT CODING SECTION
 4440
 4441
 4442                                   ;++
 4443                                   ; THE REPORT CODING SECTION CONTAINS THE
 4444                                   ; 'PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
 4445                                   ;--
 4446
 4447  025336                                   BGNRPT
 4448  025336                                                                  L$RPT::
 4449
 4450
 4451  025336  004737  020612                   JSR     PC,REPORT             ;CALL SUBROUTINE TO DUMP EVENT LOG
 4452                                                                          ; AND BASE TABLE
 4453
 4454
 4455
 4456
 4457  025342                                   ENDRPT
 4458  025342                                                                  L10010:
 4459  025342  104425                                                                 TRAP    C$RPT
```

```
                                .SBTTL  PROTECTION TABLE

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

                                        BGNPROT
                                                                        L$PROT::

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

                                        ENDPROT
```

```
 4476                                   .SBTTL  INITIALIZE SECTION
 4477
 4478                                   ;++
 4479                                   ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
 4480                                   ; AT THE BEGINNING OF EACH PASS.
 4481                                   ;--
 4482
 4483  025352                                   BGNINIT
 4484  025352                                                                          L$INIT::
 4485
 4486  025352  005037  003010                   CLR     KEYWD1                 ;INIT COMMAND STORAGE VARIABLE
 4487  025356  005737  006564                   TST     DCLFLG                 ;'EXIT' COMMAND ?
 4488  025362  001403                           BEQ     INIT1                  ;NO,BRANCH
 4489  025364  005037  006564                   CLR     DCLFLG                 ;INIT 'DO CLEAN VARIABLE'
 4490  025370                                   DOCLN                          ;GO CLEANUP AND EXIT
 4491  025370  104444                                                                  TRAP    C$DCLN
 4492
 4493  025372  012737  177777  006566  INIT1:  MOV     #-1,RESFLG             ;SET RESTART FLAG
 4494  025400                                   READEF  #EF.START              ;IF HERE CAUSE OF START,DO SOME INIT
 4495  025400  012700  000040                                                          MOV     #EF.START,R0
 4496  025404  104447                                                                  TRAP    C$REFG
 4497  025406                                   BCOMPLETE       START
 4498  025406  103417                                                                  BCS     START
 4499  025410                                   READEF  #EF.RESTART            ;IF HERE CAUSE OF RESTART, DO SOME INIT
 4500  025410  012700  000037                                                          MOV     #EF.RESTART,R0
 4501  025414  104447                                                                  TRAP    C$REFG
 4502  025416                                   BCOMPLETE       RESTRT
 4503  025416  103513                                                                  BCS     RESTRT
 4504  025420                                   READEF  #EF.CONTINUE           ;SEE IF WE'RE HERE CAUSE OF A CONTINUE
 4505  025420  012700  000036                                                          MOV     #EF.CONTINUE,R0
 4506  025424  104447                                                                  TRAP    C$REFG
 4507  025426                                   BNCOMPLETE      S1             ;BR IF NOT HERE CAUSE OF CONITNUE
 4508  025426  103002                                                                  BCC     S1
 4509  025430  000137  026120                   JMP     ENDIT                  ;JMP IF HERE CAUSE OF A CONTINUE
 4510  025434                           S1:     READEF  #EF.NEW                ;SEE IF THIS IS A 'NEW PASS''
 4511  025434  012700  000035                                                          MOV     #EF.NEW,R0
 4512  025440  104447                                                                  TRAP    C$REFG
 4513  025442                                   BCOMPLETE       NEW            ;IF YES, BR AROUND LOGUNIT # SETUP
 4514  025442  103521                                                                  BCS     NEW
 4515  025444  000523                           BR      GETPRM
 4516
 4517  025446  005037  006566           START:  CLR     RESFLG                 ;CLEAR RESTART FLAG SINCE HERE ON START
 4518  025452  005037  006626                   CLR     CLKVEC                 ;CLEAR CLK VECTOR PTR. AS A FLAG IN
 4519                                                                          ; NO CLOCK IS FOUND.
 4520  025456  012702  006622                   MOV     #CLKCSR,R2             ;SETUP R2 AS A PTR. TO CLOCK INFO BLOCK
 4521  025462                                   CLOCK   L,R1                   ;LOOK FOR A LINE CLOCK
 4522  025462  012700  000114                                                          MOV     #'L,R0
 4523  025466  104462                                                                  TRAP    C$CLCK
 4524  025470  010001                                                                  MOV     R0,R1
 4525  025472                                   BNCOMPLETE      S2             ; IF NONE THERE GO LOOK FOR A P-CLOCK
 4526  025472  103006                                                                  BCC     S2
 4527  025474  004737  017732                   JSR     PC,CLKSET              ; GO SET UP CLOCK INFO TABLE & CLK VEC.
 4528  025500  012737  000100  006632           MOV     #LCLKEN,CLKEN          ;SETUP THE ENABLE LINE CLOCK DATA
 4529  025506  000457                           BR      RESTRT
 4530
 4531  025510                           S2:     CLOCK   P,R1                   ;LOOK FOR A P-CLOCK SINCE NO LINE CLOCK
```

```
4532  025510  012700  000120                                                  MOV     #'P,R0
4533  025514  104462                                                          TRAP    C$CLCK
4534  025516  010001                                                          MOV     R0,R1
4535  025520                          BNCOMPLETE      S3              ; IF NONE THERE GO SEE IF THIS IS LSI
4536  025520  103017                                                          BCC     S3
4537  025522  004737  017732          JSR     PC,CLKSET               ; ELSE GO SET UP CLOCK INFO & VECTOR
4538  025526  062737  000002  006622  ADD     #2,CLKCSR               ;POINT CLKCSR TO P-CLK COUNT SET REG.
4539  025534  012777  001600  161060  MOV     #PCLKCT,@CLKCSR         ;LOAD CLK SET REG. WITH COUNT VALUE
4540  025542  162737  000002  006622  SUB     #2,CLKCSR               ;POINT CLKCSR BAC TO P-CLK CSR
4541  025550  012737  000111  006632  MOV     #PCLKEN,CLKEN           ;SETUP THE ENABLE THE P-CLK DATA
4542  025556  000433                  BR      RESTRT
4543
4544  025560                  S3:     READBUS                         ;READ BUS TYPE TO SEE IF ON AN LSI
4545  025560  104407                                                          TRAP    C$RDBU
4546  025562                          BNCOMPLETE      S4              ;BR IF NOT, NO CHANCE OF A CLOCK
4547  025562  103021                                                          BCC     S4
4548  025564  012737  000100  006626  MOV     #100,CLKVEC             ;LOAD 100 AS CLK VECTOR
4549  025572  005037  006624          CLR     CLKBR                   ;LOAD 0 AS CLK INT. LEVEL
4550  025576  012737  006632  006622  MOV     #CLKEN,CLKCSR           ;KLUDGE UP THE CSR & ENABLE DATA LOCS
4551  025604                          GMANID  L5060,CLKHZ,D,377,50.,60.,YES
4552  025604  104443                                                          TRAP    C$GMAN
4553  025606  000406                                                          BR      10000$
4554  025610  006630                                                          .WORD   CLKHZ
4555  025612  000052                                                          .WORD   T$CODE
4556  025614  014072                                                          .WORD   L5060
4557  025616  000377                                                          .WORD   377
4558  025620  000062                                                          .WORD   T$LOLIM
4559  025622  000074                                                          .WORD   T$HILIM
4560  025624                                                          10000$:
4561  025624  000410                  BR      RESTRT
4562
4563  025626                  S4:     PRINTF  #BDCLK                  ;INFORM OPR. NO CLOCK, & EXIT INIT
4564  025626  012746  014210                                                  MOV     #BDCLK,-(SP)
4565  025632  012746  000001                                                  MOV     #1,-(SP)
4566  025636  010600                                                          MOV     SP,R0
4567  025640  104417                                                          TRAP    C$PNTF
4568  025642  062706  000004                                                  ADD     #4,SP
4569
4570  025646  005037  006634  RESTRT: CLR     TIMMIN                  ;CLEAR TIME SINCE START LOCATIONS
4571  025652  005037  006636          CLR     TIMSEC
4572  025656  013737  006630  006640  MOV     CLKHZ,TIMTCK            ;LOAD TICKS/SEC
4573  025664  012702  006652          MOV     #EVTLOG,R2              ;INIT EVENT TABLE TO ALL 1'S AFTER EACH
4574  025670  010237  006650          MOV     R2,EVTPTR               ; START OR RES AND INIT TABLE POINTER
4575  025674  012722  177777  1$:     MOV     #-1,(R2)+
4576  025700  020227  007554          CMP     R2,#EVTEND              ;SEE IF REACHED END OF TABLE
4577  025704  001373                  BNE     1$                      ;LOOP UNTIL DONE
4578
4579  025706  012737  177777  006560  NEW:    MOV     #-1,LOGUNT              ;INITIALIZE LOGICAL UNIT #
4580
4581  025714  005237  006560  GETPRM: INC     LOGUNT                  ;POINT TO NEXT LOGICAL UNIT
4582  025720  023737  006560  002012  CMP     LOGUNT,L$UNIT           ;SEE IF PAST MAX. LOG. UNIT #
4583  025726  002367                  BGE     NEW                     ;BR IF YES, AND START OVER
4584
4585  025730                          GPHARD  LOGUNT,R1               ;GET THE P-TABLE FOR THIS LOG. UNIT
4586  025730  013700  006560                                                  MOV     LOGUNT,R0
4587  025734  104442                                                          TRAP    C$GPHRD
```

```
4588  025736  010001                          MOV     R0,R1
4589  025740                                  BNCOMPLETE      GETPRM          ;IF NO P-TABLE AVAIL., GO GET NEXT ONE
4590  025740  103365                          BCC     GETPRM
4591
4592  025742  011137  006574                  MOV     (R1),FHDPLX             ;PUT FULL OR HALF DUPLEX ANSWER IN LOC.
4593
4594
4595                                          ;DEVICE DEPENDENT PART OF GETTING INFO FROM P-TABLE
4596
4597  025746  016137  000002  011450          MOV     2(R1),RXCSR             ;STORE AWAY CSR ADDRESSES
4598
4599
4600  025754  016137  000002  011452          MOV     2(R1),PARCSR
4601  025762  062737  000002  011452          ADD     #2,PARCSR
4602  025770  016137  000002  011454          MOV     2(R1),RXDBUF
4603  025776  062737  000002  011454          ADD     #2,RXDBUF
4604  026004  016137  000002  011456          MOV     2(R1),TXCSR
4605  026012  062737  000004  011456          ADD     #4,TXCSR
4606  026020  016137  000002  011460          MOV     2(R1),TXDBUF
4607  026026  062737  000006  011460          ADD     #6,TXDBUF
4608
4609  026034  016137  000004  011462          MOV     4(R1),INVEC             ;STORE AWAY INPUT INTERRUPT VECTOR
4610  026042  016137  000004  011464          MOV     4(R1),OUTVEC
4611  026050  062737  000004  011464          ADD     #4,OUTVEC               ;BUILD OUTPUT INTERRUPT VECTOR
4612  026056  016137  000006  011466          MOV     6(R1),INTPRI            ;STORE AWAY INTERRUPT PRIORITY
4613  026064  016137  000014  011522          MOV     14(R1),RNODE            ;STORE AWAY THE REMOTE NODE TYPE
4614  026072  016137  000010  037132          MOV     10(R1),MPPTP            ;MULTI-POINT =1
4615  026100  001004                          BNE     10$                     ;IF MULTI-POINT GET ADDRESS FROM PTABLE
4616  026102  112737  000001  037065          MOVB    #1,TRIBN                ;IF POINT-POINT ADDRESS ALWAYS =1
4617  026110  000403                          BR      ENDIT                   ;EXIT
4618  026112  116137  000012  037065  10$:     MOVB    12(R1),TRIBN            ;STORE AWAY TRIB ADDRESS
4619
4620  026120                          ENDIT:
4621  026120                                  SETVEC  CLKVEC,#CLKINT,#340     ;SETUP CLOCK VECTOR
4622  026120  012746  000340                                                          MOV     #340,-(SP)
4623  026124  012746  017756                                                          MOV     #CLKINT,-(SP)
4624  026130  013746  006626                                                          MOV     CLKVEC,-(SP)
4625  026134  012746  000003                                                          MOV     #3,-(SP)
4626  026140  104437                                                                  TRAP    C$SVEC
4627  026142  062706  000010                                                          ADD     #10,SP
4628
4629                                          ;DEVICE DEPENDENT VECTOR SETUP
4630
4631  026146                                  SETVEC  INVEC,#DVRXI,#PRI05     ;SETUP INPUT INTERRUPT VECTOR
4632  026146  012746  000240                                                          MOV     #PRI05,-(SP)
4633  026152  012746  035556                                                          MOV     #DVRXI,-(SP)
4634  026156  013746  011462                                                          MOV     INVEC,-(SP)
4635  026162  012746  000003                                                          MOV     #3,-(SP)
4636  026166  104437                                                                  TRAP    C$SVEC
4637  026170  062706  000010                                                          ADD     #10,SP
4638  026174                                  SETVEC  OUTVEC,#DVTXI,#PRI05    ;SETUP OUTPUT INTERRUPT VECTOR
4639  026174  012746  000240                                                          MOV     #PRI05,-(SP)
4640  026200  012746  036230                                                          MOV     #DVTXI,-(SP)
4641  026204  013746  011464                                                          MOV     OUTVEC,-(SP)
4642  026210  012746  000003                                                          MOV     #3,-(SP)
4643  026214  104437                                                                  TRAP    C$SVEC
```

```
 4644  026216  062706  000010                                                        ADD     #10,SP
 4645
 4646  026222                        SETPRI  #PRI00          ;SET THE "RUN" PRIORITY TO 0
 4647  026222  012700  000000                                                        MOV     #PRI00,R0
 4648  026226  104441                                                               TRAP    C$SPRI
 4649  026230                        EXIT    INIT
 4650  026230  104432                                                               TRAP    C$EXIT
 4651  026232  000002                                                               .WORD   L10012-.
 4652
 4653
 4654                                .EVEN
 4655
 4656  026234                        ENDINIT
 4657  026234                                                              L10012:
 4658  026234  104411                                                               TRAP    C$INIT
```

```
                                .SBTTL  AUTODROP SECTION

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

026236                                  BGNAUTO
026236                                                                  L$AUTO::

026236                                  ENDAUTO
026236                                                                  L10013:
026236  104461                                                                  TRAP    C$AUTO
```

```
                        .SBTTL  CLEANUP CODING SECTION

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

                                BGNCLN
                                                                                L$CLEAN::

                                CLR     @CLKCSR         ;DISABLE CLOCK
                                SETPRI  #PRI07          ;SET PROCESSOR PRIORITY BACK TO 7
                                                                                MOV     #PRI07,R0
                                                                                TRAP    C$SPRI
                                CMP     #EXIT,KEYWD1    ;'EXIT' COMMAND ?
                                BEQ     EXITCLN         ;YES,BRANCH

                        ;;LOG ^C ABORT IN EVENT LOG
                                MOV     #ABO,TEMP       ;EVENT TYPE
                                MOV     OPVAR,TEMP4     ;START TIME-OUTS
                                MOV     PSCNT,TEMP2     ;PASSES
                                MOV     ERRCNT,TEMP3    ;ERRORS
                                JSR     PC,LOGS5        ;GO LOG IT
                        EXITCLN:BRESET                  ;CLEAR ALL BEFORE END
                                                                                TRAP    C$RESET

                                EXIT    CLN
                                                                                TRAP    C$EXIT
                                                                                .WORD   L10014-.


                                .EVEN

                                ENDCLN
                                                                        L10014:
                                                                                TRAP    C$CLEAN
```

```
 4711                                   .SBTTL  DROP UNIT SECTION
 4712
 4713                                   ;++
 4714                                   ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
 4715                                   ; TO NO LONGER BE TESTED.
 4716                                   ;--
 4717
 4718  026326                                   BGNDU
 4719  026326                                                                  L$DU::
 4720
 4721
 4722  026326                                   EXIT    DU
 4723  026326  000167                                                                 .WORD   J$JMP
 4724  026330  000000                                                                 .WORD   L10015-2-.
 4725
 4726
 4727                                   .EVEN
 4728
 4729  026332                                   ENDDU
 4730  026332                                                                  L10015:
 4731  026332  104453                                                                 TRAP    C$DU
```

```
 4732                                   .SBTTL  ADD UNIT SECTION
 4733
 4734                                   ;++
 4735                                   ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
 4736                                   ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
 4737                                   ;  TO THE TEST CYCLE.
 4738                                   ;--
 4739
 4740  026334                                   BGNAU
 4741  026334                                                                  L$AU::
 4742
 4743
 4744  026334                                   EXIT    AU
 4745  026334  000167                                                                 .WORD   J$JMP
 4746  026336  000000                                                                 .WORD   L10016-2-.
 4747
 4748
 4749                                   .EVEN
 4750
 4751  026340                                   ENDAU
 4752  026340                                                                  L10016:
 4753  026340  104452                                                                TRAP    C$AU
 4754
 4755
```

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


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


026342                                  BGNTST
026342                                                                          T1::

                                .SBTTL          PROGRAM SETUP SECTION

026342  013777  006632  160252          MOV     CLKEN,@CLKCSR   ;ENABLE THE CLOCK

026350                          GTXRXB:
026350  005001                  GTRA2:  CLR     R1
026352  012737  000001  006642          MOV     #1,TIMER1       ;SET TIMER TO COUNT 1 TICK
026360  005737  006642          1$:     TST     TIMER1          ;CHECK FOR IT TO BE COUNTED OFF
026364  001412                          BEQ     GTRA3           ;BRANCH IF CLOCK EXISTS (COUNTED A TICK)
026366  005301                          DEC     R1
026370  001373                          BNE     1$              ;KEEP CHECKING UNTIL R1 DOES FULL COUNTDWN
026372                                  PRINTF  #NOCLK          ;PRINT BAD CLK MSG AND WARN OF HANG IF TIMEOUT
026372  012746  014234                                                  MOV     #NOCLK,-(SP)
026376  012746  000001                                                  MOV     #1,-(SP)
026402  010600                                                          MOV     SP,R0
026404  104417                                                          TRAP    C$PNTF
026406  062706  000004                                                  ADD     #4,SP

026412  005737  006566          GTRA3:  TST     RESFLG          ;SEE IF HERE AFTER A RESTART.
026416  001112                          BNE     GTRA5           ;BR IF HERE CAUSE OF A RESTART

                                ; CLEAR COUNTS AND SET UP DEFAULTS

026420  005037  006530          GTRA4:  CLR     TOTCC           ;CLEAR TOTAL CHAR. COUNT TEMP. LOC.
026424  005037  006464                  CLR     TTOTCC          ; CLEAR TOTAL CHAR. COUNT FOR TX BUFF
026430  005037  006450                  CLR     CTOTCC          ; CLEAR TOTAL CHAR. COUNT FOR CMP BUFF
026434  012701  006150                  MOV     #PTRTAB,R1      ;INIT TRANSMIT MESSAGE POINTER
026440  010137  006442                  MOV     R1,TXPTR
026444  005037  006440                  CLR     RXPTR           ; ZERO RX POINTER

026450  012737  006244  006444          MOV     #PTR13,CMPPTR   ;INIT COMPARE MESSAGE POINTER

026456  012737  000005  006516          MOV     #5,MSGTYP       ;SET UP DEFAULT MSG TYPE (QUICK FOX - ITEP MSG)
026464  013737  002162  006520          MOV     MSG5C,CURCC     ;SET UP DEFAULT CHAR COUNT
026472  012737  003150  006466          MOV     #TXBUF,TCURAD   ;SET UP CURRENT ADD TO START OF TX BUFFER
026500  012737  005150  006452          MOV     #CMPBUF,CCURAD  ;SET UP CURRENT ADD TO START OF CMP BUFFER

026506  013737  006466  006526          MOV     TCURAD,CURADD   ;SETUP CURRENT ADDR TO START OF TXBUF
026514  013737  006442  006524          MOV     TXPTR,CPTR      ;SETUP CURRENT POINTER TABLE POINTER FOR TXBUF
026522  004737  023364                  JSR     PC,BLDBUF       ; GO BUILD POINTER TABLE AND BUFFER
026526  012737  000001  006462          MOV     #1,TXMTOT       ;BUMP TOTAL MESSAGE COUNT
```

```
4812
4813  026534  013737  006444  006524          MOV     CMPPTR,CPTR     ;SET UP START OF COMPARE POINTER TABLE
4814  026542  013737  006452  006526          MOV     CCURAD,CURADD   ;SET UP CURRENT ADDR. TO START OF CMPBUF
4815  026550  012737  000005  006516          MOV     #5,MSGTYP
4816  026556  013737  002162  006520          MOV     MSG5C,CURCC
4817  026564  004737  023364                  JSR     PC,BLDBUF       ;PUT DEFAULT MESSAGE INTO CMPBUF
4818  026570  012737  000001  006446          MOV     #1,CMPTOT       ;BUMP THE COMP MESG COUNT
4819  026576  012737  000003  006570          MOV     #ACT,MODTYP     ;SET DEFAULT MODE= ACTIVE
4820  026604  005037  006572                  CLR     MLTYP           ;SET DEFAULT MAINTENANCE LOOP MODE =NONE
4821  026610  012737  000001  006600          MOV     #1,RPASS        ;SET UP DEFAULT 'RUN PASS" COUNT TO 1
4822  026616  012737  000002  006576          MOV     #2,PARAM        ;SET UP PROG. PARAMETERS - DATACHECKING ENABLED
4823                                          ;                          OPERATOR STATUS MSGS. PRINT OFF
4824  026624                                  PRINTF  #HLP0
4825  026624  012746  012213                                                  MOV     #HLP0,-(SP)
4826  026630  012746  000001                                                  MOV     #1,-(SP)
4827  026634  010600                                                          MOV     SP,R0
4828  026636  104417                                                          TRAP    C$PNTF
4829  026640  062706  000004                                                  ADD     #4,SP
4830  026644                          GTRA5:  SETVEC  INVEC,#DVRXI,#PRI05  ;DEFAULT NON-PROTOCOL RX INTERRUPT ROUTINE
4831  026644  012746  000240                                                  MOV     #PRI05,-(SP)
4832  026650  012746  035556                                                  MOV     #DVRXI,-(SP)
4833  026654  013746  011462                                                  MOV     INVEC,-(SP)
4834  026660  012746  000003                                                  MOV     #3,-(SP)
4835  026664  104437                                                          TRAP    C$SVEC
4836  026666  062706  000010                                                  ADD     #10,SP
4837  026672  042737  000300  006576          BIC     #PRORUN!ABORT,PARAM     ;INIT PROTOCOL VARIABLES
4838  026700  013737  006570  007704          MOV     MODTYP,DEV1
4839  026706  013737  006572  007706          MOV     MLTYP,DEV2
4840  026714  013737  006600  007710          MOV     RPASS,DEV3
4841  026722  013737  006576  007712          MOV     PARAM,DEV4
4842  026730  004737  023714                  JSR     PC,SHWOP        ;PRINT TO OPERATOR THE CURRENT MODE.....
4843
4844  026734                                  MANUAL                  ;SEE IF MANUAL INTERVENTION ALLOWED
4845  026734  104450                                                          TRAP    C$MANI
4846  026736                                  BCOMPLETE       GETCL   ;  BR IF YES (UAM=0 AND NOT CHAINED)
4847  026736  103412                                                          BCS     GETCL
4848  026740  005737  006600                  TST     RPASS           ;SEE IF THIS IS FIRST "DCLT PASS"
4849  026744  001002                          BNE     1$              ; BR IF NOT COMPLETED 1 PASS
4850  026746                                  EXIT    TST             ; IF DONE 1 PASS IN UNATTENDED MODE - EXIT
4851  026746  104432                                                          TRAP    C$EXIT
4852  026750  017254                                                          .WORD   L10017-.
4853  026752  012737  000001  006572  1$:     MOV     #TTL,MLTYP      ;SET UP DEFAULT FOR UNATTENDED MODE
4854  026760  000137  031736                  JMP     GTR9            ;  'R M=ACT/LO=I/PAS=1/NOST/CH" AND RUN
4855
4856                                  .SBTTL          COMMAND LINE FETCH & INTERPRETATION SECTION
4857
4858  026764  105037  003147          GETCL:  CLRB    P$GDBD          ;CLEAR CMD LINE PARSING ERROR FLAGS
4859  026770  105037  003146                  CLRB    P$NNUF
4860  026774                                  GMANID  CLI$PM,CMDBUF,A,-1,1,72.,NO     ;GET A COMMAND LINE FROM OPR.
4861  026774  104443                                                          TRAP    C$GMAN
4862  026776  000406                                                          BR      10000$
4863  027000  002666                                                          .WORD   CMDBUF
4864  027002  000142                                                          .WORD   T$CODE
4865  027004  011570                                                          .WORD   CLI$PM
4866  027006  177777                                                          .WORD   -1
4867  027010  000001                                                          .WORD   T$LOLIM
```

```
027012  000110                                                                          .WORD   T$HILIM
027014                                                                        10000$:
027014  012737  002666  003132          MOV     #CMDBUF,P$BUFA
027022  012737  007714  003134          MOV     #CLITRE,P$TREE
027030  012737  027740  003136          MOV     #CLIACT,P$ACT
027036  005037  003012                  CLR     QUALFG                  ;CLEAR QUALIFIER FLAG LOCATION
027042  004737  024246                  JSR     PC,P$TRV                ;GO PARSE COMMAND LINE
027046  105737  003147                  TSTB    P$GDBD                  ;SEE IF PARSED OK OR AN ERROR
027052  001412                          BEQ     1$
027054                                  PRINTF  #CLIERM
027054  012746  011603                                                                  MOV     #CLIERM,-(SP)
027060  012746  000001                                                                  MOV     #1,-(SP)
027064  010600                                                                          MOV     SP,R0
027066  104417                                                                          TRAP    C$PNTF
027070  062706  000004                                                                  ADD     #4,SP
027074  000137  026764                  JMP     GETCL
027100  105737  003146          1$:     TSTB    P$NNUF                  ;SEE IF INCOMPLETE COMMAND TYPED
027104  001412                          BEQ     10$
027106                                  PRINTF  #CLINUF
027106  012746  011633                                                                  MOV     #CLINUF,-(SP)
027112  012746  000001                                                                  MOV     #1,-(SP)
027116  010600                                                                          MOV     SP,R0
027120  104417                                                                          TRAP    C$PNTF
027122  062706  000004                                                                  ADD     #4,SP
027126  000137  026764                  JMP     GETCL

027132  023727  003010  000060  10$:    CMP     KEYWD1,#SETET           ;WAS''SET EXPECT=TRANSMIT'' TYPED ?
027140  001711                          BEQ     GETCL                   ;YES,BRANCH
027142  023727  003010  000005          CMP     KEYWD1,#HLP             ;SEE IF HELP WAS TYPED
027150  001705                          BEQ     GETCL                   ;GO GET CMD AGAIN IF YES
027152  023727  003010  000055          CMP     KEYWD1,#PRNT            ;SEE IF PRINT WAS TYPED
027160  001701                          BEQ     GETCL                   ; GO GET CMD AGAIN IF YES
027162  023727  003010  000004          CMP     KEYWD1,#RUN             ;SEE IF RUN WAS TYPED
027170  001002                          BNE     11$                     ; BR IF NO
027172  000137  031736                  JMP     GTR9                    ; START EXEC. IF YES
027176  023727  003010  000052  11$:    CMP     KEYWD1,#DMPS            ;SEE IF DUMP WAS TYPED
027204  001004                          BNE     12$                     ; BR IF NO
027206  004737  023130                  JSR     PC,DUMPSR               ; ELSE, DUMP PART OF MEMORY
027212  000137  026764                  JMP     GETCL                   ; THEN RETURN TO GET ANOTHER CMD.
027216  023727  003010  000057  12$:    CMP     KEYWD1,#EXIT            ;EXIT ?
027224  001005                          BNE     13$                     ;NO,BRANCH
027226  012737  000001  006564          MOV     #1,DCLFLG               ;SET CLEANUP FLAG
027234                                  EXIT    TST                     ;GO BACK TO INIT
027234  104432                                                                          TRAP    C$EXIT
027236  016766                                                                          .WORD   L10017-.

027240  023727  003010  000001  13$:    CMP     KEYWD1,#CLEAR           ;SEE IF CLEAR WAS TYPED
027246  001646                          BEQ     GETCL                   ; IF YES, BACK TO GET ANOTHER CMD.
027250  023727  003010  000002          CMP     KEYWD1,#SHOW            ;SEE IF SHOW WAS TYPED
027256  001642                          BEQ     GETCL                   ; IF YES, BACK TO GET ANOTHER CMD.
027260  023727  003010  000010  4$:     CMP     KEYWD1,#SETEXP          ;SEE IF SET EXPECTED
027266  001512                          BEQ     2$                      ; BR IF YES (A SETEXP WAS TYPED)
027270  013737  006464  006530  5$:     MOV     TTOTCC,TOTCC
027276  023727  006530  001000          CMP     TOTCC,#BUFLIM           ;SEE IF BUFFER ALREADY FULL
027304  002414                          BLT     15$                     ; BR IF NOT FULL (BUFLIM # OF CHARS.)
027306                                  PRINTF  #MSGTRN,#BUFEX          ; ELSE TELL OPR. AND DON'T BUILD MSG.
```

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

                2$:     MOV     CTOTCC,TOTCC            ;SETUP CHAR. COUNT, CURR. ADDR. & PTR
                        CMP     TOTCC,#BUFLIM           ;SEE IF BUFFER ALREADY FULL
                        BLT     16$                     ; BR IF NOT FULL (BUFLIM # OF CHARS.)
                        PRINTF  #MSGTRN,#BUFEX          ; ELSE TELL OPR. AND DON'T BUILD MSG.
                                                                                MOV     #BUFEX,-(SP)
                                                                                MOV     #MSGTRN,-(SP)
                                                                                MOV     #2,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTF
                                                                                ADD     #6,SP
                        JMP     GETCL                   ; THEN GO GET A NEW COMMAND
                16$:    TST     CTOTCC                  ;IF FIRST "SET" THEN GET RID OF DEFAULT
                        BNE     7$
                        CLR     CMPTOT
                7$:
                        MOV     #.TR13,CMPPTR           ;INIT COMPARE MESSAGE POINTER
                        MOV     CMPTOT,R1
                        CMP     R1,#MSGLIM              ;SEE IF MSG COUNT EXCEEDED.
                        BLT     18$                     ; BR IF NO
```

```
 4980  027614
 4981  027614  012746  014315                    PRINTF  #MSGTRN,#TABEX          ; ELSE TELL OPR. AND DON'T BUILD MSG.
 4982  027620  012746  014373                                                            MOV     #TABEX,-(SP)
 4983  027624  012746  000002                                                            MOV     #MSGTRN,-(SP)
 4984  027630  010600                                                                    MOV     #2,-(SP)
 4985  027632  104417                                                                    MOV     SP,R0
 4986  027634  062706  000006                                                            TRAP    C$PNTF
 4987  027640  000137  026764                                                            ADD     #6,SP
 4988  027644  006301                          JMP     GETCL                  ; THEN GO GET A NEW COMMAND.
 4989  027646  006301                  18$:    ASL     R1                     ;# OF MSGS *4 = NEXT FREE PTR BLOCK
 4990  027650  060137  006444                  ASL     R1
 4991  027654  013737  006444  006524          ADD     R1,CMPPTR
 4992  027662  013737  006452  006526          MOV     CMPPTR,CPTR
 4993  027670  004737  023266                  MOV     CCURAD,CURADD
 4994  027674  004737  023364                  JSR     PC,ADDCC               ;ADD IN XHAR. COUNT AND CHECK TOTAL
 4995  027700  013737  006524  006444          JSR     PC,BLDBUF
 4996  027706  005237  006446                  MOV     CPTR,CMPPTR
 4997  027712  013737  006526  006452          INC     CMPTOT
 4998  027720  013737  006530  006450          MOV     CURADD,CCURAD          ;UPDATE CHAR. COUNT, CURR ADDRR. & PTR
 4999  027726  005337  003014                  MOV     TOTCC,CTOTCC
 5000  027732  001270                          DEC     QUALVL                 ;IF COPY WAS GIVEN, PUT MSG IN BUFF
 5001  027734  000137  026764                  BNE     2$                     ; AGAIN
 5002                                          JMP     GETCL                  ;GO BACK UNTIL GET A 'RUN'
 5003
 5004
 5005
 5006
```

```
                                        .SBTTL          ACTION TABLE AND ROUTINES
                                        ;       USER MUST CLEAR/SET P$GDBD IF USE "CLIBIF" IN CONNECTION WITH ACTION
                                        ;       R2 WILL HOLD ACTION CODE FROM PARSING (CLI) NODE
027740                                  CLIACT:
027740  006302                                  ASL     R2              ;MULTIPLY ACTION CODE BY 2
027742  016202  027756                          MOV     10$(R2),R2      ;OFFSET VALUE
027746  062702  027756                          ADD     #10$,R2         ;ADD BASE VALUE
027752  004712                                  JSR     PC,(R2)         ;GO DO ACTION
027754  000207                                  RTS     PC              ;RETURN TO TRVACT:

                                                                        ;BRIEF DESCRIPTION OF ACTONS TAKEN
027756  000150                          10$:    .WORD   ACTNUL-10$      ;NULL
027760  000152                                  .WORD   ACTCLR-10$      ;CLEAR
027762  000162                                  .WORD   ACTSHO-10$      ;SHOW
027764  001550                                  .WORD   ACTCHK-10$      ;CHECK
027766  000262                                  .WORD   ACTRUN-10$      ;RUN
027770  000172                                  .WORD   ACTHLP-10$      ;HELP
027772  000306                                  .WORD   ACTCSE-10$      ;CLEAR OR SHOW EXPECT
027774  000400                                  .WORD   ACTCST-10$      ;CLEAR OR SHOW TRANSMIT
027776  000722                                  .WORD   ACTSTE-10$      ;SET EXPECT
030000  000732                                  .WORD   ACTSTT-10$      ;SET TRANSMIT
030002  000750                                  .WORD   ACTSZE-10$      ;SIZE
030004  000760                                  .WORD   ACTCOP-10$      ;COPY
030006  000770                                  .WORD   ACTNUM-10$      ;NUMERIC VALUE FOR SIZE OR COPY
030010  001062                                  .WORD   ACTOPM-10$      ;QUOTED MESSAGE FROM USER
030012  001556                                  .WORD   ACTSTS-10$      ;STATUS
030014  001102                                  .WORD   ACTEQQ-10$      ;END OF QUOTED MESSAGE FROM USER
030016  001162                                  .WORD   ACTMS0-10$      ;ONES AS DATA
030020  001170                                  .WORD   ACTMS1-10$      ;ZEROS AS DATA
030022  001200                                  .WORD   ACTMS2-10$      ;1ALT AS DATA
030024  001210                                  .WORD   ACTMS3-10$      ;0ACT AS DATA
030026  001220                                  .WORD   ACTMS4-10$      ;ITEP AS DATA
030030  001230                                  .WORD   ACTMS5-10$      ;CCITT AS DATA
030032  001246                                  .WORD   ACTMS6-10$      ;ALPHA AS DATA
030034  001334                                  .WORD   ACTATV-10$      ;ACTIVE MODE
030036  001344                                  .WORD   ACTPAS-10$      ;PASSIVE MODE
030040  001364                                  .WORD   ACTREC-10$      ;RECEIVE MODE
030042  001372                                  .WORD   ACTLIS-10$      ;LISTEN MODE
030044  001402                                  .WORD   ACTDLL-10$      ;DOWNLINE LOAD
030046  001412                                  .WORD   ACTTRA-10$      ;TRANSMIT MODE
030050  001422                                  .WORD   ACTTAL-10$      ;TALK MODE
030052  001450                                  .WORD   ACTNO-10$       ;NO   IE /NOCHECK
030054  001460                                  .WORD   ACTECH-10$      ;ECHO
030056  001564                                  .WORD   ACTCRC-10$      ;CRC
030060  001600                                  .WORD   ACTPRO-10$      ;PROTOCOL
030062  001632                                  .WORD   ACTRPS-10$      ;STATUS
030064  001642                                  .WORD   ACTMOP-10$      ;SATELLITE IN MAINTENANCE LOOP MODE
030066  001652                                  .WORD   ACTTLP-10$      ;INTERNALTTL
030070  001662                                  .WORD   ACTCLP-10$      ;CABLE LOOP
030072  001672                                  .WORD   ACTLLP-10$      ;LOCAL MODEM LOOP
030074  001702                                  .WORD   ACTRLP-10$      ;REMOTE MODEM LOOP
030076  000142                                  .WORD   ACTNUF-10$      ;MORE COMMAND NEEDED
030100  001140                                  .WORD   ACTBCR-10$      ;BAD CHARACTER IN OPERATOR MESSAGE
030102  000656                                  .WORD   ACTDMS-10$      ;DUMP MEMORY START ADDRESS
030104  000706                                  .WORD   ACTDME-10$      ;DUMP MEMORY END ADDRESS
```

```
 5063  030106  000700                    .WORD   ACTDMQ-10$      ;DUMP WORD
 5064  030110  000246                    .WORD   ACTPRT-10$      ;PRINT
 5065  030112  001572                    .WORD   ACTMOS-10$      ;MODEM STATUS
 5066  030114  000236                    .WORD   ACTEXT-10$      ;EXIT ROUTINE
 5067  030116  001272                    .WORD   ACTSEX-10$      ;SET EX=TR
 5068
```

```
5069
5070  030120  112737  177777  003146  ACTNUF: MOVB    #-1,P$NNUF      ;SET FLAG TO SAY NEED MORE OF COMMAND
5071  030126  000207                  ACTNUL: RTS     PC              ;RETURN TO PARSER
5072
5073  030130  012737  000001  003010  ACTCLR: MOV     #CLEAR,KEYWD1   ;SET LOC TO SAY A CLEAR WAS TYPED
5074  030136  000207                          RTS     PC
5075
5076  030140  012737  000002  003010  ACTSHO: MOV     #SHOW,KEYWD1    ;SET LOC. TO SAY A SHOW WAS TYPED
5077  030146  000207                          RTS     PC
5078
5079  030150  012702  003016          ACTHLP: MOV     #HLPTAB,R2      ;SETUP R2 AS A POINTER TO HELP MSG TABLE
5080  030154                          1$:     PRINTF  #HLPF,(R2)+     ;PRINT HELP INFORMATION MESSAGES
5081  030154  012246                                                          MOV     (R2)+,-(SP)
5082  030156  012746  012271                                                  MOV     #HLPF,-(SP)
5083  030162  012746  000002                                                  MOV     #2,-(SP)
5084  030166  010600                                                          MOV     SP,R0
5085  030170  104417                                                          TRAP    C$PNTF
5086  030172  062706  000006                                                  ADD     #6,SP
5087  030176  020227  003036                  CMP     R2,#HLPEND      ;SEE IF ALL INFO PRINTED YET
5088  030202  001364                          BNE     1$              ;IF NO KEEP PRINTING
5089  030204  012737  000005  003010          MOV     #HLP,KEYWD1     ;SET LOC. TO SAY A HELP WAS TYPED
5090  030212  000207                          RTS     PC
5091
5092  030214  012737  000057  003010  ACTEXT: MOV     #EXIT,KEYWD1    ;EXIT COMMAND WAS INPUT
5093  030222  000207                          RTS     PC              ;RETURN
5094
5095  030224  012737  000055  003010  ACTPRT: MOV     #PRNT,KEYWD1    ;SET LOC. TO SAY A HELP WAS TYPED
5096  030232  004737  020612                  JSR     PC,REPORT       ;CALL ROUTINE TO PRINT EVENT LOG AND BASE TABLE
5097  030236  000207                          RTS     PC
5098
5099  030240  012737  000004  003010  ACTRUN: MOV     #RUN,KEYWD1     ;SET RUN FLAG
5100  030246  112737  177777  003146          MOVB    #-1,P$NNUF      ;SET FLAG TO SAY NEED MORE OF COMMAND
5101  030254  012737  000001  006600          MOV     #1,RPASS        ;SET DEFAULT RUN 'PASS' TO 1
5102  030262  000207                          RTS     PC
5103
5104  030264  012737  006244  006444  ACTCSE: MOV     #PTR13,CMPPTR   ;INIT COMPARE MESSAGE POINTER
5105  030272  013701  006444                  MOV     CMPPTR,R1
5106
5107  030276  013702  006446                  MOV     CMPTOT,R2
5108  030302  105037  003146                  CLRB    P$NNUF          ;FLAG THAT HAVE VALID COMMAND AT THIS PT.
5109  030306  023727  003010  000002          CMP     KEYWD1,#SHOW    ;SEE IF A CLEAR OR SHOW WAS TYPED
5110  030314  001471                          BEQ     ACTSHW          ;BR IF A SHOW WAS TYPED
5111  030316  012737  000001  006446          MOV     #1,CMPTOT       ;CLEAR COMPARE MESSAGE COUNT, CHAR. COUNT
5112  030324  005037  006450                  CLR     CTOTCC          ; AND RESET POINTER
5113
5114  030330  012737  006244  006444          MOV     #PTR13,CMPPTR   ;INIT COMPARE MESSAGE POINTER
5115  030336  013737  006444  006524          MOV     CMPPTR,CPTR     ;SET UP TO FILL IN DEFAULT MESSAGE
5116  030344  012701  005150                  MOV     #CMPBUF,R1
5117  030350  010137  006452                  MOV     R1,CCURAD
5118  030354  000431                          BR      ACTCLB
5119
5120  030356  012701  006150          ACTCST: MOV     #PTRTAB,R1
5121  030362  013702  006462                  MOV     TXMTOT,R2
5122  030366  105037  003146                  CLRB    P$NNUF          ;FLAG THAT HAVE VALID COMMAND AT THIS PT.
5123  030372  023727  003010  000002          CMP     KEYWD1,#SHOW    ;SEE IF A CLEAR OR SHOW WAS TYPED
5124  030400  001437                          BEQ     ACTSHW          ;BR IF A SHOW WAS TYPED
```

```
5125  030402  012737  000001  006462          MOV     #1,TXMTOT       ;CLEAR TRANSMIT MESSAGE COUNT, CHAR. COUNT
5126  030410  005037  006464                  CLR     TTOTCC          ; AND RESET POINTER
5127  030414  012737  006150  006442          MOV     #PTRTAB,TXPTR
5128  030422  013737  006442  006524          MOV     TXPTR,CPTR
5129  030430  012701  003150                  MOV     #TXBUF,R1
5130  030434  010137  006466                  MOV     R1,TCURAD
5131
5132  030440  012702  001000          ACTCLB: MOV     #BUFLIM,R2
5133  030444  010137  006526                  MOV     R1,CURADD       ;SET UP TO PUT DEFAULT MSG IN LIST AFTER 033'S
5134  030450  012737  000005  006516          MOV     #5,MSGTYP
5135  030456  013737  002162  006520          MOV     MSG5C,CURCC
5136  030464  105021                  1$:     CLRB    (R1)+           ;FILL EXPT OR TRAN BUFFER WITH 0'S IF A CLEAR
5137  030466  005302                          DEC     R2              ;DO "BUFLIM" NUMBER OF BYTE LOCATIONS
5138  030470  001375                          BNE     1$
5139  030472  004737  023364                  JSR     PC,BLDBUF       ;"CLEAR" REALLY MEANS TO PUT DEFAULT MSG IN
5140  030476  000207                          RTS     PC              ;WHEN DONE, RETURN TO PARSER
5141
5142
5143  030500  012705  003072          ACTSHW: MOV     #SHTAB,R5
5144  030504  122571  000000                  CMPB    (R5)+,@(R1)     ;LOOK AT FIRST BYTE OF MSG TO DECIPHER TYPE
5145  030510  001404                          BEQ     6$
5146  030512  020527  003101                  CMP     R5,#SHTEND      ;SEE IF LOOKED AT ALL OF DEFAULTS YET
5147  030516  001372                          BNE     5$
5148  030520  005205                          INC     R5              ;MUST BE OPR. SPEC'D THEN
5149  030522  162705  003073          6$:     SUB     #SHTAB+1,R5
5150  030526  006305                          ASL     R5
5151  030530  016137  000002  006534          MOV     2(R1),TEMP
5152  030536                                  PRINTF  #SHMSG,SHTYTB(R5),TEMP  ;PRINT MSG SIZE & TYPE
5153  030536  013746  006534                                                          MOV     TEMP,-(SP)
5154  030542  016546  003052                                                          MOV     SHTYTB(R5),-(SP)
5155  030546  012746  013516                                                          MOV     #SHMSG,-(SP)
5156  030552  012746  000003                                                          MOV     #3,-(SP)
5157  030556  010600                                                                  MOV     SP,R0
5158  030560  104417                                                                  TRAP    C$PNTF
5159  030562  062706  000010                                                          ADD     #10,SP
5160  030566  062701  000004                  ADD     #4,R1           ;BUMP R1 TO NEXT SET OF POINTERS
5161  030572  005302                          DEC     R2
5162  030574  001341                          BNE     ACTSHW
5163  030576  013737  006570  007704          MOV     MODTYP,DEV1
5164  030604  013737  006572  007706          MOV     MLTYP,DEV2
5165  030612  013737  006600  007710          MOV     RPASS,DEV3
5166  030620  013737  006576  007712          MOV     PARAM,DEV4
5167  030626  004737  023714                  JSR     PC,SHWOP        ;SHOW THE OPERATOR THE CURRENT MODE..... ALSO
5168  030632  000207                          RTS     PC
5169
5170  030634  013737  003142  006510  ACTDMS: MOV     P$NUM,STADD     ;SETUP STARTING ADDRESS FOR DUMP
5171  030642  005037  006514                  CLR     BYTBIT          ;SET DEFAULT OF WORD DUMP
5172  030646  012737  000052  003010          MOV     #DMPS,KEYWD1    ;FLAG THAT A DUMP WAS TYPED
5173  030654  000403                          BR      ACTDME
5174
5175  030656  012737  177777  006514  ACTDMQ: MOV     #-1,BYTBIT      ;SET DUMP FLAG TO "DUMP-WORD"
5176  030664  013737  003142  006512  ACTDME: MOV     P$NUM,ENADD     ;SETUP END ADDRESS FOR DUMP (=START IF NO "EEE"
5177  030672  105037  003146          ACTDMX: CLRB    P$NNUF          ;CLEAR NOT-ENOUGH FLAG, "DUMP N-N/B" IS VALID
5178  030676  000207                          RTS     PC
5179
```

```
5180
5181
5182  030700  012737  000010  003010  ACTSTE: MOV     #SETEXP,KEYWD1
5183  030706  000403                  BR      ACTSTX
5184
5185  030710  012737  000011  003010  ACTSTT: MOV     #SETTRN,KEYWD1
5186  030716  012737  000001  003014  ACTSTX: MOV     #1,QUALVL       ;SET UP DEFAULT COPY TO 1 (/COPY=0)
5187  030724  000207                  RTS     PC
5188
5189  030726  012737  000012  003012  ACTSZE: MOV     #SIZE,QUALFG
5190  030734  000207                  RTS     PC
5191
5192  030736  012737  000013  003012  ACTCOP: MOV     #QCOPY,QUALFG
5193  030744  000207                  RTS     PC
5194
5195  030746  023727  003012  000012  ACTNUM: CMP     QUALFG,#SIZE    ;SEE IF A SIZE OR COPY TYPED
5196  030754  001023                  BNE     1$              ;BR IF IT WAS A COPY
5197  030756  005737  003142          TST     P$NUM           ;CHECK TO BE SURE DIDN'T TRY SIZE=0
5198  030762  001014                  BNE     3$              ; BR IF NO
5199  030764                          PRINTF  #CLISE0
5200  030764  012746  012072                                                  MOV     #CLISE0,-(SP)
5201  030770  012746  000001                                                  MOV     #1,-(SP)
5202  030774  010600                                                          MOV     SP,R0
5203  030776  104417                                                          TRAP    C$PNTF
5204  031000  062706  000004                                                  ADD     #4,SP
5205  031004  112737  177777  003147          MOVB    #-1,P$GDBD      ;SEE ERROR-IN-CMD FLAG
5206  031012  000411                  BR      2$
5207  031014  013737  003142  006520  3$:     MOV     P$NUM,CURCC     ;IF A SIZE LOAD CURCC WITH BYTE COUNT
5208  031022  000405                  BR      2$
5209  031024  013737  003142  003014  1$:     MOV     P$NUM,QUALVL    ;IF A COPY, LOAD COPY COUNT
5210  031032  005237  003014          INC     QUALVL          ;INCREMENT SO FIRST DEC MAKES IT REAL #
5211  031036  000522                  2$:     BR      ACTMEX
5212
5213  031040  012737  000007  006516  ACTOPM: MOV     #7,MSGTYP
5214  031046  010437  006534          MOV     R4,TEMP         ;KEEP TRACK OF START OF QUOTED TEXT
5215  031052  005237  006534          INC     TEMP            ; SO CAN CALC OPCNT AT END OF QUOTES
5216  031056  000207                  RTS     PC
5217
5218  031060  010402                  ACTEQO: MOV     R4,R2
5219  031062  163702  006534          SUB     TEMP,R2
5220  031066  010237  006520          MOV     R2,CURCC        ;CALC BYTE COUNT FOR QUOTED TEXT
5221  031072  010237  002166          MOV     R2,OPCNT
5222  031076  013701  006534          MOV     TEMP,R1
5223  031102  012705  002520          MOV     #OPBUF,R5
5224  031106  112125                  1$:     MOVB    (R1)+,(R5)+     ;COPY QUOTED TEXT TO OPBUF
5225  031110  005302                  DEC     R2
5226  031112  001375                  BNE     1$
5227  031114  000473                  BR      ACTMEX
5228
5229  031116                          ACTBCR: PRINTF  #CLIBCR         ;BAD CHAR. IN OPR. QUOTED STRING
5230  031116  012746  012025                                                  MOV     #CLIBCR,-(SP)
5231  031122  012746  000001                                                  MOV     #1,-(SP)
5232  031126  010600                                                          MOV     SP,R0
5233  031130  104417                                                          TRAP    C$PNTF
5234  031132  062706  000004                                                  ADD     #4,SP
5235  031136  000207                  RTS     PC
```

```
5236
5237  031140  005037  006516          ACTMS0: CLR     MSGTYP
5238  031144  000435                          BR      ACTME1
5239  031146  012737  000001  006516  ACTMS1: MOV     #1,MSGTYP          ;SET MESSAGE TYPE = ALL ONES
5240  031154  000431                          BR      ACTME1
5241  031156  012737  000002  006516  ACTMS2: MOV     #2,MSGTYP          ;SET MESSAGE TYPE = ONES & ZEROS
5242  031164  000425                          BR      ACTME1
5243  031166  012737  000003  006516  ACTMS3: MOV     #3,MSGTYP          ;SET MESSAGE TYPE = ZEROS & ONES
5244  031174  000421                          BR      ACTME1
5245  031176  012737  000004  006516  ACTMS4: MOV     #4,MSGTYP          ;SET MESSAGE TYPE = CCITT
5246  031204  000415                          BR      ACTME1
5247  031206  012737  000005  006516  ACTMS5: MOV     #5,MSGTYP          ;SET MESS TYPE = QUICK FOX
5248  031214  013737  002162  006520          MOV     MSG5C,CURCC        ;SETUP DEFAULT SIZE FOR THIS TYPE
5249  031222  000430                          BR      ACTMEX
5250  031224  012737  000006  006516  ACTMS6: MOV     #6,MSGTYP          ;SET MESSAGE TYPE = ALPHA/NUM
5251  031232  013737  002164  006520          MOV     MSG6C,CURCC        ;SETUP DEFAULT SIZE FOR THIS TYPE
5252
5253  031240  012737  000100  006520  ACTME1: MOV     #64.,CURCC         ;SETUP DEFAULT SIZE FOR MSG0-4
5254  031246  000416                          BR      ACTMEX             ;BRANCH TO EXIT
5255
5256  031250  022737  000010  003010  ACTSEX: CMP     #SETEXP,KEYWD1     ;DID WE GET HERE FROM "SET E="COMMAND?
5257  031256  001404                          BEQ     10$                ;YES,BRANCH
5258  031260  112737  000001  003147          MOVB    #1,P$GDBD          ;SET ERROR FLAG
5259  031266  000406                          BR      ACTMEX             ;GO EXIT SUBROUTINE
5260  031270  004737  023510          10$:    JSR     PC,FACSIMILE       ;GO COPY TRANSMIT LIST TO EXPECT LIST
5261  031274  012737  000060  003010          MOV     #SETET,KEYWD1      ;SET FLAG TO BE USED IN T1::
5262  031302  000400                          BR      ACTMEX             ;EXIT SUBROUTINE
5263
5264  031304  105037  003146  ACTMEX: CLRB    P$NNUF             ;CLEAR NOT-ENOUGH FLAG
5265  031310  000207                          RTS     PC
5266
```

```
 5267  031312  012737  000003  006570  ACTATV: MOV     #ACT,MODTYP             ;MODE = ACTIVE
 5268  031320  000432                          BR      ACTM2X
 5269
 5270  031322  012737  000002  006570  ACTPAS: MOV     #PAS,MODTYP             ;MODE = PASSIVE
 5271  031330  105037  003146                  CLRB    P$NNUF                  ;CLEAR NOT-ENOUGH FLAG
 5272  031334  005037  006572                  CLR     MLTYP                   ;CLEAR MAINT LOOP TYPE
 5273  031340  000207                          RTS     PC
 5274
 5275  031342  005037  006570          ACTREC: CLR     MODTYP                  ;MODE = RECEIVE
 5276  031346  000417                          BR      ACTM2X
 5277
 5278  031350  012737  000006  006570  ACTLIS: MOV     #LIS,MODTYP             ;MODE = LISTEN
 5279  031356  000413                          BR      ACTM2X
 5280
 5281  031360  012737  000004  006570  ACTDLL: MOV     #DOW,MODTYP             ;MODE = DOWNLINE LOAD
 5282  031366  000407                          BR      ACTM2X
 5283
 5284  031370  012737  000001  006570  ACTTRA: MOV     #TRA,MODTYP             ;MODE = TRANSMIT
 5285  031376  000403                          BR      ACTM2X
 5286
 5287  031400  012737  000005  006570  ACTTAL: MOV     #TAL,MODTYP             ;MODE = TALK
 5288
 5289  031406  042737  000004  006576  ACTM2X: BIC     #ECHOB,PARAM            ;DISABLE /ECHO (ALL BUT PASSIVE MODE)
 5290  031414  105037  003146                  CLRB    P$NNUF                  ;CLEAR NOT-ENOUGH FLAG
 5291  031420  005037  006572                  CLR     MLTYP                   ;CLEAR MAINT LOOP TYPE
 5292  031424  000207                          RTS     PC
 5293
```

```
 5294  031426  012737  000036  003012  ACTNO:  MOV     #NO,QUALFG
 5295  031434  000207                  RTS     PC
 5296
 5297  031436  022737  000036  003012  ACTECH: CMP     #NO,QUALFG
 5298  031444  001422                  BEQ     1$
 5299  031446  052737  000004  006576          BIS     #ECHOB,PARAM
 5300  031454  022737  000002  006570          CMP     #PAS,MODTYP             ;BE SURE IN PASSIVE MODE IF
 5301  031462  001416                  BEQ     2$                      ;IF TRYING TO SET /ECHO
 5302  031464                          PRINTF  #CLINPS
 5303  031464  012746  011762                                                  MOV     #CLINPS,-(SP)
 5304  031470  012746  000001                                                  MOV     #1,-(SP)
 5305  031474  010600                                                          MOV     SP,R0
 5306  031476  104417                                                          TRAP    C$PNTF
 5307  031500  062706  000004                                                  ADD     #4,SP
 5308  031504  112737  177777  003147          MOVB    #-1,P$GDBD
 5309  031512  042737  000004  006576  1$:     BIC     #ECHOB,PARAM
 5310  031520  005037  003012          2$:     CLR     QUALFG                  ;CLEAR 'NO' OUT OF QUALIFIER FLAG
 5311  031524  000501                  BR      ACTLXX
 5312
 5313  031526  012701  000002          ACTCHK: MOV     #DATCKB,R1              ;SET DATA CHECK BIT
 5314  031532  000413                  BR      ACTQFG
 5315
 5316  031534  012701  000001          ACTSTS: MOV     #STATB,R1       ;SET THE STATUS BIT
 5317  031540  000410                  BR      ACTQFG
 5318
 5319  031542  012701  000020          ACTCRC: MOV     #CRCB,R1        ;SET THE CRC BIT
 5320  031546  000405                  BR      ACTQFG
 5321
 5322  031550  012701  000010          ACTMOS: MOV     #MOCHK,R1       ;SET THE MODEM BIT
 5323  031554  000402                  BR      ACTQFG
 5324
 5325  031556  012701  000040          ACTPRO: MOV     #PROTOB,R1      ;SET THE PROTOCOL BIT
 5326
 5327  031562  050137  006576          ACTQFG: BIS     R1,PARAM
 5328  031566  022737  000036  003012          CMP     #NO,QUALFG
 5329  031574  001002                  BNE     1$
 5330  031576  040137  006576                  BIC     R1,PARAM
 5331  031602  005037  003012          1$:     CLR     QUALFG                  ;CLEAR 'NO' OUT OF QUALIFIER FLAG
 5332  031606  000450                  BR      ACTLXX
 5333
 5334  031610  013737  003142  006600  ACTRPS: MOV     P$NUM,RPASS             ;GET NUMBER OF 'RUN PASSES'
 5335  031616  000444                  BR      ACTLXX
 5336
 5337  031620  012737  000005  006572  ACTMOP: MOV     #5,MLTYP
 5338  031626  000417                  BR      ACTLPX
 5339  031630  012737  000001  006572  ACTTLP: MOV     #1,MLTYP
 5340  031636  000413                  BR      ACTLPX
 5341  031640  012737  000002  006572  ACTCLP: MOV     #2,MLTYP
 5342  031646  000407                  BR      ACTLPX
 5343  031650  012737  000003  006572  ACTLLP: MOV     #3,MLTYP
 5344  031656  000403                  BR      ACTLPX
 5345  031660  012737  000004  006572  ACTRLP: MOV     #4,MLTYP
 5346
 5347  031666  022737  000003  006570  ACTLPX: CMP     #ACT,MODTYP     ;BE SURE IN ACTIVE IF TRYING TO SET LOOP
 5348  031674  001415                  BEQ     ACTLXX          ; BR IF IN ACTIVE
 5349  031676  112737  177777  003147          MOVB    #-1,P$GDBD
```

```
 5350  031704  005037  006572                CLR     MLTYP           ;CLEAR ANY LOOP TYPE THAT MAY HAVE GOT SET
 5351  031710                                PRINTF  #CLIBDL
 5352  031710  012746  011720                                                        MOV     #CLIBDL,-(SP)
 5353  031714  012746  000001                                                        MOV     #1,-(SP)
 5354  031720  010600                                                              MOV     SP,R0
 5355  031722  104417                                                              TRAP    C$PNTF
 5356  031724  062706  000004                                                      ADD     #4,SP
 5357  031730  105037  003146    ACTLXX:     CLRB    P$NNUF          ;CLEAR NOT-ENOUGH FLAG
 5358  031734  000207                        RTS     PC
 5359
```

```
5360
5361
5362  031736  005737  006572          GTR9:   TST     MLTYP           ;LOOP MODE ?
5363  031742  001422                          BEQ     10$             ;NO,BRANCH
5364  031744  032737  000002  006576          BIT     #DATCKB,PARAM   ;DATA CHECK ?
5365  031752  001416                          BEQ     10$             ;NO,BRANCH
5366  031754  023737  006446  006462          CMP     CMPTOT,TXMTOT   ;TX = EX ?
5367  031762  001412                          BEQ     10$             ;YES,BRANCH
5368  031764                                  PRINTF  #CLIPW          ;PRINT WARNING
5369  031764  012746  012123                                                  MOV     #CLIPW,-(SP)
5370  031770  012746  000001                                                  MOV     #1,-(SP)
5371  031774  010600                                                          MOV     SP,R0
5372  031776  104417                                                          TRAP    C$PNTF
5373  032000  062706  000004                                                  ADD     #4,SP
5374  032004  000137  026764                  JMP     GETCL           ;TRY AGAIN
5375
5376
5377                                          ; RX ALLOCATE CODE
5378  032010  012737  006150  006442  10$:    MOV     #PTRTAB,TXPTR   ;INIT TRANSMIT MESSAGE POINTER
5379  032016  012737  006244  006444          MOV     #PTR13,CMPPTR   ;INIT COMPARE MESSAGE POINTER
5380  032024  012737  006340  006440          MOV     #PTR23,RXPTR    ;INIT RECEIVE MESSAGE POINTER
5381
5382  032032  013737  006446  006476          MOV     CMPTOT,RXMTOT   ;MAKE COMPARE AND RX MESSAGE COUNTS EQUAL
5383
5384
5385  032040  005037  006602          GTREX:  CLR     FLAG            ;CLEAR FLAG
5386  032044  005037  006502                  CLR     OPVAR           ;CLEAR OPTIONAL VARIABLE COUNTER
5387  032050  005037  006504                  CLR     PSCNT           ;CLEAR PASS COUNT
5388  032054  005037  006506                  CLR     ERRCNT          ;CLEAR ERROR COUNT
5389  032060  005037  011516                  CLR     MGLCNT          ;CLEAR GLITCH COUNT
5390  032064  005037  011520                  CLR     MHRCNT          ;CLEAR HARD ERR. COUNT
5391  032070  005037  006500                  CLR     LNCNT           ;CLEAR LINE COUNTER
5392  032074  012737  000626  011506          MOV     #626,SYNCW      ;SET UP SYNCW FOR 226 SYNC +TSOM
5393  032102  052737  000200  011470          BIS     #BIT7,DUPPAR    ;RX SYNC=226
5394  032110  005737  011522                  TST     RNODE
5395  032114  001406                          BEQ     1$              ;IF NON ITEP GO TO 1
5396  032116  042737  000200  011470          BIC     #BIT7,DUPPAR    ;SET UP FOR 26 SYNC WORD ON RX.
5397  032124  012737  000426  011506          MOV     #426,SYNCW      ;ELSE SET UP SYNC FOR 26 AND TSOM
5398  032132  004737  020200          1$:     JSR     PC,LOGDVI       ;LOG ABOUT TO INIT DEVICE
5399  032136  004737  034306                  JSR     PC,DVINIT       ;INIT DEVICE
5400
5401  032142  012737  001000  006520  GTRX2:  MOV     #BUFLIM,CURCC   ;SET CHAR COUNT TO 'BUFLIM'' NO. OF BYTES
5402  032150  012737  004150  006526          MOV     #RXBUF,CURADD   ;SET UP RX BUFFER AS CURRRENT ADD.
5403  032156  013737  006440  006524          MOV     RXPTR,CPTR
5404  032164  012737  000010  006516          MOV     #10,MSGTYP      ;SET UP FOR 33 TO FILL RX BUFFERS
5405  032172  004737  023364                  JSR     PC,BLDBUF       ;CLEAR RX BUFFER
5406  032176  013702  006570                  MOV     MODTYP,R2
5407  032202  006302                          ASL     R2
5408  032204  000172  006604                  JMP     @MODE(R2)       ;MODE DISPATCH
5409
```

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

032210                                  RXONLY:
032210  013737  006440  006522          RXON2:  MOV     RXPTR,CPTRR
032216  013737  006476  006474                  MOV     RXMTOT,DVRCT    ;SET UP MESSAGE COUNT
032224  052737  000104  006602                  BIS     #QRX+#ERX,FLAG  ;SET UP RX QUE
032232  005037  006524                          CLR     CPTR            ;CLEAR THE TX POINTER
032236  000137  032400                          JMP     ALLTR           ;GO RX.
```

```
                                        .SBTTL          TRANSMIT MODE SECTION

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

 032242  042737  000002  006576  TXONLY: BIC     #DATCKB,PARAM   ;SET NOCHECK
 032250  013737  006442  006524  TXON2:  MOV     TXPTR,CPTR
 032256  013737  006462  006460          MOV     TXMTOT,DVTCT    ;COPY COUNTER FOR THIS PASS
 032264  052737  000210  006602          BIS     #QTX+#ETX,FLAG  ;SET THE QUE TX FLAG
 032272  005037  006522                  CLR     CPTRR           ;CLEAR RX POINTER
 032276  000137  032400                  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
                                        ;--

032302                                  PLCK:
032302  013737  006462  006460  PLCK2:  MOV     TXMTOT,DVTCT    ;SET UP  THE TRANSMIT COUNT
032310  013737  006442  006524          MOV     TXPTR,CPTR      ;SET UP CPTR TO TRANSMIT POINTER
032316  013737  006440  006522  PLCK3:  MOV     RXPTR,CPTRR     ;SET UP CPTRR TO REC POINTER
032324  052737  000104  006602          BIS     #QRX+#ERX,FLAG  ;SET UP Q AND EXPECT RX
032332  000137  032400                  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
                                ;--

032336  013737  006462  006460  ALCK:   MOV     TXMTOT,DVTCT    ;# OF MESSAGES TO TRANSMIT(DEVICE TX COUNT)
032344  013737  006442  006524          MOV     TXPTR,CPTR      ;SETUP TX MESSAGE LIST POINTER
032352  013737  006476  006474          MOV     RXMTOT,DVRCT    ;# OF MESSAGES TO RECEIVE(DEVICE RX COUNT)
032360  013737  006440  006522          MOV     RXPTR,CPTRR     ;SETUP RX MESSAGE LIST POINTER
032366  052737  000314  006602          BIS     #QRX+#QTX+#ETX+#ERX,FLAG
032374  000137  032400                  JMP     ALLTR
```

```
5508                                    .SBTTL          TRANSMIT - RECEIVE FOR ALL STANDARD MODES
5509
5510                                    ;++
5511                                    ; FUNCTIONAL DESCRIPTION:
5512                                    ;       THIS CODE PERFORMS THE FOLLOWING FUNCTIONS
5513                                    ;       1.) IF RX BUFFERS ARE TO BE QUED,TELL DEVICE
5514                                    ;           CODE TO QUE THEM ,LOG RECEIVE QUED.
5515                                    ;       2.) IF TX BUFFERS ARE TO BE QUED ,TELL DEVICE
5516                                    ;           CODE TO QUE THEM, LOG TRANSMIT QUED.
5517                                    ;       3.) WAIT FOR EITHER RECIVE BUFFER OR TRANSMIT BUFFER OR
5518                                    ;           BOTH TO COMPLETE
5519                                    ;       4.) IF RECEIVE COMPLETE LOG IT UPDATE RX TABLE IF DATA
5520                                    ;           CHECKING.
5521                                    ;       5.) IF TRANSMIT COMPLETE LOG IT.
5522                                    ;       6.) WHEN BOTH TRANSMIT AND RECIEVE LISTS ARE DONE
5523                                    ;           GO TO THE COMPARE BUFFER CODE
5524                                    ;
5525                                    ; SUBORDINATE ROUTINES USED:
5526                                    ;               'DVRXQ'' -QUE RECEIVE BUFFER SPACE TO DEVICE
5527                                    ;               'LOGRXQ''-LOG RECEIVE BUFFER SPACE TO EVENT LOG
5528                                    ;               'LOGTXQ''-LOG TRANSMIT BUFFER QUED TO EVENT LOG
5529                                    ;               'DVTXRX''-QUE TRANSMIT BUFFER AND WAIT FOR RX
5530                                    ;                         OR TX TO COMPLETE
5531                                    ;               'LOGRXC''-LOG RECEIVE BUFFER COMPLETED TO EVENT LOG
5532                                    ;               'LOGTXC''-LOG TRANSMIT BUFFER COMPLETED TO EVENT LOG
5533                                    ;
5534                                    ; USE OF FLAG BITS:
5535                                    ;               QRX - SET ON INPUT TO ALLTR IF REC IS TO BE QUED TO
5536                                    ;                     DEVICE. CLEARED BY DVRXQ AND THEN SET BY DVTXRX
5537                                    ;                     WHEN RX BUFFER IS COMPLETED.
5538                                    ;               QTX - SET ON INPUT TO ALLTR IF TRANSMIT IS TO BE QUED TO
5539                                    ;                     DEVICE. CLEARED ON ENTRY TO DVTXRX AND SET BY DVTXRX
5540                                    ;                     WHEN TX BUFFER IS COMPLETED.
5541                                    ;               ETX - USED BY DVTXRX TO DETERMINE IF TX BUFFER COMPLETED IS
5542                                    ;                     EXPECTED.
5543                                    ;               ERX - USED BY DVTXRX TO DETERMINE IF RX BUFFER COMPLETED IS
5544                                    ;                     EXPECTED.
5545                                    ;
5546                                    ;
5547                                    ; CALLING SEQUENCE:
5548                                    ;       JMP     ALLTR           ;GO TO TRANSMIT-RECEIVE FOR ALL STANDARD MODES
5549                                    ;--
5550
5551
5552  032400                            ALLTR:
5553  032400  032737  000004  006602    ALCK5:  BIT     #QRX,FLAG       ;QUE RX ADDRESS & COUNT? (ARE WE RECEIVING?)
5554  032406  001424                            BEQ     ALCK1           ;NO,BRANCH ?
5555  032410  013702  006522                    MOV     CPTRR,R2        ;GET CURRENT RX MESSAGE LIST POINTER(POINTS
5556                                                                    ;;TO RX MESSAGE LIST POINTER TABLE)
5557  032414  011237  006540                    MOV     (R2),TEMP2      ;SAVE RX ADDRESS FOR LOG
5558  032420  012237  006470                    MOV     (R2)+,DVRXA     ;DEVICE RX ADDRESS
5559  032424  011237  006542                    MOV     (R2),TEMP3      ;SAVE RX CHAR COUNT FOR LOG
5560  032430  011237  006472                    MOV     (R2),DVRCC      ;DEVICE RX CHARACTER COUNT
5561  032434  010237  006522                    MOV     R2,CPTRR        ;STORE UPDATED RX POINTER
5562  032440  004737  020134                    JSR     PC,LOGRXQ       ;LOG RECEIVER QUED
5563  032444  032737  000040  006576  10$:     BIT     #PROTOB,PARAM   ;'/PROTOCOL' ?
```

```
 5564  032452  001002                          BNE     ALCK1           ;YES,BRANCH
 5565  032454  004737  035170                  JSR     PC,DVRXQ        ;GO QUE RX BUFFERS & ENABLE RECEIVER
 5566
 5567  032460  032737  000010  006602  ALCK1:  BIT     #QTX,FLAG       ;ARE WE TRANSMITTING ?
 5568  032466  001416                          BEQ     ALCK2           ;NO,BRANCH
 5569  032470  013702  006524                  MOV     CPTR,R2         ;CURRENT TRANSMIT MESSAGE LIST POINTER
 5570  032474  011237  006540                  MOV     (R2),TEMP2      ;SAVE ADDRESS FOR LOG
 5571  032500  012237  006454                  MOV     (R2)+,DVTXA     ;TRANSMIT BUFFER ADDRESS
 5572  032504  011237  006542                  MOV     (R2),TEMP3      ;SAVE CHAR COUNT FOR LOG
 5573  032510  012237  006456                  MOV     (R2)+,DVTCC     ;TRANSMIT CHAR COUNT
 5574  032514  010237  006524                  MOV     R2,CPTR         ;SAVE UPDATED POINTER
 5575  032520  004737  020100                  JSR     PC,LOGTXQ       ;LOG TX QUE
 5576
 5577  032524  032737  000040  006576  ALCK2:  BIT     #PROTOB,PARAM   ;'/PROTOCOL' ?
 5578  032532  001410                          BEQ     10$             ;NO,BRANCH
 5579  032534  004737  040706                  JSR     PC,PROTOC       ;GO DO DDCMP PROTOCOL PROCESSING
 5580  032540  032737  000200  006576          BIT     #ABORT,PARAM    ;PROTOCOL ABORT ?
 5581  032546  001404                          BEQ     20$             ;NO,BRANCH
 5582  032550  000137  026644                  JMP     GTRA5           ;ABORT!! AND RETURN TO 'DCLT >' PROMPT
 5583
 5584  032554  004737  035272          10$:    JSR     PC,DVTXRX       ;IF TRANSMITTING QUE TX BUFFERS & ENABLE TX
 5585                                                                  ;;WAIT FOR TX/RX COMPLETE
 5586  032560  032737  000004  006602  20$:    BIT     #QRX,FLAG       ;RECEIVED MESSAGE ?
 5587  032566  001514                          BEQ     ALCK3           ;NO,BRANCH
 5588  032570  013737  006470  006540          MOV     DVRXA,TEMP2     ;RX BUFFER ADDRESS
 5589  032576  013737  006472  006542          MOV     DVRCC,TEMP3     ;RX CHAR COUNT
 5590  032604  004737  020152                  JSR     PC,LOGRXC       ;LOG REC COMPLETE
 5591  032610  032737  000004  006576  UPTABL: BIT     #ECHOB,PARAM    ;IS THIS ECHO MODE(PASSIVE)
 5592  032616  001406                          BEQ     UPTA4           ;IF NOT GO TO 4
 5593  032620  013702  006524                  MOV     CPTR,R2         ;ELSE SET R2 TO PRESENT TX TABL
 5594  032624  013722  006540                  MOV     TEMP2,(R2)+     ;STORE OFF RX ADD
 5595  032630  013712  006542                  MOV     TEMP3,(R2)      ;AND CC
 5596  032634  032737  000002  006576  UPTA4:  BIT     #DATCKB,PARAM   ;IS DATA CHECKING ASKED FOR
 5597  032642  001015                          BNE     UPTA1           ;IF SO GO TO 1
 5598  032644  012737  000001  006474          MOV     #01,DVRCT       ;ELSE SET DVRCT TO A 1
 5599  032652  013737  006440  006522          MOV     RXPTR,CPTRR     ;RESET POINTER
 5600  032660  022737  000003  006570          CMP     #ACT,MODTYP     ;IS THIS ACTIVE
 5601  032666  001002                          BNE     UPTA3
 5602  032670  005237  006474                  INC     DVRCT           ;IF YES BUMP COUNT
 5603  032674  000424                  UPTA3:  BR      UPTEX
 5604  032676  013702  006522          UPTA1:  MOV     CPTRR,R2
 5605  032702  011237  006534                  MOV     (R2),TEMP       ;LOAD TEMP WITH PREV. COUNT
 5606  032706  163737  006542  006534          SUB     TEMP3,TEMP      ;LOAD TEMP WITH PREV.COUNT-CURRENT
 5607  032714  013722  006542                  MOV     TEMP3,(R2)+
 5608  032720  063737  006542  006540          ADD     TEMP3,TEMP2
 5609  032726  013722  006540                  MOV     TEMP2,(R2)+     ;STORE OF NEW ADD
 5610  032732  013712  006534                  MOV     TEMP,(R2)       ;AND NEW CC
 5611  032736  162702  000002                  SUB     #2,R2           ;PUT POINTER BACK TO ADDR.
 5612
 5613  032742  010237  006522                  MOV     R2,CPTRR        ;AND RESTORE IT.
 5614
 5615  032746                          UPTEX:
 5616  032746  022737  000002  006570          CMP     #PAS,MODTYP
 5617  032754  001007                          BNE     ALCK2A          ;IF NOT PASSIVE LOOP THEN GO TO 2A
 5618  032756  042737  000104  006602          BIC     #QRX+#ERX,FLAG  ;CLEAR BOTH EXPECTED AND COMPLETED FLAGS
 5619  032764  052737  000210  006602          BIS     #QTX+#ETX,FLAG  ;SET THE TX FLAGS
```

```
 5620  032772  000632                        BR      ALCK1
 5621
 5622  032774  005337  006474        ALCK2A: DEC     DVRCT           ;DEC REC COUNT
 5623  033000  005737  006474                TST     DVRCT           ;IS IT ALL DONE
 5624  033004  001005                        BNE     ALCK3           ;NO. GO CHECK TX
 5625  033006  042737  000004  006602        BIC     #QRX,FLAG       ;CLEAR THE RX FLAG
 5626  033014  005037  006522                CLR     CPTRR           ;YES. CLEAR POINTER
 5627  033020  032737  000010  006602 ALCK3: BIT     #QTX,FLAG       ;IS IT TX
 5628  033026  001447                        BEQ     ALCK4           ;IF NOT TX THEN GO BACK
 5629  033030  013737  006454  006540        MOV     DVTXA,TEMP2
 5630  033036  013737  006456  006542        MOV     DVTCC,TEMP3     ;LOG TX COMPLETED
 5631  033044  004737  020116                JSR     PC,LOGTXC
 5632  033050  005337  006460                DEC     DVTCT           ;DEC TX COUNT
 5633  033054  022737  000002  006570        CMP     #PAS,MODTYP
 5634  033062  001013                        BNE     ALCK3A          ;IF NOT PASSIVE MODE GO TO 3A
 5635  033064  042737  000210  006602        BIC     #QTX+ETX,FLAG   ;CLEAR THE TX FLAGS
 5636  033072  052737  000104  006602        BIS     #QRX+ERX,FLAG   ;AND SET THE RX FLAGS
 5637  033100  005737  006460                TST     DVTCT
 5638  033104  001005                        BNE     ALCK3C          ;IF MORE RX'S DO IT
 5639  033106  000137  033166                JMP     CMPSR           ; ELSE COMPARE
 5640  033112  005737  006460        ALCK3A: TST     DVTCT           ;IS IT ALL DONE
 5641  033116  001402                        BEQ     ALCK3B          ;IF NOT GO BACK TO 5
 5642  033120  000137  032400        ALCK3C: JMP     ALCK5
 5643  033124  005037  006524        ALCK3B: CLR     CPTR            ;IF SO CLEAR POINTER
 5644  033130  042737  000010  006602        BIC     #QTX,FLAG       ;CLEAR TX FLAG
 5645  033136  032737  000002  006576        BIT     #DATCKB,PARAM   ;IS IT DAT CK
 5646  033144  001403                        BEQ     ALCK4A          ;IF NOT THEN END WO CKING RX.
 5647  033146  005737  006522        ALCK4:  TST     CPTRR
 5648
 5649  033152  001362                        BNE     ALCK3C          ;IF SOME RX'S LEFT GO BACK
 5650  033154  005737  006524        ALCK4A: TST     CPTR
 5651  033160  001402                        BEQ     ALCK4B          ;BRANCH IF ANY TX'S LEFT
 5652  033162  000137  032524                JMP     ALCK2
 5653  033166                        ALCK4B:
 5654
 5655
 5656
```

```
                        .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
                        ;--

033166  032737  000002  006576  CMPSR:  BIT     #DATCKB,PARAM   ;IS DATA CHECKING TO BE DONE
033174  001522                        BEQ     CMPSEX          ;IF NOT THEN EXIT
033176  013737  006440  006524        MOV     RXPTR,CPTR      ;PUT START OF RX POINTERS TO CPTR
033204  013737  006444  006522        MOV     CMPPTR,CPTRR    ; AND START OF COMPARE POINTS TO CPTRR
033212  013737  006476  006474        MOV     RXMTOT,DVRCT

033220                        CMPS3:
033220  013702  006524              MOV     CPTR,R2         ;MOVE CURRET RX PT.TO R2
033224  011237  006540              MOV     (R2),TEMP2      ;MOVE RX ADD TO EVENT LOG
033230  012201                      MOV     (R2)+,R1        ;SET R1 TO START ADD OF RX
033232  012237  006542              MOV     (R2)+,TEMP3     ;SET CHAR  COUNT TO EVENT LOG
033236  010237  006524              MOV     R2,CPTR         ;RESTORE RX POINT

033242  013702  006522              MOV     CPTRR,R2        ;PUT R2 AT COMPARE TABLE
033246  012203                      MOV     (R2)+,R3        ;SET R3 TO COMPARE ADD
033250  012204                      MOV     (R2)+,R4        ;SET R4 TO COMP CC
033252  010237  006522              MOV     R2,CPTRR        ;RESTORE POINTER
033256  010437  006544              MOV     R4,TEMP4
033262  004737  020246              JSR     PC,LOGCMP       ;LOG COMPARE START.

033266  020437  006542              CMP     R4,TEMP3        ;IS COMPARE COUNT = TO RX COUNT
033272  001410                      BEQ     CMPS7           ;IF SO GO TO 7
033274  005237  006506              INC     ERRCNT
033300                              ERRSOFT 1,EDDLE,ERR10   ;PRINT ERROR
```

```
                                                                                        TRAP    C$ERSOFT
                                                                                        .WORD   1
                                                                                        .WORD   EDDLE
                                                                                        .WORD   ERR10
033310  004737  020264                JSR     PC,LOGCML       ;LOG LENGTH ERROR

033314  005037  006544        CMPS7:  CLR     TEMP4           ;CLEAR BAD BYTE COUNTER
033320  012737  000001  006532        MOV     #1,OFSET        ;SET OFSET BYTE COUNT TO 1
033326  122123                CMPS1:  CMPB    (R1)+,(R3)+     ;COMPARE RX WITH EXPETED
033330  001422                        BEQ     CMPS6           ;IF EQUAL THEN GO TO 6

033332  005237  006544        CMPS2:  INC     TEMP4           ;INC BAD COUNT
033336  023727  006544  000005        CMP     TEMP4,#5        ;IS IT MORE THEN 5
033344  101014                        BHI     CMPS6           ;IF SO GO  FOR MORE
033346  114337  006554                MOVB    -(R3),GOOD      ;STORE GOOD BYTE FOR ERROR
033352  114137  006555                MOVB    -(R1),BAD       ;STORE BAD BYTE FOR ERROR
033356  005237  006506                INC     ERRCNT
033362                                ERRSOFT 2,EDDDE,ERR1    ;REPORT COMPARISON FAILURE TO OPR.
                                                                                        TRAP    C$ERSOFT
                                                                                        .WORD   2
                                                                                        .WORD   EDDDE
                                                                                        .WORD   ERR1
033372  005201                        INC     R1
033374  005203                        INC     R3
033376  005237  006532        CMPS6:  INC     OFSET           ;INC OFFSET
033402  005304                        DEC     R4              ;ELSE DEC CHAR COUNT AND SEE IF 0
033404  001350                        BNE     CMPS1           ;IF NOT GO BACK
033406  005737  006544                TST     TEMP4           ;SEE IF ANY CMP ERRS FOR THIS MSG
033412  001410                        BEQ     CMPS5A          ;BR IF NONE
033414  005237  006506                INC     ERRCNT
033420                                ERRSOFT 3,EDDDE,ERR2    ;REPORT # OF MISMATCHES FOR MESSAGE
                                                                                        TRAP    C$ERSOFT
                                                                                        .WORD   3
                                                                                        .WORD   EDDDE
                                                                                        .WORD   ERR2
033430  004737  020302        CMPS5:  JSR     PC,LOGCMD       ;LOG DATA ERROR IN COMPARE
033434                        CMPS5A:
033434  005337  006474                DEC     DVRCT
033440  001267                        BNE     CMPS3           ;IF NOT ALL DONE GO BACK
```

```
                                .SBTTL          MODEM CHANGE REPORTS

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

033442  005737  011516  CMPSEX: TST     MGLCNT          ;CHECK FOR ANY GLITCH ERRORS
033446  001003                  BNE     MCREP           ;IF NON ZERO REPORT THEM
033450  005737  011520          TST     MHRCNT          ;CHECK FOR ANY HARD ERRORS
033454  001412                  BEQ     ENDPS           ;IF NONE GO TO END OF PASS
                                ;
                                ;REPORT ANY MODEM ERRORS HERE
                                ;
033456  005237  006506  MCREP:  INC     ERRCNT          ;BUMP ERROR COUNT
033462                          ERRSOFT 4,MSCMS,ERR4
033462  104457                                                  TRAP    C$ERSOFT
033464  000004                                                  .WORD   4
033466  014764                                                  .WORD   MSCMS
033470  017604                                                  .WORD   ERR4
033472  005037  011516          CLR     MGLCNT          ;CLEAR GLITCH COUNT
033476  005037  011520          CLR     MHRCNT          ;CLEAR THE HARD COUNT
```

```
                                .SBTTL          INTERNAL END OF PASS CODE


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

033502  005237  006504          ENDPS:  INC     PSCNT           ;BUMP PASS COUNT

033506  013737  006502  006544          MOV     OPVAR,TEMP4
033514  013737  006504  006540          MOV     PSCNT,TEMP2
033522  013737  006506  006542          MOV     ERRCNT,TEMP3
033530  004737  020320                  JSR     PC,LOGEOP       ;LOG END OF PASS

033534  022737  177777  006600          CMP     #-1,RPASS       ;SEE IF RPASS=-1
033542  001403                          BEQ     1$              ;IF IT IS DON'T DECRMNT, LOOP FOREVER
033544  005337  006600                  DEC     RPASS           ;DEC PASS COUNT
033550  001402                          BEQ     2$              ;IF DONE EXIT TEST
033552  000137  032142          1$:     JMP     GTRX2           ;ELSE GO BACK AND DISPATCH
033556  042777  000120  155664  2$:     BIC     #RINTEN!RXENA,@RXCSR    ;TURN OFF RX
033564  000137  026644                  JMP     GTRA5           ;WHEN RPASS=0 GO BACK TO "DCLT>"
```

```
 5826                                   .SBTTL          DOWN-LINE-LOAD SECTION
 5827                                   ;++
 5828                                   ; FUNCTIONAL DESCRIPTION:
 5829                                   ;       DOWN LINE LOAD IS NOT SUPPORTED BY THIS DEVICE..
 5830                                   ;       IF THIS MODE IS CALLED BY THE COMMAND LINE INTERPRETER
 5831                                   ;       THEN A MESSAGE WILL BE PRINTED ....THAT SAYS DOWN LINE
 5832                                   ;       LOAD IS NOT!! SUPPORTED BY THIS DEVICE.
 5833                                   ;--
 5834
 5835
 5836
 5837  033570                           DLL:
 5838  033570                                   PRINTF  #DLLCM
 5839  033570  012746  014130                                                  MOV     #DLLCM,-(SP)
 5840  033574  012746  000001                                                  MOV     #1,-(SP)
 5841  033600  010600                                                          MOV     SP,R0
 5842  033602  104417                                                          TRAP    C$PNTF
 5843  033604  062706  000004                                                  ADD     #4,SP
 5844  033610  000137  026644                   JMP     GTRA5
 5845
```

```
                        .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
                        ;--

033614                  TALCK:
033614  042737  000002  006576          BIC     #DATCKB,PARAM   ;SET NOCHECK
033622  012702  002520                  MOV     #OPBUF,R2
033626  012722  177777  1$:             MOV     #-1,(R2)+       ;CLEAR OUT OPBUFFER FIRST
033632  022702  002642                  CMP     #OPEND,R2
033636  001373                          BNE     1$
033640                                  GMANID  OPRMM,OPBUF,A,-1,1,72.,NO       ;GET TALK MESSAGE
033640  104443                                                          TRAP    C$GMAN
033642  000406                                                          BR      10001$
033644  002520                                                          .WORD   OPBUF
033646  000142                                                          .WORD   T$CODE
033650  014065                                                          .WORD   OPRMM
033652  177777                                                          .WORD   -1
033654  000001                                                          .WORD   T$LOLIM
033656  000110                                                          .WORD   T$HILIM
033660                                                          10001$:
033660  005002                          CLR     R2              ;NOW GET CHAR COUNT
033662  122762  000377  002520  2$:     CMPB    #377,OPBUF(R2)
033670  001402                          BEQ     3$
033672  005202                          INC     R2
033674  000772                          BR      2$
033676  010237  002166          3$:     MOV     R2,OPCNT

033702  012737  002520  006454          MOV     #OPBUF,DVTXA    ;SET UP TX ADDR.
033710  012737  002520  006540          MOV     #OPBUF,TEMP2
033716  013737  002166  006542          MOV     OPCNT,TEMP3
033724  013737  002166  006456          MOV     OPCNT,DVTCC     ;SET UP TX CC
033732  004737  020100                  JSR     PC,LOGTXQ
033736  052737  000210  006602          BIS     #QTX+#ETX,FLAG  ;SET UP FLAGS
033744  005037  006522                  CLR     CPTRR           ;CLEAR RX POINTER

                        ;;THIS CODE ADDED FOR PROTOCOL
033750  032737  000040  006576          BIT     #PROTOB,PARAM   ;'/PROTOCOL'?
033756  001003                          BNE     20$             ;YES,BRANCH
033760  004737  035272                  JSR     PC,DVTXRX
033764  000405                          BR      25$             ;JUMP AROUND PROTOCOL
033766  042737  000004  006602  20$:     BIC     #RXQ,FLAG       ;MAKE SURE NOT TO RECEIVE
```

```
 5902  033774  004737  040706                  CALL    PROTOCOL        ;DO DDCMP PROTOCOL
 5903
 5904  034000  013737  006454  006540  25$:    MOV     DVTXA,TEMP2
 5905  034006  013737  006456  006542          MOV     DVTCC,TEMP3
 5906  034014  004737  020116                  JSR     PC,LOGTXC
 5907  034020  022737  054105  002520          CMP     #''EX,OPBUF     ;CHECK FOR EXIT
 5908  034026  001272                          BNE     TALCK
 5909  034030  022737  052111  002522          CMP     #''IT,OPBUF+2
 5910  034036  001266                          BNE     TALCK
 5911  034040  042737  000210  006602          BIC     #QTX+#ETX,FLAG  ;CLEAR THE TX BITS
 5912  034046  012737  000006  006570          MOV     #LIS,MODTYP     ;CHANGE TO LISTEN MODE
 5913  034054  000137  032142                  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
                                ;--

034060  042737  000002  006576  LISCK:  BIC     #DATCKB,PARAM   ;CLEAR CHECK BIT
034066                          LISCK:  PRINTF  #LISP           ;PRINT PROMPT FOR OPR.
034066  012746  014054                                                  MOV     #LISP,-(SP)
034072  012746  000001                                                  MOV     #1,-(SP)
034076  010600                                                          MOV     SP,R0
034100  104417                                                          TRAP    C$PNTF
034102  062706  000004                                                  ADD     #4,SP
034106  012737  002520  006470  LISCKA: MOV     #OPBUF,DVRXA    ;SET DEVICE UP TO REC AT OPBUF
034114  012737  002520  006540          MOV     #OPBUF,TEMP2
034122  012737  000122  006472          MOV     #82.,DVRCC      ;SET UP CHAR COUNT TO 82.
034130  012737  000122  006542          MOV     #82.,TEMP3
034136  052737  000104  006602          BIS     #QRX+#ERX,FLAG  ;SET UP FLAG
034144  005037  006524                  CLR     CPTR            ;CLEAR THE TX.

                                ;; WAS PROTOCOL SELECTED ?
034150  032737  000040  006576          BIT     #PROTOB,PARAM   ;'/PROTOCOL' ?
034156  001007                          BNE     20$             ;YES,BRANCH

034160  004737  035170                  JSR     PC,DVRXQ        ;QUE RX
034164  004737  020134                  JSR     PC,LOGRXQ
034170  004737  035272                  JSR     PC,DVTXRX       ;GO TO DEVICE RX. SUBROUTINE
034174  000402                          BR      25$
034176  004737  040706          20$:    CALL    PROTOC          ;DO DDCMP PROTOCOL
034202  013737  006470  006540  25$:    MOV     DVRXA,TEMP2
034210  013737  006472  006542          MOV     DVRCC,TEMP3     ;SET UP ADDR.AND CC.
034216  004737  020152                  JSR     PC,LOGRXC       ;LOG COMPLETED
034222  063737  006470  006472          ADD     DVRXA,DVRCC
034230  105077  152236                  CLRB    @DVRCC
034234                                  PRINTF  #OPBFPT
034234  012746  002514                                                  MOV     #OPBFPT,-(SP)
034240  012746  000001                                                  MOV     #1,-(SP)
034244  010600                                                          MOV     SP,R0
034246  104417                                                          TRAP    C$PNTF
034250  062706  000004                                                  ADD     #4,SP
034254  022737  054105  002520          CMP     #'EX,OPBUF      ;COMPARE FOR EX OF "EXIT"
034262  001311                          BNE     LISCKA          ;IF NOT EXIT THEN GO BACK
```

```
 5970  034264  022737  052111  002522          CMP     #''IT,OPBUF+2   ;IF FIRST HALF OK CHECK NEXT PART
 5971  034272  001305                          BNE     LISCKA          ;IF NOT EXIT THE GO BACK
 5972  034274  012737  000005  006570          MOV     #TAL,MODTYP     ;CHANGE MODE TO TALK
 5973  034302  000137  032142                  JMP     GTRX2           ;RETURN TO DISPATCHER
 5974
 5975
```

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

                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       DVINIT- DEVICE INIT ROUTINE
                        ;       THIS ROUTINE IS DEVICE DEPENDENT CODE THAT INITS
                        ;       THE DEVICE BEING TESTED.
                        ;       IT SETS THE DEVICE UP TO THE MODE IT IS TO RUN IN AND
                        ;       INITIATES THE START,STACK,ACK SEQUENCE IF THE "RNODE"(REMOTE
                        ;       NODE)INPUT INDICATES THE REMOTE NODE IS NON-ITEP.
                        ;
                        ; INPUTS:       "FHDPLX" INDICATES IF MODE IS FULL OR HALF DUPLEX. (1=FULL)
                        ;               ADDRESS POINTERS (SEL0,...) ALREADY POINT TO DEVICE'S REG.S
                        ;
                        ;               "MLTYP" INDICATES THE LOOP TYPE (1=TTL,2=CAB,3=RM,4=LM)
                        ;               "RNODE" INDICATES THE TYPE OF REMOTE NODE (ITEP=1,NON-ITEP=0)
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;
                        ;               "CTSSR" - CLEAR TO SEND SUB ROUTINE
                        ;               "DVIN31" - SEND CONTROL AND REC OR TIME OUT
                        ;               "CLRRTS" - CLEAR REQUEST TO SEND ROUTINE
                        ;                "LGDVE" - LOG DEVICE ERROR TO EVENT LOG
                        ;
                        ; CALLING SEQUENCE:
                        ;               JSR     PC,DVINIT
                        ;--
                        DVINIT:
                        ;; DO MASTER CLEAR
034306 012777 000400 155142 20$:    MOV     #RESET,@TXCSR   ;DO A MASTER CLEAR
034314 000240               NOP                             ;WAIT A WHILE FOR
034316 000240               NOP                             ;;IT TO CLEAR
034320 012777 000400 155130 MOV     #RESET,@TXCSR   ;DO A CLEAR AGAIN(NECESSARY WITH LOOP=CABLE)
034326 117737 155116 006542 MOVB    @RXCSR,TEMP3    ;SEE IF IT WORKED
034334 001423               BEQ     DVIN1           ;BRANCH IF OK
034336                      BREAK
034336 104422                                                               TRAP    C$BRK

                        ;REPORT ERROR IF RESET
                        ;DOES NOT WORK

034340 012737 016747 006540 MOV     #DVEM0,TEMP2
034346 017737 155076 006542 MOV     @RXCSR,TEMP3
034354 017737 155076 006544 MOV     @TXCSR,TEMP4    ;LOAD UP ERRM. AND REG OUTPUTS
034362 004737 020162        JSR     PC,LGDVE        ;LOG TIME OUT WAITING FOR RUN
034366 005237 006506        INC     ERRCNT
034372                      ERRSOFT 5,DVEM0,ERR13
034372 104457                                                               TRAP    C$ERSOFT
```

```
6032  034374  000005                                                          .WORD   5
6033  034376  016747                                                          .WORD   DVEM0
6034  034400  017636                                                          .WORD   ERR13
6035  034402  000741                  BR DVINIT               ;GO BACK AND TRY MSTR CLR AGAIN IF ERROR
6036
6037                                  ;SET TTL LOOP IF REQU'D
6038
6039  034404  042737  000003  006602  DVIN1:  BIC     #3,FLAG           ;CLEAR INPUT AND OUTPUT INT FLAGS
6040  034412  042777  004000  155036          BIC     #TTLL,@TXCSR      ;CLEAR INTERNAL LOOP
6041  034420  022737  000001  006572          CMP     #TTL,MLTYP        ;IS TTL SELECTED
6042  034426  001004                          BNE     DVIN3             ; IF NOT GO TO 3
6043  034430  052777  004000  155020          BIS     #TTLL,@TXCSR      ;ELSE SET INTERNAL LOOP
6044  034436  000461                          BR      DVIN37
6045
6046  034440  022737  000002  006572  DVIN3:  CMP     #CABLE,MLTYP      ;CABLE LOOP ?
6047  034446  001004                          BNE     10$               ;NO,BRANCH
6048  034450  052777  010000  155000          BIS     #CABLOP,@TXCSR    ;SET EXTERNAL LOOP (TURN AROUND CONNECTOR)
6049  034456  000451                          BR      DVIN37            ;
6050
6051  034460  022737  000004  006570  10$:    CMP     #DOW,MODTYP       ;CHECK IF DLL
6052  034466  001002                          BNE     DVIN3A            ;BRANCH IF NOT DLL
6053  034470  000137  035130                  JMP     DVINEX            ;ELSE EXIT
6054
6055  034474  012777  000002  154746  DVIN3A: MOV     #DTR,@RXCSR       ;SET UP DTR.
6056
6057  034502  012737  002000  006642          MOV     #2000,TIMER1
6058  034510  005737  006642  DVIN38: TST     TIMER1
6059  034514  001022                          BNE     DVIN39            ;IF TIMER NOT OUT GO TO 39
6060
6061                                  ;SET ERROR FOR NO MODEM READY
6062
6063  034516  012737  017377  006540          MOV     #DVEM6,TEMP2
6064  034524  017737  154720  006542          MOV     @RXCSR,TEMP3
6065  034532  017737  154720  006544          MOV     @TXCSR,TEMP4
6066  034540  004737  020162                  JSR     PC,LGDVE
6067  034544  005237  006506                  INC     ERRCNT
6068  034550                                  ERRSOFT 11,DVEM6,ERR13
6069  034550  104457                                                          TRAP    C$ERSOFT
6070  034552  000013                                                          .WORD   11
6071  034554  017377                                                          .WORD   DVEM6
6072  034556  017636                                                          .WORD   ERR13
6073  034560  000745                          BR      DVIN3A            ;THEN TRY TO SET DTR AGAIN
6074  034562                          DVIN39: BREAK
6075  034562  104422                                                          TRAP    C$BRK
6076  034564  017737  154660  011474          MOV     @RXCSR,IRXCSR     ;GET COPY OF RXCSR
6077  034572  032737  001000  011474          BIT     #BIT9,IRXCSR      ;IS MODEM READY SET
6078  034600  001743                          BEQ     DVIN38
6079  034602  013777  011470  154642  DVIN37: MOV     DUPPAR,@PARCSR    ;SET PARAMETER REGISTER
6080  034610  005737  011522                  TST     RNODE             ;REMOTE ITEP ?
6081  034614  001145                          BNE     DVINEX            ;YES,BRANCH
6082  034616  005737  006574                  TST     FHDPLX            ;FULL DUPLEX ?
6083  034622  001542                          BEQ     DVINEX            ;NO,BRANCH
6084  034624  032737  000040  006576          BIT     #PROTOB,PARAM     ;'/PROTOCOL' ?
6085  034632  001136                          BNE     DVINEX            ;YES,BRANCH
6086
6087                                  ;; THIS START-STACK ROUTINE USED IN NON-PROTOCOL,NON-ITEP,FULL DUPLEX MODE
```

```
6088
6089                                            ;SET UP TO SEND STRT
6090  034634  112737  000005  002645            MOVB    #5,HDMSG+1      ;SET UP ENQ
6091  034642  052737  000060  006602            BIS     #RXM!TXM,FLAG   ;SET FLAG WORD
6092  034650  012737  000074  006646            MOV     #60.,TIMERS     ;SET TIMER FOR 1 MINUTE
6093  034656  004737  036576                    JSR     PC,CTSSR        ;SET CTS IF NESC.
6094  034662  012737  000006  002646  DVIN41:   MOV     #6,HDMCC        ;SET UP STRT CODE
6095  034670  004737  036432                    JSR     PC,DVIN31       ;GO TX STRT AND CHK FOR RX.
6096  034674  005737  006646                    TST     TIMERS
6097  034700  001466                            BEQ     DVIN81          ;IF TIMER EXPIERED EXIT
6098
6099  034702  022737  000006  002660  DVIN4:    CMP     #6,RHDMCC       ;IS THE RCVD=STRT
6100  034710  001441                            BEQ     DVIN8           ;IF SO GO TO ASTRT
6101  034712  022737  000007  002660            CMP     #7,RHDMCC       ;IS IT A STACK
6102  034720  001360                            BNE     DVIN41          ;IF NOT STACK ETIHER GO BACK
6103
6104  034722  004737  036576  DVIN9:    JSR     PC,CTSSR        ;SET REQUEST TO SEND
6105  034726  042737  001010  006602            BIC     #QTX!PAD,FLAG   ;CLEAR TX COMPT FLAG.
6106  034734  012737  000001  002646            MOV     #1,HDMCC        ;SET UP ACK
6107  034742  012737  002645  011500            MOV     #HDMSG+1,MSGPTR ;SET UP POINTER
6108  034750  013737  002654  011502            MOV     HDMC,MSGCC
6109  034756  012737  000010  011504            MOV     #8.,SYNCC       ;SET UP SYNC COUNT
6110  034764  052777  000120  154464            BIS     #SEND!TINTEN,@TXCSR ;TURN ON TX, ENABLE TX INTERRUPT
6111  034772  032737  000010  006602  DIVN91:   BIT     #QTX,FLAG
6112  035000  001053                            BNE     DVINEX          ;EXIT IF ACK SENT
6113  035002                                    BREAK
6114  035002  104422                                                            TRAP    C$BRK
6115  035004  005737  006646                    TST     TIMERS
6116  035010  001370                            BNE     DIVN91          ;IF NOT TIMER EXPIRED RECHK TX.
6117  035012  000421                            BR      DVIN81          ;IF TIMER OUT REPORT IT
6118
6119  035014  012737  000007  002646  DVIN8:    MOV     #7,HDMCC        ;SET POTINTER TO STACK
6120  035022  004737  036432                    JSR     PC,DVIN31               ;AND GO SEND STACK
6121  035026  005737  006646                    TST     TIMERS
6122  035032  001411                            BEQ     DVIN81          ;REPORT ERROR IF TIME OUT
6123  035034  022737  000001  002660            CMP     #1,RHDMCC       ;IS IT ACK RCVD?
6124  035042  001432                            BEQ     DVINEX          ;IF SO EXIT
6125  035044  022737  000007  002660            CMP     #7,RHDMCC       ;IS IT STACK RCVD
6126  035052  001723                            BEQ     DVIN9           ;IF SO SEND ACK
6127  035054  000757                            BR      DVIN8           ;IF NEITHER SEND ANOTHER ACK
6128
6129                                            ;DO ERROR AND REPEAT
6130
6131  035056  012737  017313  006540  DVIN81:   MOV     #DVEM5,TEMP2
6132  035064  013737  002660  006542            MOV     RHDMCC,TEMP3
6133  035072  013737  002646  006544            MOV     HDMCC,TEMP4     ;LOAD UP ERRM. AND REG OUTPUTS
6134  035100  004737  020162                    JSR     PC,LGDVE        ;LOG TIME OUT WAITING FOR RUN
6135  035104  005237  006506                    INC     ERRCNT
6136  035110                                    ERRSOFT 10.,DVEM5,ERR13
6137  035110  104457                                                            TRAP    C$ERSOFT
6138  035112  000012                                                            .WORD   10
6139  035114  017313                                                            .WORD   DVEM5
6140  035116  017636                                                            .WORD   ERR13
6141  035120  005237  006502                    INC     OPVAR           ;COUNT HOW MANY TIMES WE DO THIS.
6142  035124  000137  034306                    JMP     DVINIT          ;TRY ALL OVER AGAIN
6143
```

```
 6144  035130  004737  037010          DVINEX: JSR     PC,CLRRTS       ;CLEAR RTS IF NESC
 6145  035134  042737  173777  006602          BIC     #173777,FLAG    ;CLEAR FLAG WORD
 6146  035142  052737  002000  006602          BIS     #INOVR,FLAG     ;SET THE INITT OVER FLAG
 6147  035150  000207                          RTS     PC              ;RETURN TO CALLER
 6148
 6149
```

```
                                .SBTTL                  DEVICE GET MODEM STATUS SUBROUTINE
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       "DVMODS"   GET MODEM STATUS
                                ;
                                ; IMPLICIT INPUTS:
                                ;       THE BIT POSITION AND AVAILABILITY OF THE MODEM SIGNALS CTS,DSR,...RI,..
                                ;       FOUND IN THE DEPENDENT PORTION OF THE GLOBAL EQUATES SECTION.
                                ;
                                ; OUTPUTS:
                                ;       CURRENT MODEM SIGNAL VALUES IN 'MODS"
                                ;
                                ;
                                ; CALLING SEQUENCE:
                                ;       JSR     PC,DVMODS
                                ;--
035152  017737  154272  007556  DVMODS: MOV     @RXCSR,MODS     ;READ MODEM STATUS
035160  042737  104761  007556  DVMEX:  BIC     #104761,MODS    ;CLEAR BITS NOT RELATING TO MODEM
035166  000207                          RTS     PC              ;RETURN TO CALLER
```

```
                        .SBTTL                  DEVICE QUEUE RECEIVE SPACE SUBROUTINE

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

035170                  DVRXQ:
035170  032737  000004  006602        BIT     #QRX,FLAG               ;ARE WE RECEIVING ?
035176  001434                        BEQ     DVREX                   ;NO,BRANCH
035200  042737  000444  006602        BIC     #QRX+#BCC+#RXM,FLAG     ;CLEAR FLAG FOR RX
035206  005737  011522                TST     RNODE                   ;ITEP MODE ?
035212  001415                        BEQ     DVRX2                   ;NO,BRANCH
035214  052737  000440  006602        BIS     #RXM+#BCC,FLAG          ;GET JUST THE DATA NO CRC.
035222  013737  006470  011510        MOV     DVRXA,RMSGPT            ;RECEIVE DATA BUFFER ADDRESS
035230  012737  000072  011512        MOV     #72,RMSGCC              ;SET UP RX TO GET ITEP MSG.
035236  012737  000070  006472        MOV     #70,DVRCC
035244  000406                        BR      DVRX3

                        ;ENABLE RX, RX INTERRUPTS,AND DATA SET INTERRUPTS

035246  012737  002657  011510  DVRX2:  MOV     #RHDMSG+1,RMSGPT        ;SETUP RX BUFFER ADDRESS
035254  013737  002654  011512        MOV     HDMC,RMSGCC             ;SETUP CHARACTER COUNT
035262  052777  000560  154160  DVRX3:  BIS     #RINTEN!RXENA!#DSITEN!#STRIP,@RXCSR  ;ENABLE RECEIVER
035270  000207            DVREX:  RTS     PC                      ;RETURN TO CALLER
```

```
                        .SBTTL                  DEVICE TRANSMIT AND RECEIVE SUBROUTINE


                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       DVTXRX-DEVICE TRANSMIT AND RECEIVE ROUTINE
                        ;       THIS CODE QUES THE TRANSMIT BUFFER TO THE DEVICE
                        ;       IF NEEDED. THE CODE THEN WAITS FOR A TX COMPLE,
                        ;       RX COMPLETE OR BOTH. THE CODE REPORTS A TIME OUT
                        ;       ERROR IF NO  OUTPUT INTERRUPT IS RECIEVED BEFORE
                        ;       60 SECONDS. AFTER REPORTING ERROR TIMER IS RE STARTED
                        ;       AND DEVICE WILL CONTINUE TO WAIT FOR INTERRUPT.
                        ;
                        ;INPUTS:
                        ;       "DVTXA" = ADDRESS OF TRANSMIT MSG.
                        ;       "DVTCC" = BYTE COUNT OF TRANSMIT MSG.
                        ;    "QTX" BIT  = SET IF TRANSMIT REQUESTED
                        ;    "ETX" BIT  = SET IF TRNASMIT EXPECTED
                        ;    "ERX" BIT  = SET IF RECIEVE EXPECTED
                        ;
                        ;OUTPUTS:
                        ;       "DVTXA" = ADDRESS OF TX MSG. COMPLETED
                        ;       "DVTCC" = BYTE COUNT OF TX MSG. COMPLETED
                        ;       "QTX"   = SET IF TX COMPLETED
                        ;       "DVRXA" = ADDRESS OF RX MSG. COMPLETED
                        ;       "DVRCC" = BYTE COUNT OF RX MSG. COMPLETED
                        ;       "QRX"   = SET IF RX COMPLETED
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;
                        ;               "LGDVE"  - LOG DEVICE ERROR TO EVENT LOG
                        ;
                        ; CALLING SEQUENCE:
                        ;       JSR     PC,DVTXRX
                        ;--

035272  032737  000010  006602  DVTXRX: BIT     #QTX,FLAG               ;ANY TX TO QUE ?(ARE WE TRANSMITTING?)
035300  001444                          BEQ     DVTR3                   ;NO,BRANCH
035302  042737  001030  006602          BIC     #QTX+#TXM+PAD,FLAG      ;CLEAR FLAG
035310  004737  036576                  JSR     PC,CTSSR                ;GO SET CTS
035314  005737  011522                  TST     RNODE                   ;REMOTE NODE = ITEP ?
035320  001412                          BEQ     DVTR1                   ;NO,BRANCH
035322  052737  000020  006602          BIS     #TXM,FLAG               ;TX ONLY DATA MESSAGE
035330  013737  006454  011500          MOV     DVTXA,MSGPTR            ;MESSAGE ADDRESS
035336  013737  006456  011502          MOV     DVTCC,MSGCC             ;MESSAGE CHAR COUNT
035344  000414                          BR      DVTR2                   ;
                                        ;ENABLE TX AND TX INTER.

035346  112737  000201  002645  DVTR1:  MOVB    #201,HDMSG+1            ;SET UP SOH
035354  012737  002645  011500          MOV     #HDMSG+1,MSGPTR         ;SET POINTER TO HEADER
035362  013737  006456  002646          MOV     DVTCC,HDMCC
035370  013737  002654  011502          MOV     HDMC,MSGCC              ;SET CC FOR HEADER
035376  012737  000177  011504  DVTR2:  MOV     #177,SYNCC              ;SET UP FOR 177 SYNCS.
                                                                        ;(SO MANY SYNCS ARE NECESSARY
                                                                        ;;IF THE OTHER NODE HAS A
```

```
                                                        ;;;SLOWER CPU.)
                                                        ;TURN ON TX
035412  012737  000074  006646  DVTR3:  MOV     #60.,TIMERS             ;SET TIMER FOR 60 SECS
035420                          DVTR8:  BREAK
035420  104422                                                          TRAP    C$BRK
035422  005737  006646                  TST     TIMERS                  ;IS TIMER EXPIRED
035426  001022                          BNE     TOINOT
                                        ;LOG ERROR TIME OUT RX OR TX NOT COMPLETED
035430  012737  017110  006540          MOV     #DVEM2,TEMP2
035436  017737  154006  006542          MOV     @RXCSR,TEMP3
035444  017737  154006  006544          MOV     @TXCSR,TEMP4
035452  004737  020162                  JSR     PC,LGDVE
035456  005237  006506                  INC     ERRCNT
035462                                  ERRSOFT 7,DVEM2,ERR13
035462  104457                                                          TRAP    C$ERSOFT
035464  000007                                                          .WORD   7
035466  017110                                                          .WORD   DVEM2
035470  017636                                                          .WORD   ERR13
035472  000747                          BR      DVTR3           ;RETURN TO CHECK TIMER
035474  032737  000010  006602  TOINOT: BIT     #QTX,FLAG       ;IS IT TX COMPL?
035502  001406                          BEQ     DVTR4           ;BRANCH IF TX NOT DONE.
035504  004737  037010                  JSR     PC,CLRRTS
035510  032737  000100  006602          BIT     #ERX,FLAG       ;ARE WE EXPECTING TO RX
035516  001416                          BEQ     DVTREX          ;BRANCH IF NOT.
035520  032737  000004  006602  DVTR4:  BIT     #QRX,FLAG       ;IS RX DONE
035526  001734                          BEQ     DVTR8           ;GO BACK AND TIME IF NOT
035530  032737  000200  006602          BIT     #ETX,FLAG       ;ARE WE EXPECTG TO TX.
035536  001406                          BEQ     DVTREX          ;BRANCH IF NOT.
035540  032737  000010  006602          BIT     #QTX,FLAG       ;IS IT TX COMPLETED
035546  001724                          BEQ     DVTR8           ;GO BACK AND TIME OUT
035550  004737  037010                  JSR     PC,CLRRTS       ;CLEAR RTS IF NESC.
035554  000207          DVTREX: RTS     PC                      ;AND EXIT
```

```
                        ; DEVICE DEPENDENT SUBROUTINES



                        .SBTTL                  DEVICE INTERRUPT SERVICE ROUTINES


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

035556                          BGNSRV  DVRXI
035556                                                                  DVRXI::
035556  010246                  MOV     R2,-(SP)        ;SAVE R2
035560  017737  153664  011474  MOV     @RXCSR,IRXCSR   ;MOV RX CSR TO IMAGE
035566  032737  000010  006576  BIT     #MOCHK,PARAM    ;ANY MODEM CHANGES TO REPORT
035574  001447                  BEQ     RXIN21          ;IF NOT IGNORE DS CHANGE.
035576  032737  002000  006602  BIT     #INOVR,FLAG     ;IS INIT OVER
035604  001443                  BEQ     RXIN21          ;NO THEN IGNORE DS CHANGE.
035606  005737  011474          TST     IRXCSR
035612  100040                  BPL     RXIN21          ;IF DATA SET CHANGE IS NOT SET BR
035614  013737  011474  011472  MOV     IRXCSR,CMODS    ;MOV THE NEW MODEM STATUS IN
035622  042737  104761  011472  BIC     #104761,CMODS   ;CLEAR BITS NOT RELATING TO MODEM STATUS
```

```
 6369  035630  013737  011472  006542  RXIN2:  MOV     CMODS,TEMP3
 6370  035636  013737  007556  006544          MOV     MODS,TEMP4
 6371  035644  023737  006544  006542          CMP     TEMP4,TEMP3     ;COMPARE OLD TO CURRENT
 6372  035652  001406                          BEQ     10$             ;INC GLITCH COUNT
 6373  035654  005237  011520                  INC     MHRCNT          ;INC HARD COUNT
 6374  035660  012737  016717  006540          MOV     #HRDMSG,TEMP2   ;SET UP HARD MESG.
 6375  035666  000405                          BR      RXIN1
 6376  035670  005237  011516          10$:    INC     MGLCNT          ;INC GLITCH COUNT
 6377  035674  012737  016671  006540          MOV     #GLMSG,TEMP2    ;SET UP GLITCH
 6378  035702  004737  020336          RXIN1:  JSR     PC,LOGMSC       ;GO LOG MODEM STATUS CHANGE
 6379  035706  013737  011472  007556          MOV     CMODS,MODS      ;MOVE CURRENT TO OLD
 6380                                          ;
 6381                                          ;TEST FOR DATA
 6382
 6383  035714  032737  000200  011474  RXIN21: BIT     #RXDONE,IRXCSR  ;RX DONE ?
 6384  035722  001540                          BEQ     RXINEX          ;NO,BRANCH
 6385  035724  017737  153524  011476          MOV     @RXDBUF,IRXDBUF ;READ DATA
 6386  035732  032737  100000  011476          BIT     #RERR,IRXDBUF   ;OVERRUN ERROR ?
 6387  035740  001055                          BNE     RXIN3           ;YES,BRANCH
 6388
 6389                                          ;GET HERE WITH GOOD DATA
 6390
 6391  035742  013702  011510  RXIN4:  MOV     RMSGPT,R2       ;SET RX MESSAGE POINTER
 6392  035746  113722  011476          MOVB    IRXDBUF,(R2)+   ;STORE DATA AWAY
 6393  035752  010237  011510          MOV     R2,RMSGPT       ;SAVE UPDATED MESSAGE POINTER
 6394
 6395
 6396  035756  005337  011512          DEC     RMSGCC          ;ALL DATA RECEIVED ?
 6397  035762  001120                  BNE     RXINEX          ;NO,BRANCH
 6398  035764  032737  000400  006602  BIT     #BCC,FLAG       ;CHECK CRC ?
 6399  035772  001426                  BEQ     RXIN6           ;YES,BRANCH
 6400  035774  032737  010000  011476  BIT     #CRCOK,IRXDBUF  ;CRC GOOD ?
 6401  036002  001056                  BNE     RXIN5           ;YES,BRANCH
 6402  036004  013737  011476  006542  MOV     IRXDBUF,TEMP3   ;SET UP TO
 6403  036012  013737  011474  006544  MOV     IRXCSR,TEMP4    ;;LOG AND
 6404  036020  012737  017205  006540  MOV     #DVEM3,TEMP2    ;;;PRINT CRC ERROR
 6405  036026  004737  020162          JSR     PC,LGDVE        ;LOG ERROR
 6406  036032  005237  006506          INC     ERRCNT          ;BUMP COUNT
 6407  036036                          ERRSOFT 8,DVEM3,ERR13   ;PRINT ERROR TO USER
 6408  036036  104457                                                  TRAP    C$ERSOFT
 6409  036040  000010                                                  .WORD   8
 6410  036042  017205                                                  .WORD   DVEM3
 6411  036044  017636                                                  .WORD   ERR13
 6412
 6413  036046  000463                  BR      RXIN8           ;DISABLE INTERRUPTS AND EXIT
 6414
 6415                                  ;;;IN ORDER TO CHECK CRC, WE MUST READ 2 MORE CHARACTERS(CRC)
 6416  036050  052737  000400  006602  RXIN6:  BIS     #BCC,FLAG       ;SET CRC ALREADY CHECKED FLAG
 6417  036056  012737  000002  011512          MOV     #2,RMSGCC       ;COUNT TWO CHARACTERS
 6418  036064  012737  011514  011510          MOV     #BCCW,RMSGPT    ;CRC STORAGE ADDRESS
 6419  036072  000454                          BR      RXINEX          ;EXIT
 6420
 6421
 6422  036074                          RXIN3:  ;LOG OVERRUN ERROR
 6423
 6424  036074  012737  017246  006540          MOV     #DVEM4,TEMP2
```

```
 6425  036102  013737  011476  006542          MOV     IRXDBUF,TEMP3
 6426  036110  013737  011474  006544          MOV     IRXCSR,TEMP4
 6427  036116  004737  020162                  JSR     PC,LGDVE
 6428  036122  005237  006506                  INC     ERRCNT
 6429  036126                                  ERRSOFT 9,DVEM4,ERR13
 6430  036126  104457                                                          TRAP    C$ERSOFT
 6431  036130  000011                                                          .WORD   9
 6432  036132  017246                                                          .WORD   DVEM4
 6433  036134  017636                                                          .WORD   ERR13
 6434  036136  000424                          BR      RXIN7
 6435
 6436  036140  032737  000040  006602  RXIN5:  BIT     #RXM,FLAG       ;IS THE RX M BODY BIT SET
 6437  036146  001020                          BNE     RXIN7           ;IF YES THEN ALL DONE
 6438  036150  052737  000040  006602          BIS     #RXM,FLAG
 6439  036156  042737  000400  006602          BIC     #BCC,FLAG       ;CLEAR BCC AND SET RXM
 6440  036164  013737  006470  011510          MOV     DVRXA,RMSGPT    ;MOVE ADDRESS TO POINTER
 6441  036172  013737  002660  011512          MOV     RHDMCC,RMSGCC   ;MOVE THE CHAR COUNT IN
 6442  036200  013737  002660  006472          MOV     RHDMCC,DVRCC    ;SET THE CC TO AMOUNT IN HEADER
 6443  036206  000406                          BR      RXINEX          ;AND FINISH.
 6444
 6445  036210  052737  000004  006602  RXIN7:  BIS     #QRX,FLAG       ;SET MESSAGE RECEIVED IN FLAG
 6446
 6447  036216  042777  000120  153224  RXIN8:  BIC     #RINTEN+RXENA,@RXCSR    ;CLEAR INTAND RX ENABLE
 6448
 6449  036224  012602                  RXINEX: MOV     (SP)+,R2        ;RESTORE R2
 6450  036226                                  ENDSRV
 6451  036226                                                                  L10020:
 6452  036226  000002                                                          RTI
```

```
                        .SBTTL                  DEVICE TRANSMIT INTERRUPT ROUTINE

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

036230                          BGNSRV  DVTXI
036230                                                                  DVTXI::
036230  010246                  MOV     R2,-(SP)        ;SAVE R2
036232  005737  011504          TST     SYNCC           ;ANY SYNCS TO SEND
036236  001406                  BEQ     TXIN1           ;IF NOT GO TO 1
036240  013777  011506  153212  MOV     SYNCW,@TXDBUF   ;ELSE SET TSOM AND SYNC WORD
036246  005337  011504          DEC     SYNCC           ;DEC SYNC COUNT
036252  001065                  BNE     TXINEX          ;IF NOT ZERO EXIT
036254  032737  001000  006602  TXIN1:  BIT     #PAD,FLAG       ;IS THE PAD BIT SET
036262  001414                  BEQ     TXIN2           ;GO TO 2 IF NOT SET
036264  012777  000377  153166  MOV     #377,@TXDBUF    ;LOAD FF TO TX DATA REG.
036272  042777  000120  153156  BIC     #TINTEN!SEND,@TXCSR     ;CLEAR TX INT ENABLE
036300  005237  037126          INC     TXREADY         ;TELL PROTOCOL MODULE WE'RE DONE
036304  052737  000010  006602  BIS     #QTX,FLAG       ;SET THE TX COMPLETE
036312  000445                  BR      TXINEX          ;AND EXIT
036314  005737  011502  TXIN2:  TST     MSGCC           ;ALL DATA SENT ?
036320  001416                  BEQ     TXIN4           ;YES,BRANCH
036322  005037  006556          CLR     DATAWORD        ;BE SURE ITS CLEAR
036326  013702  011500          MOV     MSGPTR,R2       ;LOAD R2 WITH TX BUFFER POINTER ADDR.
036332  112237  006556          MOVB    (R2)+,DATAWORD  ;PUT DATA IN LOW BYTE
036336  010237  011500          MOV     R2,MSGPTR       ;RESTORE UPDATED POINTER
```

```
 6509  036342  013777  006556  153110          MOV     DATAWORD,@TXDBUF;HI BYTE TSOM=0.. LO BYTE = DATA
 6510  036350  005337  011502                  DEC     MSGCC           ;BUMP CHAR COUNT
 6511  036354  000424                          BR      TXINEX          ;
 6512  036356  012777  001000  153074  TXIN4:  MOV     #TEOM,@TXDBUF   ;SEND CRC CHARACTER
 6513  036364  032737  000020  006602          BIT     #TXM,FLAG       ;IS THIS THE END OF DATA MSG.
 6514  036372  001012                          BNE     TXIN3           ;IF SO SET THE PAD BIT
 6515  036374  052737  000020  006602          BIS     #TXM,FLAG       ;IF NOT MUST BE END OF HEADER
 6516  036402  013737  006454  011500          MOV     DVTXA,MSGPTR    ;SO SET UP MSGPTR FOR MSG
 6517  036410  013737  006456  011502          MOV     DVTCC,MSGCC     ;AND THE CC FOR MSG.
 6518  036416  000403                          BR      TXINEX
 6519  036420  052737  001000  006602  TXIN3:  BIS     #PAD,FLAG       ;SET THE PAD BIT
 6520
 6521  036426  012602                  TXINEX: MOV     (SP)+,R2        ;RESTORE R2
 6522  036430                                  ENDSRV
 6523  036430                                                                  L10021:
 6524  036430  000002                                                                 RTI
```

```
                        .SBTTL                  DEVICE TRANSMIT CONTROL MSG
                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       THIS ROUTINE DOES THE FOLLOWING
                        ;       QUES A RX SPACE AT RHDMSG+1
                        ;       QUES A TX MSG FROM HDMSG+1
                        ;       CHECKS FOR A TIMER EXPIRED
                        ;       IF EXPIRED RETURN TO CALLER
                        ;       ELSE CHECK FOR A TX MSG COMPLETED
                        ;       IF TX COMPLETED CHECK FOR RX COMPLETED
                        ;       ELSE RECHECK TIMER AND TX COMPLETED UNTIL
                        ;       EITHER TX COMPLETE OR TIME OUT
                        ;       IF TX COMPLETE AND RX NOT COMPLETE THEN
                        ;       REQUE TX MSG.
                        ;       ELSE IF RX COMPLETE RETURN.
                        ;
                        ; INPUTS:
                        ;       TXM             - SET IN FLAG WORD
                        ;       HDMSG+2         - TYPE OF CONTROL MSG..
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;               "CLRRTS"        - CLEAR REQUEST TO SEND IF HALF DUP.
                        ; CALLING SEQUENCE:
                        ;               JSR     PC,DVIN31
                        ; RETURN:
                        ;       RETURN TO CALLER IF SOMETHING RX'D OR TIMER OUT.
                        ;
                        ;--
036432  042737  000404  006602  DVIN31: BIC     #QRX!#BCC,FLAG          ;CLEAR RX COMPLETE & CRC ALREADY CHECK
036440  012737  002657  011510          MOV     #RHDMSG+1,RMSGPT        ;SET UP POINTER
036446  013737  002654  011512          MOV     HDMC,RMSGCC             ;AND CC
036454  052777  000560  152766          BIS     #RINTEN!RXENA!DSITEN!STRIP,@RXCSR  ;TURN ON RX
                                        ;SET UP TXMITTER TO SEND
036462  004737  036576          DVIN32: JSR     PC,CTSSR                ;SET RTS .
036466  042737  001010  006602          BIC     #QTX!PAD,FLAG           ;CLEAR TX COMPT FLAG.
036474  012737  002645  011500          MOV     #HDMSG+1,MSGPTR         ;MOVE THE CURRENT POINTER TO MSGPTR.
036502  013737  002654  011502          MOV     HDMC,MSGCC
036510  012737  000010  011504          MOV     #8.,SYNCC               ;SET UP SYNC COUNT
036516  052777  000120  152732          BIS     #SEND!TINTEN,@TXCSR     ;TURN ON TX
                                        ;NOW WAIT FOR TIME OUT OR TX COMPLETE
036524                          DVIN35: BREAK                                   TRAP    C$BRK
036524  104422
036526  005737  006646                  TST     TIMERS          ;IS IT TIMED OUT
036532  001420                          BEQ     DVIN34          ;IF YES EXIT
036534  032737  000010  006602          BIT     #QTX,FLAG       ;IS TX DONE
036542  001770                          BEQ     DVIN35          ;IF NOT GO BACK AND CK TIME OUT
036544  004737  037010                  JSR     PC,CLRRTS       ;CLEAR RTS IF HALF DUPLEX
036550  012737  002000  006536          MOV     #2000,TEMP1     ;WAIT FOR RX TO COMPLETE
```

```
 6581  036556  005337  006536          DVIN36: DEC     TEMP1           ;BUMP COUNTER
 6582  036562  001375                          BNE     DVIN36          ;DO IT AGAIN
 6583  036564  032737  000004  006602          BIT     #QRX,FLAG       ;DID WE RX ANYTHING
 6584  036572  001733                          BEQ     DVIN32          ;IF NOT RETRANSMIT LAST
 6585  036574  000207                  DVIN34: RTS     PC              ;RETURN TO CALLER
 6586
```

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

036576  022737  000001  006572  CTSSR:  CMP     #1,MLTYP        ;IS THIS TTL LOOP
036604  001500                          BEQ     DVTXR9          ;BR IF YES
                                        ;SET RTS AND WAIT FOR CTS
036606  032737  004000  006602  DVTXR3: BIT     #FIRST,FLAG
036614  001014                          BNE     CTSS3           ;IF NOT FIRST TIME SKIP DELY
036616  012737  177777  006534          MOV     #-1,TEMP
036624  005237  006534          CTSS4:  INC     TEMP
036630                                  BREAK
036630  104422                                                  TRAP    C$BRK
036632  005737  006534                  TST     TEMP
036636  001372                          BNE     CTSS4           ;IF NOT ZERO GO BACK
036640  052737  004000  006602          BIS     #FIRST,FLAG     ;SET FIRST FLAG.
036646  012737  001000  006642  CTSS3:  MOV     #1000,TIMER1    ;SET UP TIMER FOR 1000 TICKS
036654  005737  006574                  TST     FHDPLX          ;FULL DUPLEX ?
036660  001012                          BNE     CTSS7           ;YES,BRANCH
036662  004737  035152          10$:    CALL    DVMODS          ;GET MODEM STATUS
036666  032737  010000  007556          BIT     #DCD,MODS       ;CARRIER DETECTED?
036674  001404                          BEQ     CTSS7           ;NO,BRANCH
036676  005737  006642                  TST     TIMER1          ;TIME DONE ?
036702  001417                          BEQ     DVTXR4          ;YES,BRANCH
036704  000766                          BR      10$             ;TRY AGAIN

036706  052777  000004  152534  CTSS7:  BIS     #RTS,@RXCSR     ;SET REQUEST TO SEND
036714  012737  001750  006642          MOV     #1000.,TIMER1   ;SET UP TIMER
036722                          DVTXR2: BREAK
036722  104422                                                  TRAP    C$BRK
036724  032777  020000  152516          BIT     #CTS,@RXCSR     ;IS CLEAR TO SEND BACK
036732  001025                          BNE     DVTXR1          ;BR. IF CTS IS SET
036734  005737  006642                  TST     TIMER1          ;ELSE TEST IF TIME EXPIRED
036740  001370                          BNE     DVTXR2          ;BR IF TIME NOT EXPRIED.

                                        ;SET ERROR FOR NO CTS

036742  012737  017026  006540  DVTXR4: MOV     #DVEM1,TEMP2
```

DEVICE RTS TO CTS DELAY

```
 6643  036750  017737  152474  006542          MOV     @RXCSR,TEMP3
 6644  036756  017737  152474  006544          MOV     @TXCSR,TEMP4
 6645  036764  004737  020162                  JSR     PC,LGDVE
 6646  036770  005237  006506                  INC     ERRCNT
 6647  036774                                  ERRSOFT 6,DVEM1,ERR13
 6648  036774  104457                                                  TRAP    C$ERSOFT
 6649  036776  000006                                                  .WORD   6
 6650  037000  017026                                                  .WORD   DVEM1
 6651  037002  017636                                                  .WORD   ERR13
 6652  037004  000700                  BR      DVTXR3          ;THEN TRY TO SET RTS AGAIN
 6653  037006                  DVTXR1:
 6654  037006  000207          DVTXR9: RTS     PC              ;
```

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

037010                                  CLRRTS:
037010  005737  037132                          TST     MPPTP           ;MULTI-POINT ?
037014  001003                                 BNE     20$             ;YES,BRANCH
037016  005737  006574                          TST     FHDPLX          ;IS THIS FULL DUPLEX
037022  001012                                 BNE     DVTR5           ;BRANCH IF YES
037024  017737  152426  006540    20$:          MOV     @TXCSR,TEMP2    ;GET RX STATUS
037032  032737  001000  006540                  BIT     #TXACT,TEMP2    ;ALL DATA SENT ?
037040  001371                                 BNE     20$             ;NO,WAIT
037042  042777  000004  152400                  BIC     #RTS,@RXCSR     ;CLEAR REQUEST TO SEND
037050  000207                          DVTR5:  RTS     PC              ;RETURN TO CALLER
```

```
                                        .SBTTL  DDCMP PROTOCOL MODULE
                                ;*****************************************************************************
                                        ; DCLT DDCMP PROTOCOL MODULE

                                        .EVEN
                                ;; LOCAL STORAGE

                                ;; TABLE OF STATISTICS AND ERRORS
                                ;;; NOTE: KEEP THE VARIABLES TOGETHER AND IN SEQUENCE
                                ;;;       OTHERWISE THE RPT> ROUTINE WILL PRINT WRONG INFO.

037052  000000                  PRSTAT: .WORD 0                 ;STATUS FLAGS
                                                                ;BIT0 = BCCOK
                                                                ;BIT1 = BCCBAD
                                                                ;BIT2 = SNAK
                                                                ;BIT3 = SACK
                                                                ;BIT4 = SDATA
                                                                ;SPARE
                                                                ;BIT6 = RXD
                                                                ;BIT7 = SPARE
                                                                ;BIT8 = NAKRX
                                                                ;BIT9 = MYDATA
                                                                ;BIT10 = SSTACK
                                                                ;BIT11 =SSTART
037054  000000                  TMESTX: .WORD 0                 ;TOTAL DATA MESSAGES TRANSMITTED(16 BIT COUNTER
037056  000000                  TMESRX: .WORD 0                 ;TOTAL DATA MESSAGES RECEIVED(16 BIT COUNTER)

037060      000                 N:      .BYTE 0                 ;# OF HIGHEST SEQUENTIAL DATA MESSAGE TRANS
                                                                ;; MITTED BY THIS STATION
037061      000                 A:      .BYTE 0                 ;# OF THE HIGHEST SEQUENTIAL DATA MESSAGE
                                                                ;; THAT HAS BEEN ACKNOWLEDGE TO THIS STATION

037062      000                 T:      .BYTE 0                 ;# OF THE NEXT DATA MESSAGE TO BE TRANSMITTED
037063      000                 X:      .BYTE 0                 ;LAST MESSAGE NUMBER TRANSMITTED

037064      000                 R:      .BYTE 0                 ;LAST MESSAGE RECEIVED
037065      000                 TRIBN:  .BYTE 0                 ;TRIB ADDRESS  PT TO PT = 1

                                ;; ERROR COUNTERS
037066      000                 REMTMO: .BYTE 0                 ;REMOTE REPLY TIMEOUTS(ACKS SENT NUM=R)
037067      000                 GLOBCC: .BYTE 0                 ;GLOBAL CRC ERRORS

037070      000                 REANAK: .BYTE 0                 ;REASON FOR LAST NAK SENT
037071      000                 SELTHER:.BYTE 0                 ;SELECTION THRESHOLD ERROR

037072      000                 RXTHER: .BYTE 0                 ;RECEIVE THRESHOLD ERRORS
037073      000                 TXTHER: .BYTE 0                 ;TRANSMIT THRESHOLD ERRORS
```

```
6724
6725  037074  000          DEROUT: .BYTE 0          ;DATA ERRORS OUTBOUND (NAKS RECEIVED
6726  037075  000          OUTMASK:.BYTE 0          ;; REASONS = 1,2,OR 3)
6727                                                ;MASK VALUES -- BIT0 = HEADER CRC ERROR
6728                                                ;             -- BIT1 = DATA FIELD CRC ERROR
6729                                                ;             -- BIT2 = REP RESPONSE NUM<>R
6730
6731  037076  000          DERIN:  .BYTE 0          ;DATA ERRORS INBOUND (NAKS TRANSMITTED
6732  037077  000          INMASK: .BYTE 0          ;; REASONS = 1,2,OR 3)
6733                                                ;MASK VALUES -- BIT0 = HEADER CRC ERROR
6734                                                ;             -- BIT1 = DATA FIELD CRC ERROR
6735                                                ;             -- BIT2 = REP RESPONSE NUM<>R
6736
6737  037100  000          LBUFER: .BYTE 0          ;LOCAL BUFFER ERRORS (NAKS SENT
6738  037101  000          LBMASK: .BYTE 0          ;; REASONS = 8. OR 16.)
6739                                                ;MASK VALUES -- BIT0 = BUFFER NOT AVAILABLE
6740                                                ;             -- BIT1 = MESSAGE TOO LONG
6741
6742  037102  000          RBUFER: .BYTE 0          ;REMOTE BUFFER ERRORS (NAKS RECEIVED
6743  037103  000          RBMASK: .BYTE 0          ;REASONS 8. OR 16.)
6744                                                ;MASK VALUES -- BIT0 = BUFFER NOT AVAILBLE
6745                                                ;             -- BIT1 = MESSAGE TOO LONG
6746
6747  037104  000          RMSTER: .BYTE 0          ;REMOTE STATION ERRORS (NAKS RECEIVED
6748  037105  000          RMMASK: .BYTE 0          ;REASON 9. OR 17.)
6749                                                ;MASK VALUES-- BIT0 = RECEIVER OVERRUN
6750                                                ;              BIT1 = FORMAT ERROR
6751
6752  037106  000          LOSTER: .BYTE 0          ;LOCAL STATION ERRORS (NAKS SENT
6753  037107  000          LSMASK: .BYTE 0          ;; REASON 9. OR 17.)
6754                                                ;MASK VALUES -- BIT0 = RECEIVER OVERRUN
6755                                                ;             -- BIT1 = FORMAT ERROR
6756
6757  037110  000000       RXTXTE: .WORD 0          ;RX AND TX THRESHOLD ERRORS (OVERFLOWS)
6758  037112  000          SPARE0: .BYTE 0
6759  037113  000          SPARE1: .BYTE 0
6760  037114  000000       PROEND: .WORD 0          ;END OF PROTOCOL COUNTERS
6761  037116  000000       IMFLAG: .WORD 0          ;IMAGE OF MAIN CODE FLAG WORD
6762  037120  000000       RXPRC:  .WORD 0          ;-1 = MESSAGE RX'ED & 'ACK' SENT
6763  037122  000000       TXPRC:  .WORD 0          ;-1 = MESSAGE TX'ED & 'ACK' RECEIVED
6764  037124  000000       ASTRT:  .WORD 0          ;-1 = STACK SENT
6765  037126  000000       TXREADY:.WORD 0          ; 1 = READY TO SEND ANOTHER MESSAGE
6766  037130  000000       PRUN:   .WORD 0          ; 1 = PROTOCOL RUNNING. USED IN THIS MODULE
6767  037132  000000       MPPTP:  .WORD 0          ; 1 = MULTI POINT NETWORK
6768  037134  000000       SELECT: .WORD 0          ; 1 =THIS STATION CAN NOW TRANSMIT(HALF/DUPLEX)
6769  037136  000000       IMPRSTAT:.WORD 0         ; COPY OF PROTOCOL STATUS WORD
6770  037140  000000       PRFLAG: .WORD 0          ; USED TO COMMUNICATE WITH RX INTER. ROUTINE
6771  037142  000000       HDXMTP: .WORD 0          ; 1 = HALF DUPLEX OR MULTI-POINT
6772  037144  000000       PRTEMP: .WORD 0          ; TEMPORARY WORK LOCATION
6773  037146  000000       TURNON: .WORD 0          ; 1 = RECEIVER IS ALREADY ON
6774  037150  000000       TIMEOUT: .WORD 0         ; 20 = PRINT 'TX OR RX NOT COMPLETE'
6775
```

```
6776
6777
6778                            ;; NAK REASONS VALUES AS USED IN NAK CONTROL MESSAGES
6779
6780    000001                          HEADBCC = 1             ;HEADER BCC ERROR
6781    000002                          DATABCC = 2             ;DATA BCC ERROR
6782    000003                          REPSENT = 3             ;REP RESPONSE
6783    000010                          BUFFNA  = 10            ;BUFFER TEMPORARILY NOT AVAILABLE
6784    000011                          RXOVRUN = 11            ;RECEIVER OVERRUN
6785    000020                          MESLONG = 20            ;MESSAGE TOO LONG
6786    000021                          FORMERR = 21            ;HEADER FORMAT ERROR
6787
6788
6789                            ;; ADDITIONAL NAK BIT MASKS AS USED IN COUNTERS
6790
6791    000004                          REPMSK = BIT2           ;REPLY RESPONSE
6792    000001                          RXOVMSK= BIT0           ;RECEIVER OVERRUN
6793    000002                          FMTMSK = BIT1           ;FORMAT ERROR
6794    000002                          MTLMSK = BIT1           ;MESSAGE TOO LONG
6795    000001                          BNAMSK = BIT0           ;BUFFER NOT AVAILABLE
6796
6797                            ;; MESSAGE TYPE DEFINITIONS
6798
6799    000201                          SOH = 201               ;DATA MESSAGE
6800    000144                          MAINT = 144             ;MAINTENANCE MESSAGE
6801    000005                          ENQ = 5                 ;CONTROL MESSAGE
6802
6803
6804                            ;;SUBTYPES OF CONTROL MESSAGES
6805
6806    000001                          ACK = 1                 ;ACKNOWLEDGE MESSAGE
6807    000002                          NAK = 2                 ;NEGATIVE ACKNOWLEDGE MESSAGE
6808    000003                          REP = 3                 ;REPLY TO MESSAGE NUMBER
6809    000006                          STRT = 6                ;START MESSAGE
6810    000007                          STACK = 7               ;START ACKNOWLEDGE MESSAGE
6811
6812
6813                            ;;STATUS WORD BIT DEFINITIONS
6814
6815    000001                          BCCOK = BIT0            ;BCC CHECKED GOOD
6816    000002                          BCCBAD = BIT1           ;BCC CHECKED BAD
6817    000004                          SACK = BIT2             ;SEND ACK
6818    000010                          SNAK = BIT3             ;SEND NAK
6819    000020                          SDATA = BIT4            ;SEND DATA
6820    000100                          RXD = BIT6              ;RECEIVER DONE
6821    000400                          NAKRX = BIT8            ;NAK RECEIVED
6822    001000                          MYDATA = BIT9           ;MY DATA
6823    002000                          SSTACK = BIT10          ;SEND START ACKNOWLEDGE
6824    004000                          SSTART = BIT11          ;SEND START
6825
```

```
                        ;*****************************************************************************
                        ;       THE BELOW TABLES AND ASCIZ MESSAGES ARE USED IN DCLT
                        ;               REPORTING OF ERROR COUNTERS. THEY MUST REMAIN IN THE
                        ;               CURRENT SEQUENCE ELSE WE'LL BE REPORTING ERRONEOUS
                        ;               DATA.
                        ;*****************************************************************************

037152  037262  STALST: .WORD   STA0A          ;POINTER FOR OFFSET 0 ASCII
037154  037311           .WORD   STA1A          ;POINTER FOR OFFSET 1 ASCII
037156  037347           .WORD   STA2A          ;POINTER FOR OFFSET 2 ASCII
037160  037403           .WORD   STA3A          ;POINTER FOR OFFSET 3 ASCII
037162  037475           .WORD   STA4A          ;POINTER FOR OFFSET 4 ASCII
037164  037561           .WORD   STA5A          ;POINTER FOR OFFSET 5 ASCII
037166  037641           .WORD   STA6A          ;POINTER FOR OFFSET 6 ASCII
037170  037725           .WORD   STA7A          ;POINTER FOR OFFSET 7 ASCII
037172  040007           .WORD   STA10A         ;POINTER FOR OFFSET 10 ASCII
037174  040071           .WORD   STA11A         ;POINTER FOR OFFSET 11 ASCII
037176  040166           .WORD   STA12A         ;POINTER FOR OFFSET 12 ASCII
037200  040262           .WORD   STA13A         ;POINTER FOR OFFSET 13 ASCII
037202  040360           .WORD   STA14A         ;POINTER FOR OFFSET 14 ASCII
037204  040456           .WORD   STA15A         ;POINTER FOR OFFSET 15 ASCII
037206  040544           .WORD   STA16A         ;POINTER FOR OFFSET 16 ASCII
037210  040631           .WORD   STA17A         ;POINTER FOR OFFSET 17 ASCII

                        ;TABLE FOR PRINT ROUTINES
                        ;PRIW: WORD ROUTINE
                        ;PRIBB: BYTE/BYTE ROUTINE
                        ;PRIBS: BYTE SPECIAL ROUTINE

037212  021642  STAIND: .WORD   PRIW
037214  021642           .WORD   PRIW
037216  021642           .WORD   PRIW
037220  021670           .WORD   PRIBB
037222  021670           .WORD   PRIBB
037224  021670           .WORD   PRIBB
037226  021670           .WORD   PRIBB
037230  021670           .WORD   PRIBB
037232  021670           .WORD   PRIBB
037234  021726           .WORD   PRIBS
037236  021726           .WORD   PRIBS
037240  021726           .WORD   PRIBS
037242  021726           .WORD   PRIBS
037244  021726           .WORD   PRIBS
037246  021726           .WORD   PRIBS
037250  021642           .WORD   PRIW

037252  000000  LAST:    .WORD   0              ;LAST MESSAGE TO PRINT
037254  000000  FIR:     .WORD   0              ;FIRST MESSAGE TO PRINT
037256  000000  MES:     .WORD   0              ;HOLDS MESSAGE
037260  000000  MESDATA: .WORD   0              ;DATA PART OF MESSAGE
```

```
6879
6880                                    ;*****************************************************************************
6881                                    ;       THE BELOW ASCIZ MESSAGES USED IN 'RPT>' LEVEL OF DCLT
6882                                    ;
6883                                    .NLIST BEX
037262  047045  047445  022466  STA0A:  .ASCIZ  /%N%06%S2%ASTATUS FLAGS/
037311    045  022516  033117  STA1A:  .ASCIZ  /%N%06%S2%ADATA MSGS. TX'MITTD/
037347    045  022516  033117  STA2A:  .ASCIZ  /%N%06%S2%ADATA MSGS. RX'CVD/
037403    045  022516  031517  STA3A:  .ASCIZ  /%N%03%S5%AHIGHEST MSG # TX'D%N%03%S5%AHIGHEST MSG # ACK'D/
037475    045  022516  031517  STA4A:  .ASCIZ  /%N%03%S5%ANEXT MSG # TO TX%N%03%S5%ALAST MSG # TX'D/
037561    045  022516  031517  STA5A:  .ASCIZ  /%N%03%S5%AHIGHEST MSG # RX'D%N%03%S5%ATRIB ADDR/
037641    045  022516  031517  STA6A:  .ASCIZ  /%N%03%S5%AREMOTE TIME OUTS%N%03%S5%AGLOBAL CRC ERRS/
037725    045  022516  031517  STA7A:  .ASCIZ  /%N%03%S5%ANAK REASON%N%03%S5%ASELECT THRESH. ERRS/
040007    045  022516  031517  STA10A: .ASCIZ  /%N%03%S5%ARX THRESH ERRS%N%03%S5%ATX THRESH. ERRS/
040071    045  022516  031517  STA11A: .ASCIZ  /%N%03%S5%ADATA ERRORS OUT%N%S8%AHBCC %01%A BCC %01%A REP %01/
040166  047045  047445  022463  STA12A: .ASCIZ  /%N%03%S5%ADATA ERRORS IN%N%S8%AHBCC %01%A BCC %01%A REP %01/
040262  047045  047445  022463  STA13A: .ASCIZ  /%N%03%S5%ALOCAL BUFFER ERRS%N%S8%A NO BUFF %01%A TOO BIG  %01/
040360  047045  047445  022463  STA14A: .ASCIZ  /%N%03%S5%AREMOTE BUFFER ERRS%N%S8%A NO BUFF %01%A TOO BIG %01/
040456  047045  047445  022463  STA15A: .ASCIZ  /%N%03%S5%AREMOTE STA ERRS%N%S8%AOVRN %01%A FORMAT %01/
040544  047045  047445  022463  STA16A: .ASCIZ  /%N%C3%S5%ALOCAL STA ERRS%N%S8%AOVRN %01%A FORMAT %01/
040631    045  022516  033117  STA17A: .ASCIZ  /%N%06%S2%ATX & RX THRESHOLD ERRORS(OVERFLOW)/
                                        .EVEN
                                    .LIST BEX
6884                                    ;*****************************************************************************
6885
6886
6887
6888
6889
6890
6891
6892
6893
6894
6895
```

```
                        ;*****************************************************************************
                        ; PROTOCOL ROUTINE:
                        ;
                        ;       DESCRIPTION: IF THE USER SPECIFIES THE '/PROTOCOL' SWITCH THIS
                        ;                    ROUTINE WILL BE CALLED. THIS ROUTINE DECIDES IF
                        ;                    WE ARE TRANSMITTING AND/OR RECEIVING AND CALLS
                        ;                    THE NECESSARY ROUTINES.
                        ;                 THIS CODE WAS WRITTEN ONLY TO BE USED WITH DCLT.
                        ;
                        ;*****************************************************************************

040706  013737  006602  037116  PROTOC: MOV     FLAG,IMFLAG     ;SAVE COPY OF MAIN CODE 'FLAG' VARIABLE
040714  012737  000001  037126          MOV     #1,TXREADY      ;INIT TRANSMIITER DONE FLAG
040722  005037  037120                  CLR     RXPRC           ;INIT RX PROCOTOL DONE
040726  005037  037122                  CLR     TXPRC           ;INIT TX PROCOTOL DONE
040732  005037  037150                  CLR     TIMEOUT         ;INIT PRINT TIMER
040736  032737  000100  006576          BIT     #PRORUN,PARAM   ;PROTOCOL RUNNING ?
040744  001067                          BNE     7$              ;YES,BRANCH

                        ;; PROTOCOL NOT RUNNING -- SO FIRE UP THE LINK
040746                          SETVEC  INVEC,#PRRXI,#PRI05     ;LOAD RX PROTOCOL INTERRUPT ROUTINE
040746  012746  000240                                                  MOV     #PRI05,-(SP)
040752  012746  044510                                                  MOV     #PRRXI,-(SP)
040756  013746  011462                                                  MOV     INVEC,-(SP)
040762  012746  000003                                                  MOV     #3,-(SP)
040766  104437                                                          TRAP    C$SVEC
040770  062706  000010                                                  ADD     #10,SP
040774  005037  037142                  CLR     HDXMTP          ;INIT HALF DUPLEX/MULTI-POINT FLAG
041000  005737  006574                  TST     FHDPLX          ;HALF DUPLEX ?
041004  001403                          BEQ     2$              ;YES,BRANCH
041006  005737  037132                  TST     MPPTP           ;MULTI POINT ?
041012  001403                          BEQ     3$              ;NO,BRANCH
041014  012737  000001  037142  2$:     MOV     #1,HDXMTP       ;SET HALF DUPLEX/MULTI-POINT

041022  012737  000036  006646  3$:     MOV     #30.,TIMERS     ;30 SECONDS TO START
041030  012737  000001  037134          MOV     #1,SELECT       ;INIT SELECT
041036  005037  037146                  CLR     TURNON          ;INIT YET ANOTHER FLAG
041042  005037  037130                  CLR     PRUN            ;INIT ANOTHER FLAG
041046  005037  037124                  CLR     ASTRT           ;INIT 'STACK SENT' FLAG
041052  005037  037052                  CLR     PRSTAT          ;INIT STATUS WORD
041056  004737  041734                  JSR     PC,PROINT       ;INIT PROTOCOL COUNTERS AND VARIABLES
041062  005737  037132                  TST     MPPTP           ;MULTI - POINT MODE ?
041066  001005                          BNE     4$              ;YES,BRANCH
041070  052737  004000  037052          BIS     #SSTART,PRSTAT  ;TELL TX ROUTINE TO SEND 'START'
041076  004737  045336                  JSR     PC,TXPROTO      ;GO SEND IT
041102  004737  042104          4$:     JSR     PC,RXPROTO      ;GO WAIT FOR 'STACK' OR 'START'
041106  032737  000100  006576          BIT     #PRORUN,PARAM   ;DID PROTOCOL START ?
041114  001742                          BEQ     3$              ;NO,TRY AGAIN
041116  012737  000001  037130          MOV     #1,PRUN         ;THIS FLAG USED IN RXPROTO ROUTINE

                        ;; IF HALF DUPLEX OR MULTI POINT, WE MUST MANAGE THE LINK DIFFERENTLY
041124  012737  000003  006646  7$:     MOV     #3.,TIMERS      ;SET UP TIMER
041132  005737  037142                  TST     HDXMTP          ;HALF DUPLEX OR MULTI - POINT?
041136  001076                          BNE     PROHDX          ;YES,BRANCH
```

```
 6952
 6953                                        ;; IF FULL DUPLEX AND ACTIVE MODE-- JUMP
 6954  041140  022737  000003  006570              CMP     #ACT,MODTYP     ;ACTIVE MODE?
 6955  041146  001440                              BEQ     200$            ;YES, BRANCH
 6956
 6957                                        ;; PROTOCOL IS RUNNING -- LINK IS HOT SO SEND DATA
 6958
 6959  041150  032737  000010  037116  10$:     BIT     #QTX,IMFLAG     ;TRANSMITTING A MESSAGE ?
 6960  041156  001414                          BEQ     100$            ;NO,BRANCH
 6961  041160  052737  000020  037052  20$:     BIS     #SDATA,PRSTAT   ;SEND DATA FLAG
 6962  041166  004737  045336                  CALL    TXPROTO         ;GO SEND THE MESSAGE
 6963  041172  004737  042104                  CALL    RXPROTO         ;CHECK THE REPLY
 6964  041176  005737  037122                  TST     TXPRC           ;MESSAGE TRANSMITTED & "ACK'ED"?
 6965  041202  001766                          BEQ     20$             ;NO,BRANCH
 6966  041204  005237  037054                  INC     TMESTX          ;BUMP "TOTAL MESSAGES TRANSMITTED" COUNTER
 6967
 6968  041210  005737  037120          100$:   TST     RXPRC           ;RECEIVE PROTOCOL FINISHED ?
 6969  041214  001011                          BNE     110$            ;YES,BRANCH
 6970  041216  032737  000004  037116          BIT     #QRX,IMFLAG     ;RECEIVING A MESSAGE ?
 6971  041224  001002                          BNE     105$            ;YES,BRANCH
 6972  041226  000137  041674                  JMP     PROTEX          ;EXIT
 6973
 6974
 6975  041232  004737  042104          105$:   CALL    RXPROTO         ;GO PROCESS INCOMING MESSAGE
 6976  041236  000764                          BR      100$            ;SEE IF RECEIVE PROTOCOL COMPLETE
 6977  041240  005237  037056          110$:   INC     TMESRX          ;BUMP "TOTAL MESSAGES RECEIVED" COUNTER
 6978  041244  000137  041674                  JMP     PROTEX          ;EXIT
 6979
 6980
 6981                                        ;; ACTIVE MODE (FULL DUPLEX AND POINT TO POINT LINKS)
 6982
 6983  041250  004737  042000          200$:   CALL    RXON            ;TURN ON RECEIVER
 6984  041254  052737  000020  037052  210$:   BIS     #SDATA,PRSTAT   ;SEND DATA FLAG
 6985  041262  004737  045336                  CALL    TXPROTO         ;DO SEND DATA MESSAGE
 6986  041266  004737  042104          215$:   CALL    RXPROTO         ;GO PROCESS INCOMING MESSAGE
 6987  041272  005737  037122                  TST     TXPRC           ;TX PROTOCOL DONE ?
 6988  041276  001766                          BEQ     210$            ;NO,BRANCH
 6989  041300  005737  037120                  TST     RXPRC           ;RX PROTOCOL DONE ?
 6990  041304  001770                          BEQ     215$            ;NO,BRANCH
 6991  041306  005237  037056                  INC     TMESRX          ;BUMP 'TOTAL MESSAGES RECEIVED'
 6992  041312  005237  037054                  INC     TMESTX          ;BUMP "TOTAL MESSAGE SENT" COUNTER
 6993
 6994                                        ;; TXREADY SET IN TX INTERRUPT ROUTINE
 6995  041316  005737  037126          220$:   TST     TXREADY         ;MESSAGE SENT ?
 6996  041322  001775                          BEQ     220$            ;NO,BRANCH
 6997  041324  004737  042000                  CALL    RXON            ;TURN ON RECEIVER
 6998  041330  000137  041674                  JMP     PROTEX          ;EXIT
 6999
 7000
 7001                                        ;;; THIS ROUTINE(PROHDX) IS USE IN HALF-DUPLEX PT-PT & MTP
 7002
 7003  041334                          PROHDX:
 7004  041334  005737  006574          10$:    TST     FHDPLX          ;FULL DUPLEX ?
 7005  041340  001072                          BNE     PROFDX          ;YES,BRANCH
 7006  041342  032737  000010  037116          BIT     #QTX,IMFLAG     ;TRANSMITTING ?
 7007  041350  001424                          BEQ     100$            ;NO,BRANCH
```

```
 7008  041352  005737  037134          20$:    TST     SELECT          ;DO WE HAVE THE SELECT BIT ?
 7009  041356  001005                          BNE     30$             ;YES,BRANCH
 7010  041360  004737  042000                  CALL    RXON            ;TURN ON RX
 7011  041364  004737  042110          25$:    CALL    RXWAIT          ;TURN ON RX AND WAIT FOR SELECT BIT
 7012  041370  000770                          BR      20$             ;DID WE GET THE SELECT BIT ?
 7013  041372  052737  000020  037052  30$:    BIS     #SDATA,PRSTAT   ;SEND DATA FLAG
 7014  041400  004737  045336                  CALL    TXPROTO         ;GO SENT IT
 7015  041404  004737  042104                  CALL    RXPROTO         ;CHECK REPLY
 7016  041410  005737  037122                  TST     TXPRC           ;TX PROTOCOL DONE ?
 7017  041414  001756                          BEQ     20$             ;NO,BRANCH
 7018  041416  005237  037054                  INC     TMESTX          ;BUMP TOTAL MESSAGES SENT
 7019  041422  012737  000001  037122  100$:   MOV     #1,TXPRC        ;SET TX PROTOCOL DONE
 7020  041430  005737  037120          103$:   TST     RXPRC           ;RX PROTOCOL DONE ?
 7021  041434  001026                          BNE     150$            ;YES,BRANCH
 7022  041436  032737  000004  037116          BIT     #QRX,IMFLAG     ;RECEIVING ?
 7023  041444  001002                          BNE     110$            ;YES,BRANCH
 7024  041446  000137  041674                  JMP     PROTEX          ;EXIT
 7025                          ;;WAS THE BALL TOSSED BACK IN OUR COURT ?
 7026  041452  005737  037134          110$:   TST     SELECT          ;HAVE WE RECEIVED THE SELECT BIT YET?
 7027  041456  001005                          BNE     130$            ;YES,BRANCH
 7028  041460  004737  042000                  CALL    RXON            ;TURN ON RECEIVER
 7029  041464  004737  042110          115$:   CALL    RXWAIT          ;PROCESS DATA
 7030  041470  000757                          BR      103$            ;TRY AGAIN
 7031  041472  052737  000004  037052  130$:   BIS     #SACK,PRSTAT    ;SEND ACK TO TURN THE LINE AROUND
 7032  041500  004737  045336                  CALL    TXPROTO         ;SEND IT
 7033  041504  004737  042104                  CALL    RXPROTO         ;GO RECEIVE THE PENDING MESSAGE
 7034  041510  000747                          BR      103$            ;BRANCH
 7035  041512  005237  037056          150$:   INC     TMESRX          ;BUMP'RECIEVED MESSAGE COUNTER'
 7036  041516  004737  042000                  CALL    RXON            ;TURN ON RX
 7037  041522  000137  041674                  JMP     PROTEX          ;EXIT
 7038
 7039                          ;;THIS ROUTINE(PROFDX:) USED WITH FULL DUPLEX-MULTI POINT LINKS
 7040
 7041  041526  032737  000010  037116  PROFDX: BIT     #QTX,IMFLAG     ;TRANSMITTING ?
 7042  041534  001003                          BNE     10$             ;YES,BRANCH
 7043  041536  012737  000001  037122          MOV     #1,TXPRC        ;SET TRANSMIT PROTOCOL COMPLETE
 7044  041544  005737  037120          10$:    TST     RXPRC           ;WAS THE 1ST MESSAGE RX'ED DURING STARTUP?
 7045  041550  001015                          BNE     30$             ;YES,BRANCH
 7046  041552  032737  000004  037116          BIT     #QRX,IMFLAG     ;RECEIVING ?
 7047  041560  001004                          BNE     20$             ;YES,BRANCH
 7048  041562  012737  000001  037120          MOV     #1,RXPRC        ;SET RECEIVE PROTOCOL COMPLETE
 7049  041570  000410                          BR      100$            ;BRANCH
 7050  041572  004737  042104          20$:    CALL    RXPROTO         ;PROCESS INCOMING MESSAGE
 7051  041576  005737  037120                  TST     RXPRC           ;DONE ?
 7052  041602  001773                          BEQ     20$             ;NO,BRANCH
 7053  041604  005237  037056          30$:    INC     TMESRX          ;BUMP RX MESSAGE COUNT
 7054  041610  000400                          BR      100$            ;BRANCH
 7055
 7056  041612  005737  037122          100$:   TST     TXPRC           ;ANYTHING TO SEND ?
 7057  041616  001024                          BNE     135$            ;NO,BRANCH
 7058
 7059  041620  005737  037134          120$:   TST     SELECT          ;DO WE HAVE PERMISSION TO SEND ?
 7060  041624  001005                          BNE     130$            ;YES,BRANCH
 7061  041626  004737  042000                  CALL    RXON            ;TURN ON TX
 7062  041632  004737  042110          125$:   CALL    RXWAIT          ;WAIT ON SELECT BIT
 7063  041636  000770                          BR      120$            ;TRY AGAIN
```

```
 7064  041640  052737  000020  037052  130$:   BIS     #SDATA,PRSTAT   ;SEND DATA FLAG
 7065  041646  004737  045336                  CALL    TXPROTO         ;SEND DATA
 7066  041652  004737  042104                  CALL    RXPROTO         ;CHECK REPLY
 7067  041656  005737  037122                  TST     TXPRC           ;TRANSMIT PROTOCOL DONE ?
 7068  041662  001756                          BEQ     120$            ;NO,TRY AGAIN
 7069  041664  005237  037054                  INC     TMESTX          ;BUMP 'TOTAL MESSAGE SENT' COUNTER
 7070  041670  004737  042000          135$:   CALL    RXON            ;TURN ON RX SO THAT WE WON'T MISS ANY MESSAGES
 7071                                                                  ;; WHILE WE AWAY.
 7072  041674  022737  177777  037056  PROTEX: CMP     #-1,TMESRX      ;;MAX?
 7073  041702  001002                          BNE     10$             ;NO,BRANCH
 7074  041704  005337  037056                  DEC     TMESRX          ;LATCH AT 177776
 7075  041710  022737  177777  037054  10$:    CMP     #-1,TMESTX      ;MAX?
 7076  041716  001002                          BNE     15$             ;NO,BRANCH
 7077  041720  005337  037054                  DEC     TMESTX          ;LATCH AT 177776
 7078  041724  013737  037116  006602  15$:    MOV     IMFLAG,FLAG     ;RESTORE 'FLAG' TO ITS ORIGINAL VALUE
 7079  041732  000207                          RTS     PC              ;EXIT
 7080
 7081
```

```
7082
7083                                    ;*****************************************************************************
7084                                    ;
7085                                    ;       PROTOCOL INIT ROUTINE:
7086                                    ;
7087                                    ;               THIS ROUTINE WILL INITIALIZE THE ERROR COUNTERS AND  MESSAGE
7088                                    ;               COUNTERS AS NEEDED FOR PROPER DDCMP PROTOCOL OPERATION.
7089                                    ;               DURING NORMAL OPERATION THIS CODE WILL BE CALLED ONCE FROM
7090                                    ;               PROTOCOL STARTUP ROUTINE.
7091                                    ;
7092                                    ;*****************************************************************************
7093
7094
7095  041734  010146                   PROINT: MOV     R1,-(SP)        ;SAVE R1
7096  041736  113737  037065  006534            MOVB    TRIBN,TEMP      ;SAVE TRIB NUMBER
7097  041744  012701  037052                    MOV     #PRSTAT,R1      ;FIRST LOCATION TO CLEAR
7098  041750  005021                   10$:     CLR     (R1)+           ;CLEAR AND INCREMENT
7099  041752  020127  037114                    CMP     R1,#PROEND      ;LAST LOCATION TO CLEAR
7100  041756  001374                            BNE     10$             ;NO,BRANCH
7101  041760  113737  006534  037065  20$:     MOVB    TEMP,TRIBN      ;RESTORE TRIB #
7102  041766  112737  000001  037062            MOVB    #1,T            ;FIRST MESSAGE # TO BE TRANMITTED
7103  041774  012601                            MOV     (SP)+,R1        ;RESTORE R1
7104  041776  000207                            RETURN                  ;EXIT
7105
7106                                    ;*****************************************************************************
7107                                    ; TURN ON RECEIVER ROUTINE:
7108                                    ;
7109                                    ;       DESCRIPTION: THIS ROUTINE SIMPLY ENABLES THE RECEIVER AND
7110                                    ;                    INITIALIZES VARIABLES THEN RETURNS.
7111                                    ;
7:12                                    ;*****************************************************************************
7113
7114  042000  005737  037146           RXON:   TST     TURNON          ;RX ALREADY ON ?
7115  042004  001036                            BNE     RXONEX          ;YES,BRANCH
7116  042006  005037  037052                    CLR     PRSTAT          ;INIT STATUS WORD
7117  042012  005037  002656                    CLR     RHDMSG          ;INIT 1ST WORD OF RX BUFFER
7118  042016  042737  000444  037140            BIC     #QRX!#BCC!#RXM,PRFLAG   ;FLAGS USED IN RX INTERRUPT ROUTINE
7119  042024  052737  001000  037052            BIS     #MYDATA,PRSTAT  ;ASSUME MESSAGE FOR ME
7120  042032  012737  002657  011510            MOV     #RHDMID,RMSGPT  ;BUFFER ADDRESS FOR HEADER PART ON MESSAGE
7121  042040  013737  002654  011512            MOV     HDMC,RMSGCC     ;INIT CHARACTER COUNT = 6
7122  042046  032737  000010  006576            BIT     #MOCHK,PARAM    ;MODEM CHANGES WANTED ?
7123  042054  001004                            BNE     20$             ;YES,BRANCH
7124  042056  052777  000520  147364            BIS     #RINTEN!RXENA!STRIP,@RXCSR      ;TURN ON RX
7125  042064  000403                            BR      25$             ;BRANCH
7126  042066  052777  000560  147354  20$:     BIS     #RINTEN!RXENA!DSITEN!STRIP,@RXCSR       ;TURN ON RX
7127  042074  012737  000001  037146  25$:     MOV     #1,TURNON       ;RX IS ON FLAG
7128  042102  000207                   RXONEX: RETURN
7129
```

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


042104                  RXPROTO:
042104  004737  042000          CALL    RXON            ;TURN ON RECEIVER
                        ;; WAIT FOR FIRST CHARACTER TO APPEAR IN RX BUFFER
042110                  RXWAIT:
042110                  20$:    BREAK                   ;CHECK FOR ^C
042110  104422                                                          TRAP    C$BRK
042112  105737  002657          TSTB    RHDMID          ;FIRST CHARACTER READ ?
042116  001007                  BNE     30$             ;YES, BRANCH
042120  005737  006646          TST     TIMERS          ;60 SECONDS ELAPSED ?
042124  001371                  BNE     20$             ;NO,BRANCH
042126  004737  043732          JSR     PC,ERRPRC       ;CALL ERROR PROCESSOR
042132  000137  043730          JMP     RXPREX          ;EXIT

                        ;; IF A CONTROL MESSAGE THEN TELL RX INTR. TO PROCESS HEADER ONLY
042136  123727  002657  000005  30$:    CMPB    RHDMID,#ENQ     ;CONTROL MESSAGE ?
042144  001003                  BNE     40$             ;NO,BRANCH
042146  052737  000040  037140          BIS     #RXM,PRFLAG     ;PROCESS HEADER ONLY

                        ;; WAIT FOR CRC TO BE CHECKED
042154  032737  000003  037052  40$:    BIT     #BCCOK!BCCBAD,PRSTAT    ;CRC CHECKED ?
042162  001021                  BNE     50$                     ;YES,BRANCH
042164  005737  006646          TST     TIMERS          ;60 SECONDS ELAPSED ?
042170  001004                  BNE     45$             ;NO,BRANCH
042172  004737  043732          JSR     PC,ERRPRC       ;GO PROCESS ERROR
042176  000137  042104          JMP     RXPROTO         ;TRY AGAIN
```

```
7174
7175                                    ;; CHECK THAT RX WAS SERVICED QUICK ENOUGH(DETERMINED BY RX INTER. ROUTINE)
7176  042202  032737  000010  037052  45$:    BIT     #SNAK,PRSTAT    ;RX OVERRUN ?
7177  042210  001761                          BEQ     40$             ;NO,BRANCH
7178  042212  004737  043732                  JSR     PC,ERRPRC       ;GO PROCESS ERROR
7179  042216  004737  045336                  JSR     PC,TXPROTO      ;GO SEND NAK
7180  042222  000137  042104                  JMP     RXPROTO         ;TRY AGAIN
7181
7182                                    ;; IF HEADER CRC ERROR THEN LOG IT AND SEND NAK
7183  042226  032737  000002  037052  50$:    BIT     #BCCBAD,PRSTAT  ;CRC ERROR ?
7184  042234  001430                          BEQ     60$             ;NO,BRANCH
7185  042236  052737  000010  037052          BIS     #SNAK,PRSTAT    ;SET SNAK (SEND NAK)
7186  042244  152737  000001  037077          BISB    #HEADBCC,INMASK ;SET THE MASK
7187  042252  112737  000001  037070          MOVB    #HEADBCC,REANAK ;NAK REASON = 1
7188  042260  105237  037076                  INCB    DERIN           ;LOG DATA ERROR INBOUND
7189  042264  001003                          BNE     55$             ;BRANCH IF NOT OVERFLOW
7190  042266  112737  000377  037076          MOVB    #377,DERIN      ;LATCH COUNTER AT 255.
7191  042274  004737  043732  55$:    JSR     PC,ERRPRC       ;GO PROCESS ERROR
7192  042300  012737  000001  037134          MOV     #1,SELECT       ;IF HALF/DUPLEX, WE ASSUME S-BIT WAS SET
7193  042306  004737  045336                  JSR     PC,TXPROTO      ;GO SEND NAK
7194  042312  000137  042104                  JMP     RXPROTO         ;TRY AGAIN
7195
7196                                    ;; NOW CHECK THE ADDRESS OF THE MESSAGE- IS IT FOR ME ?
7197  042316  123737  037065  002664  60$:    CMPB    TRIBN,RHDADR    ;MY ADDRESS ?
7198  042324  001422                          BEQ     70$             ;YES, BRANCH
7199
7200                                    ;; ITS NOT FOR ME, BUT COUNT IT OUT TO KEEP RX IN SYNC
7201  042326  042737  001000  037052  62$:    BIC     #MYDATA,PRSTAT  ;MESSAGE NOT FOR ME
7202  042334  032737  000100  037052          BIT     #RXD,PRSTAT     ;RECEIVER DONE ?
7203  042342  001003                          BNE     65$             ;YES,BRANCH
7204  042344  005737  006646                  TST     TIMERS          ;HAVE WE DAWDLED LONG ENOUGH ?
7205  042350  001366                          BNE     62$             ;NO,BRANCH
7206
7207  042352  032737  000001  037052  65$:    BIT     #BCCOK,PRSTAT   ;DATA CRC OK ?
7208  042360  001002                          BNE     67$             ;YES,BRANCH
7209  042362  105237  037067                  INCB    GLOBCC          ;LOG GLOBAL CRC ERROR
7210  042366  000137  042104          67$:    JMP     RXPROTO         ;GO RE-QUE BUFFER
7211
7212
7213                                    ;; IS IT A CONTROL MESSAGE ? IF IT IS PROCESS IT
7214
7215  042372  105037  037076  70$:    CLRB    RXTHER          ;INIT RX THRESHOLD ERROR COUNTER
7216  042376  122737  000005  002657          CMPB    #ENQ,RHDMID     ;CONTROL MESSAGE ?
7217  042404  001402                          BEQ     75$             ;YES,BRANCH
7218  042406  000137  043256                  JMP     200$            ;GO PROCESS DATA MESSAGE
7219
7220                                    ;; IS IT A NAK ?
7221  042412  122737  000002  002660  75$:    CMPB    #NAK,RHDTYP     ;NAK?
7222  042420  001022                          BNE     90$             ;NO,BRANCH
7223  042422  032737  000100  006576          BIT     #PRORUN,PARAM   ;PROTOCOL RUNNING ?
7224  042430  001002                          BNE     80$             ;YES,BRANCH
7225  042432  000137  042104                  JMP     RXPROTO         ;IGNORE THIS MESSAGE
7226  042436  052737  000400  037052  80$:    BIS     #NAKRX,PRSTAT   ;FLAG NAK RECEIVED
7227  042444  004737  043732                  JSR     PC,ERRPRC       ;GO LOG NAK REASON
7228  042450  052737  000020  037052          BIS     #SDATA,PRSTAT   ;SEND DATA
7229  042456  004737  045336                  JSR     PC,TXPROTO      ;GO RE-TRANSMIT PREVIOUS MESSAGE
```

```
  7230  042462  000137  042104                JMP     RXPROTO         ;GO RE-QUE RX
```

```
7231
7232                                        ;; IS IT A ACK ?
7233  042466  122737  000001  002660  90$:  CMPB  #ACK,RHDTYP     ;ACK ?
7234  042474  001057                        BNE   100$            ;NO,BRANCH
7235  042476  032737  000100  006576        BIT   #PRORUN,PARAM   ;PROTOCOL RUNNING ?
7236  042504  001004                        BNE   93$             ;YES,BRANCH
7237  042506  052737  000100  006576        BIS   #PRORUN,PARAM   ;TELL THE WORLD THAT LINK HAS STARTED
7238  042514  000445                        BR    97$             ;EXIT
7239  042516  123737  037062  002662  93$:  CMPB  T,RHDREP        ;CORRECT MESSAGE # ACKNOWLEDGED ?
7240  042524  001405                        BEQ   95$             ;YES,BRANCH
7241  042526  005737  037142                TST   HDXMTP          ;HALF DUPLEX/MULTI -POINT ?
7242  042532  001036                        BNE   97$             ;YES,BRANCH
7243  042534  000137  042104                JMP   RXPROTO         ;TRY AGAIN
7244  042540  105037  037073        95$:    CLRB  TXTHER          ;INIT. TX THRESHOLD COUNTER
7245  042544  113737  037062  037060        MOVB  T,N             ;HIGHEST SEQUENTIAL MESSAGE # SENT
7246  042552  113737  037062  037063        MOVB  T,X             ;HIGHEST MESSAGE # SENT
7247  042560  113737  002662  037061        MOVB  RHDREP,A        ;HIGHEST MESSAGE # ACKNOWLEDGED TO THIS STATION
7248  042566  105237  037062                INCB  T               ;# OF NEXT DATA MESSAGE TO BE TRANSMITTED
7249  042572  012737  177777  037122        MOV   #-1,TXPRC       ;TRANSMIT PROTOCOL COMPLETE
7250  042600  022737  000003  006570        CMP   #ACT,MODTYP     ;ACTIVE MODE ?
7251  042606  001010                        BNE   97$             ;NO,BRANCH
7252  042610  005737  037120                TST   RXPRC           ;RX PROTOCOL COMPLETE?
7253  042614  001005                        BNE   97$             ;YES,BRANCH
7254  042616  005737  006574                TST   FHDPLX          ;HALF DUPLEX?
7255  042622  001402                        BEQ   97$             ;YES,BRANCH
7256  042624  000137  042104                JMP   RXPROTO         ;GO PROCESS INCOMING MESSAGE
7257
7258  042630  000137  043730        97$:    JMP   RXPREX          ;EXIT
7259
7260                                        ;; IS IT A REP ?
7261  042634  122737  000003  002660  100$:  CMPB  #REP,RHDTYP     ;REP ?
7262  042642  001054                        BNE   150$            ;NO,BRANCH
7263
7264                                        ;; NUM = R ?
7265  042644  032737  000100  006576        BIT   #PRORUN,PARAM   ;PROTOCOL RUNNING ?
7266  042652  001002                        BNE   110$            ;YES,BRANCH
7267  042654  000137  042104                JMP   RXPROTO         ;IGNORE MESSAGE- TRY AGAIN
7268  042660  123737  002663  037064  110$:  CMPB  RHDNUM,R        ;HAVE WE RECEIVED THIS MESSAGE ?
7269  042666  001015                        BNE   120$            ;NO, BRANCH
7270  042670  052737  000004  037052        BIS   #SACK,PRSTAT    ;SET SEND ACK
7271  042676  105237  037066                INCB  REMTMO          ;BUMP REMOTE TIME OUT COUNTER
7272  042702  001003                        BNE   115$            ;BRANCH IF NOT OVERFLOW
7273  042704  112737  000377  037066        MOVB  #377,REMTMO     ;LATCH COUNTER AT 255.
7274  042712  004737  045336        115$:   JSR   PC,TXPROTO      ;GO SEND ACK
7275  042716  000137  042104                JMP   RXPROTO         ;TRY AGAIN
7276
7277                                        ;; NUM <> R
7278  042722  052737  000010  037052  120$:  BIS   #SNAK,PRSTAT    ;SET SEND NAK
7279  042730  112737  000003  037070        MOVB  #REPSENT,REANAK ;SET REASON FOR NAK
7280  042736  105237  037076                INCB  DERIN           ;BUMP DATA ERROR INBOUND
7281  042742  001003                        BNE   125$            ;BRANCH IF NOT OVERFLOW
7282  042744  112737  000377  037076        MOVB  #377,DERIN      ;LATCH AT 255.
7283  042752  152737  000004  037077  125$:  BISB  #REPMSK,INMASK  ;ERROR REASON IS REMOTE TIME OUT
7284  042760  004737  043732                JSR   PC,ERRPRC       ;PROCESS NAK
7285  042764  004737  045336                JSR   PC,TXPROTO      ;GO SEND NAK
7286  042770  000137  042104                JMP   RXPROTO         ;TRY AGAIN
```

```
7287
7288                                        ;; IS IT A START ?
7289  042774  122737  000006  002660  150$:  CMPB    #STRT,RHDTYP    ;START ?
7290  043002  001071                        BNE     170$            ;NO,BRANCH
7291  043004  032737  000100  006576        BIT     #PRORUN,PARAM   ;PROTOCOL RUNNING ?
7292  043012  001007                        BNE     160$            ;YES,BRANCH
7293  043014  052737  002000  037052        BIS     #SSTACK,PRSTAT  ;SEND START ACKNOWLEDGE
7294  043022  004737  045336                JSR     PC,TXPROTO      ;GO SEND STACK
7295  043026  000137  042104                JMP     RXPROTO         ;GO TO RX ROUTINE AND EXPECT ACK OR DATA
7296                                                                ;; IF DMV OR DMP.
7297
7298  043032  052737  000200  006576  160$:  BIS     #ABORT,PARAM    ;TELL MAIN CODE TO ABORT!!
7299  043040  012737  177777  037120        MOV     #-1,RXPRC       ;RECEIVE PROTCOL DONE
7300  043046  012737  177777  037122        MOV     #-1,TXPRC       ;TRANSMIT PROTOCOL DONE
7301  043054                                PRINTF  #165$           ;FATAL ERROR
7302  043054  012746  043100                                                    MOV     #165$,-(SP)
7303  043060  012746  000001                                                    MOV     #1,-(SP)
7304  043064  010600                                                            MOV     SP,R0
7305  043066  104417                                                            TRAP    C$PNTF
7306  043070  062706  000004                                                    ADD     #4,SP
7307  043074  000137  043730                JMP     RXPREX          ;EXIT
7308
7309                                      .NLIST BEX
      043100  047045  040445  052123  165$:  .ASCIZ  /%N%ASTART RECEIVED WITH PROTOCOL RUNNING--ABORTING!!/
              043166                      .EVEN
                                        .LIST BEX
7310
7311                                        ;; IS IT A STACK ? IF SO SEND AN 'ACK'
7312
7313  043166  122737  000007  002660  170$:  CMPB    #STACK,RHDTYP   ;STACK ?
7314  043174  001012                        BNE     180$            ;NO, BRANCH
7315  043176  052737  000004  037052        BIS     #SACK,PRSTAT    ;TELL TX ROUTINE TO SEND ACK
7316  043204  004737  045336                JSR     PC,TXPROTO      ;SEND ACK
7317  043210  052737  000100  006576        BIS     #PRORUN,PARAM   ;SET 'PROTOCOL RUNNING' FLAG
7318  043216  000137  043730                JMP     RXPREX          ;EXIT
7319
7320
7321                                        ;; IF WE GOT TO HERE, WE HAVE A STRANGE PROBLEM !
7322  043222  052737  000010  037052  180$:  BIS     #SNAK,PRSTAT    ;SET SEND NAK FLAG
7323  043230  105237  037106                INCB    LOSTER          ;LOCAL STATION ERROR
7324  043234  152737  000021  037107        BISB    #FORMERR,LSMASK ;FORMAT ERROR
7325  043242  004737  043732                JSR     PC,ERRPRC       ;PROCESS ERROR
7326  043246  004737  045336                JSR     PC,TXPROTO      ;SEND NAK
7327  043252  000137  042104                JMP     RXPROTO         ;TRY AGAIN
7328                                      ;; END OF CONTROL MESSAGE PROCESSOR
7329
7330
7331
```

```
7332
7333                                    ;;;HERE WE BEGIN PROCESSING DATA PART OF MESSAGE
7334
7335
7336  043256  005737  037120            200$:   TST     RXPRC           ;ALREADY PROCESSED A MESSAGE?
7337  043262  001432                            BEQ     215$            ;NO,BRANCH
7338  043264  042737  001000  037052            BIC     #MYDATA,PRSTAT  ;TELL RX INTERRUPT NOT TO STORE THIS
7339  043272  005737  037122                    TST     TXPRC           ;TX PROTOCOL DONE ?
7340  043276  001037                            BNE     220$            ;YES,BRANCH
7341                                    ;; SEE IF IMPLICIT ACK IMBEDDED IN THIS MESSAGE
7342  043300  123737  037062  002662            CMPB    T,RHDREP        ;RESP=MESSAGE SENT?
7343  043306  001033                            BNE     220$            ;NO,BRANCH
7344  043310  113737  037062  037060            MOVB    T,N             ;HIGHEST # SENT
7345  043316  113737  037062  037063            MOVB    T,X             ;
7346  043324  113737  037062  037061            MOVB    T,A             ;HIGHEST MESSAGE ACK'ED
7347  043332  105237  037062                    INCB    T               ;NEXT MESSAGE TO SEND
7348  043336  012737  177777  037122            MOV     #-1,TXPRC       ;TX PROTOCOL DONE
7349  043344  000137  043730                    JMP     RXPREX          ;EXIT
7350                                    ;;CHECK SEQUENCE OF MESSAGE
7351  043350  105237  037064            215$:   INCB    R               ;EXPECTED #?
7352  043354  123737  037064  002663            CMPB    R,RHDNUM        ;CORRECT MESSAGE #?
7353  043362  001423                            BEQ     300$            ;YES,PROCESS IT
7354  043364  105337  037064                    DECB    R               ;SUBTRACT 1
7355  043370  042737  001000  037052            BIC     #MYDATA,PRSTAT  ;JUST COUNT OUT MESSAGE-DON'T PUT IN BUFFER
7356  043376  032737  000100  037052  220$:     BIT     #RXD,PRSTAT     ;WAIT FOR DONE
7357  043404  001003                            BNE     250$            ;BRANCH
7358  043406  005737  006646                    TST     TIMERS          ;TIME OUT?
7359  043412  001371                            BNE     220$            ;NO,BRANCH
7360
7361                                    ;;SEND AN "ACK"
7362  043414  052737  000004  037052  250$:     BIS     #SACK,PRSTAT    ;SEND ACK
7363  043422  004737  045336                    CALL    TXPROTO         ;GO SEND IT
7364  043426  000137  042104                    JMP     RXPROTO         ;TRY AGAIN
7365
7366                                    ;; IS DATA PART OF MESSAGE COMPLETE ?
7367  043432  032737  000100  037052  300$:     BIT     #RXD,PRSTAT     ;MESSAGE COMPLETE ?
7368  043440  001021                            BNE     330$            ;YES,BRANCH
7369
7370                                    ;; IS THE LINE DEAD ?
7371  043442  005737  006646                    TST     TIMERS          ;TIMED-OUT ?
7372  043446  001004                            BNE     305$            ;NO,BRANCH
7373  043450  004737  043732                    JSR     PC,ERRPRC       ;GO PROCESS TIMER ERROR
7374  043454  000137  042104                    JMP     RXPROTO         ;TRY AGAIN
7375
7376                                    ;; CHECK FOR RECEIVER OVERRUN OR BUFFER PROBLEM
7377  043460  032737  000010  037052  305$:     BIT     #SNAK,PRSTAT    ;DID RX INTERRUPT SET THIS ?
7378  043466  001761                            BEQ     300$            ;NO,BRANCH
7379
7380                                    ;;RX ERROR SEND A NAK AND TRY AGAIN
7381  043470  004737  043732                    JSR     PC,ERRPRC       ;GO PROCESS ERROR
7382  043474  004737  045336                    JSR     PC,TXPROTO      ;SEND NAK
7383  043500  000137  042104                    JMP     RXPROTO         ;TRY AGAIN
7384
7385                                    ;;CHECK FOR DATA CRC ERROR
7386  043504  032737  000001  037052  330$:     BIT     #BCCOK,PRSTAT   ;DATA CRC GOOD ?
7387  043512  001022                            BNE     400$            ;YES,BRANCH
```

```
7388
7389                                        ;; LOG CRC ERROR AND SEND A NAK
7390  043514  052737  000010  037052          BIS     #SNAK,PRSTAT    ;SET SEND NAK FLAG
7391  043522  105237  037076                  INCB    DERIN           ;BUMP DATA ERROR INBOUND COUNTER
7392  043526  001003                          BNE     340$            ;BRANCH IF NOT OVERFLOW
7393  043530  112737  000377  037076          MOVB    #377,DERIN      ;LATCH AT 255.
7394  043536  152737  000002  037077  340$:   BISB    #DATABCC,INMASK ;SET DATA CRC BIT
7395  043544  004737  043732                  JSR     PC,ERRPRC       ;GO PROCESS ERROR
7396  043550  004737  045336                  JSR     PC,TXPROTO      ;GO SEND NAK
7397  043554  000137  042104                  JMP     RXPROTO         ;TRY AGAIN
7398
7399
7400                                        ;; WE HAVE A GOOD MESSAGE !!! SO ACKNOWLEDGE IT
7401  043560  032737  000100  006576  400$:   BIT     #PRORUN,PARAM   ;PROTOCOL RUNNING?
7402  043566  001007                          BNE     420$            ;YES,BRANCH
7403  043570  005737  037124                  TST     ASTRT           ;DID WE SEND A STACK?
7404  043574  001001                          BNE     415$            ;YES,BRANCH
7405  043576  000454                          BR      RXPREX          ;EXIT
7406
7407                                        ;; NOTE: DMV/DPM WILL SEND 'START - STACK - DATA' FOR STARTUP SEQUENCE
7408  043600  052737  000100  006576  415$:   BIS     #PRORUN,PARAM   ;SET PROTOCOL RUNNING
7409
7410                                        ;; CHECK FOR AN IMPLICIT 'ACK'
7411  043606  123737  037062  002662  420$:   CMPB    T,RHDREP        ;RESP = MESSAGE SENT ?
7412  043614  001016                          BNE     450$            ;NO,BRANCH
7413  043616  113737  037062  037060          MOVB    T,N             ;HIGHEST SEQ MESSAGE # SENT
7414  043624  113737  037062  037063          MOVB    T,X             ;HIGHEST MESSAGE SENT
7415  043632  113737  037062  037061          MOVB    T,A             ;HIGHEST MESSAGE 'ACK'ED'
7416  043640  105237  037062                  INCB    T               ;NEXT MESSAGE # TO TRANSMIT
7417  043644  012737  177777  037122          MOV     #-1,TXPRC       ;SET TRANSMIT PROTOCOL COMPLETE
7418  043652  052737  000004  037052  450$:   BIS     #SACK,PRSTAT    ;SET SEND ACK FLAG
7419  043660  004737  045336                  JSR     PC,TXPROTO      ;SEND ACK
7420  043664  012737  177777  037120          MOV     #-1,RXPRC       ;RECEIVE MESSAGE PROTOCOL FINISHED
7421  043672  005737  037130                  TST     PRUN            ;PROTOCOL RUNNING ?
7422  043676  001414                          BEQ     RXPREX          ;NO,BRANCH
7423  043700  005737  037142                  TST     HDXMTP          ;FULL DUPLEX PT-PT?
7424  043704  001011                          BNE     RXPREX          ;NO,BRANCH
7425  043706  022737  000003  006570          CMP     #ACT,MODTYP     ;ACTIVE MODE ?
7426  043714  001005                          BNE     RXPREX          ;NO,BRANCH
7427  043716  005737  037122                  TST     TXPRC           ;TRANSMIT PROTOCOL COMPLETE ?
7428  043722  001002                          BNE     RXPREX          ;YES,BRANCH
7429  043724  000137  042104                  JMP     RXPROTO         ;GO PROCESS MESSAGE
7430
7431  043730  000207          RXPREX: RETURN                          ;DONE !!
7432
```

```
                        ;*****************************************************************************
                        ;       ERROR PROCESSING ROUTINE (ERRPRC):
                        ;
                        ;       DESCRIPTION: THIS ROUTINE IS USED TO PROCESS INBOUND AND
                        ;                    OUTBOUND ERRORS. ALSO THE 60 SECOND 'WATCHDOG'
                        ;                    TIMER IS CHECKED.
                        ;
                        ;                  THE MAJORITY OF THE CODE IS USED IN PROCESSING
                        ;                    OUTBOUND ERRORS (NAKS RECEIVED). THE NAK REASON
                        ;                    TYPE IS DETERMINED AND THE APPROIATE ERROR
                        ;                    COUNTER IN INCREMENTED. IF THE TRANSMIT THRESHOLD
                        ;                    COUNTER (TXTHER) REACHES 7, IT IS CLEARED
                        ;                    AND THE CUMULATIVE RECEIVE/TRANSMIT THRESHOLD
                        ;                    ERROR (RXTXTE) COUNTER IS BUMPED.
                        ;*****************************************************************************


                        ;;CHECK THE WATCHDOG TIMER

043732  005737  006646  ERRPRC: TST     TIMERS          ;60 SECONDS ELAPSED
043736  001075                  BNE     10$             ;NO,BRANCH
043740  032737  000100  006576        BIT     #PRORUN,PARAM   ;PROTOCOL RUNNING ?
043746  001034                  BNE     7$              ;YES,BRANCH

                        ;; INFORM USER OF 'START - STACK' TIMEOUT
043750  005037  006542          CLR     TEMP3           ;INIT IT
043754  005037  006544          CLR     TEMP4           ;INIT IT
043760  012737  017313  006540  MOV     #DVEM5,TEMP2    ;"TIME OUT IN START-STACK SEQUENCE"
043766  113737  002660  006542  MOVB    RHDMCC,TEMP3    ;RECEIVED DATA
043774  113737  002646  006544  MOVB    HDMCC,TEMP4     ;TRANSMITTED DATA
044002  004737  020162          JSR     PC,LGDVE        ;LOG TIME OUT IN EVENT LOG
044006  005237  006506          INC     ERRCNT          ;BUMP ERROR COUNT
044012                          ERRSOFT 10.,DVEM5,ERR13 ;PRINT ERROR
044012  104457                                                  TRAP    C$ERSOFT
044014  000012                                                  .WORD   10
044016  017313                                                  .WORD   DVEM5
044020  017636                                                  .WORD   ERR13
044022  005237  006502          INC     OPVAR           ;BUMP ERROR COUNTER
044026  012737  000036  006646  MOV     #30.,TIMERS     ;RE-INIT TIMER
044034  000137  044506          JMP     ERREXT          ;EXIT

                        ;; INFORM USER OF "DATA MESSAGE" TIMEOUT
044040  005237  037150  7$:     INC     TIMEOUT         ;BUMP COUNTER
044044  022737  000024  037150  CMP     #20.,TIMEOUT    ;60 SECONDS ?
044052  001023                  BNE     9$              ;NO,BRANCH
044054  012737  017110  006540  MOV     #DVEM2,TEMP2    ;"TIME OUT WAITING FOR RX OR TX TO COMPLETE"
044062  017737  145362  006542  MOV     @RXCSR,TEMP3    ;RECEIVER ADDRESS
044070  017737  145362  006544  MOV     @TXCSR,TEMP4    ;TRANSMIT ADDRESS
044076  004737  020162          JSR     PC,LGDVE        ;LOG ERROR
044102  005237  006506          INC     ERRCNT          ;BUMP ERROR COUNT
044106                          ERRSOFT 7,DVEM2,ERR13   ;PRINT ERROR
044106  104457                                                  TRAP    C$ERSOFT
044110  000007                                                  .WORD   7
044112  017110                                                  .WORD   DVEM2
```

```
044114  017636                                                                  .WORD   ERR13
044116  005037  037150                  CLR     TIMEOUT         ;INIT TIMEOUT

044122  012737  000003  006646  9$:     MOV     #3.,TIMERS      ;SET UP TIMER
044130  000566                          BR      ERREXT          ;EXIT


                                ;;IF A 'NAK' RECEIVED THEN PROCESS IT
044132  032737  000400  037052  10$:    BIT     #NAKRX,PRSTAT   ;NAK RECEIVED?
044140  001542                          BEQ     100$            ;NO,BRANCH

                                ;;IF TRANSMIT THRESHOLD COUNTER = 7 THEN BUMP CUMULATIVE TXRX COUNTER
044142  122737  000007  037073          CMPB    #7,TXTHER       ;THRESHOLD REACHED?
044150  001403                          BEQ     20$             ;YES,BRANCH
044152  105237  037073                  INCB    TXTHER          ;BUMP TRANSMIT THESHOLD
044156  000404                          BR      30$             ;BRANCH
044160  005237  037110          20$:    INC     RXTXTE          ;BUMP TRANSMIT/RECEIVE THRESHOLD COUNTER
044164  105037  037073                  CLRB    TXTHER          ;SET TRANSMIT COUNTER TO ZERO

                                ;;;;DETERMINE THE 'NAK' REASON

                                ;;HEADER CRC ERROR ?
044170  042737  140000  002660  30$:    BIC     #BIT15!BIT14,RHDTYP     ;CLEAR SELECT & QS FLAG
044176  122737  000001  002661          CMPB    #HEADBCC,RHDTYP+1       ;HEADER CRC ERROR?
044204  001012                          BNE     35$                     ;NO,BRANCH
044206  105237  037074                  INCB    DEROUT                  ;LOG ERROR
044212  001003                          BNE     32$                     ;BRANCH IF NOT OVERFLOW
044214  112737  000377  037074          MOVB    #377,DEROUT             ;LATCH AT 255.
044222  152737  000001  037075  32$:    BISB    #HEADBCC,OUTMASK        ;SET MASK
044230  000526                          BR      ERREXT                  ;EXIT

                                ;;DATA CRC ERROR ?
044232  122737  000002  002661  35$:    CMPB    #DATABCC,RHDTYP+1       ;DATA CRC ERROR ?
044240  001012                          BNE     40$                     ;NO,BRANCH
044242  105237  037074                  INCB    DEROUT                  ;LOG ERROR
044246  001003                          BNE     37$                     ;BRANCH IF NOT OVERFLOW
044250  112737  000377  037074          MOVB    #377,DEROUT             ;LATCH AT 255.
044256  152737  000002  037075  37$:    BISB    #DATABCC,OUTMASK        ;SET MASK
044264  000510                          BR      ERREXT                  ;EXIT

                                ;;REMOTE STATION BUFFER NOT AVAILABLE?
044266  122737  000010  002661  40$:    CMPB    #BUFFNA,RHDTYP+1        ;BUFFER NOT AVAILABLE?
044274  001012                          BNE     45$                     ;NO,BRANCH
044276  105237  037102                  INCB    RBUFER                  ;LOG ERROR
044302  001003                          BNE     43$                     ;BRANCH IF NOT OVERFLOW
044304  112737  000377  037102          MOVB    #377,RBUFER             ;LATCH AT 255.
044312  152737  000001  037103  43$:    BISB    #BNAMSK,RBMASK          ;SET MASK
044320  000472                          BR      ERREXT                  ;EXIT

                                ;;REMOTE STATION RECEIVER OVERRUN?
044322  122737  000011  002661  45$:    CMPB    #RXOVRUN,RHDTYP+1       ;RECEIVER OVERRUN?
044330  001012                          BNE     50$                     ;NO, BRANCH
044332  105237  037104                  INCB    RMSTER                  ;LOG ERROR
044336  001003                          BNE     47$                     ;BRANCH IF NO OVERFLOW
044340  112737  000377  037104          MOVB    #377,RMSTER             ;LATCH AT 255.
044346  152737  000001  037105  47$:    BISB    #RXOVMSK,RMMASK         ;SET MASK
```

```
7545  044354  000454                          BR      ERREXT                  ;EXIT
7546
7547                                  ;;REMOTE STATION MESSAGE TOO LONG?
7548  044356  122737  000020  002661  50$:    CMPB    #MESLONG,RHDTYP+1       ;MESSAGE TOO LONG?
7549  044364  001012                          BNE     55$                     ;NO,BRANCH
7550  044366  105237  037102                  INCB    RBUFER                  ;LOG REMOTE STATION BUFFER ERROR
7551  044372  001003                          BNE     52$                     ;BRANCH IF NO OVERFLOW
7552  044374  112737  000377  037102          MOVB    #377,RBUFER             ;LATCH AT 255.
7553  044402  152737  000002  037103  52$:    BISB    #MTLMSK,RBMASK          ;SET MASK
7554  044410  000436                          BR      ERREXT                  ;EXIT
7555
7556                                  ;;REMOTE STATION FORMAT ERROR?
7557  044412  122737  000021  002661  55$:    CMPB    #FORMERR,RHDTYP+1       ;REMOTE STATION FORMAT ERROR?
7558  044420  001012                          BNE     100$                    ;NO,BRANCH
7559  044422  105237  037104                  INCB    RMSTER                  ;LOG ERROR
7560  044426  001003                          BNE     57$                     ;BRANCH IF NO OVERFLOW
7561  044430  112737  000377  037104          MOVB    #377,RMSTER             ;LATCH AT 255.
7562  044436  152737  000002  037105  57$:    BISB    #FMTMSK,RMMASK          ;SET MASK
7563  044444  000420                          BR      ERREXT                  ;EXIT
7564
7565
7566                                  ;; IF SEND NAK (SNAK=1) THEN BUMP RECEIVER THRESHOLD ERROR COUNTER
7567                                  ;;;  IF RXTHER = 7 THE BUMP RXTXTE COUNTER AND CLEAR RXTHER
7568  044446  032737  000010  037052  100$:   BIT     #SNAK,PRSTAT            ;SEND NAK ?
7569  044454  001414                          BEQ     ERREXT                  ;NO, BRANCH
7570
7571  044456  122737  000007  037072          CMPB    #7,RXTHER               ;RECEIVER THRESHOLD = 7?
7572  044464  001403                          BEQ     120$                    ;YES,BRANCH
7573  044466  105237  037072                  INCB    RXTHER                  ;BUMP COUNTER
7574  044472  000405                          BR      ERREXT                  ;BRANCH
7575
7576  044474  005237  037110          120$:   INC     RXTXTE                  ;BUMP CUMULATIVE COUNTER
7577  044500  105037  037072                  CLRB    RXTHER                  ;INIT RECEIVER THRESHOLD COUNTER
7578  044504  000400                          BR      ERREXT                  ;EXIT
7579
7580
7581
7582  044506  000207          ERREXT: RETURN
7583
```

```
                                .SBTTL          RECEIVER PROTOCOL INTERRUPT ROUTINE

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       THIS ROUTINE IS USED ONLY WHEN THE "/PROTOCOL" SWITCH
                                ;       IS SPECIFIED BY THE USER.
                                ;
                                ;       WHEN A RX INT. OCCURS THIS ROUTINE DECIDES IF IT IS A DATA SET
                                ;       CHANGE OR DATA INTERRUPT. IF IT IS A DATA SET CHANGE
                                ;       INTERRUPT IT PUTS THE STATUS IN "CMODS" AND COMPARES
                                ;       THAT STATUS TO THE OLD STATUS IN 'MODS". IF THEY ARE
                                ;       THE SAME THAT MEANS THE INTERRUPT WAS CAUSED BY A GLITCH
                                ;       ON ONE OF THE LINES. IF THEY ARE DIFFERENT THEN A HARD
                                ;       MODEM ERROR HAS OCCURED. IN ANY EVENT THE MODEM STATUS
                                ;       CHANGE IS LOGGED.
                                ;
                                ;       IF A DATA INTERRUPT, THE ROUTINE CHECK FOR AN OVERRUN
                                ;       CONDITION AND IF SET
                                ;
                                ; INPUTS:
                                ;       RMSGPT - ADDRESS OF RX BUFFER
                                ;       RMSCC  - COUNT OF DATA TO BE RXED.
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;               "LOGMSC"        - LOG MODEM STATUS CHANGE
                                ;               "LGDVE"         - LOG DEVICE ERROR
                                ;--
                                
 044510                                 BGNSRV  PRRXI
 044510                                                                          PRRXI::
 044510  010246                         MOV     R2,-(SP)        ;SAVE R2
 044512  017737  144732  011474         MOV     @RXCSR,IRXCSR   ;MOV RX CSR TO IMAGE
 044520  032737  000010  006576         BIT     #MOCHK,PARAM    ;ANY MODEM CHANGES TO REPORT
 044526  001453                         BEQ     PRIN21          ;IF NOT IGNORE DS CHANGE.
 044530  032737  002000  037116         BIT     #INOVR,IMFLAG   ;IS INIT OVER
 044536  001447                         BEQ     PRIN21          ;NO THEN IGNORE DS CHANGE.
 044540  032737  004000  037116         BIT     #FIRST,IMFLAG   ;FIRST TIME HERE?
 044546  001443                         BEQ     PRIN21          ;YES,BRANCH
 044550  005737  011474                 TST     IRXCSR          ;DATA SET CHANGE ?
 044554  100040                         BPL     PRIN21          ;IF DATA SET CHANGE IS NOT SET BR
 044556  013737  011474  011472         MOV     IRXCSR,CMODS    ;MOV THE NEW MODEM STATUS IN
 044564  042737  104761  011472         BIC     #104761,CMODS   ;CLEAR BITS NOT RELATING TO MODEM STATUS
 044572  013737  011472  006542 PRIN2:  MOV     CMODS,TEMP3
 044600  013737  007556  006544         MOV     MODS,TEMP4
 044606  023737  006544  006542         CMP     TEMP4,TEMP3     ;COMPARE OLD TO CURRENT
 044614  001406                         BEQ     GLINC           ;INC GLITCH COUNT
 044616  005237  011520                 INC     MHRCNT          ;INC HARD COUNT
 044622  012737  016717  006540         MOV     #HRDMSG,TEMP2   ;SET UP HARD MESG.
 044630  000405                         BR      PRIN1
 044632  005237  011516 GLINC:  INC     MGLCNT          ;INC GLITCH COUNT
 044636  012737  016671  006540         MOV     #GLMSG,TEMP2    ;SET UP GLITCH
 044644  004737  020336 PRIN1:  JSR     PC,LOGMSC       ;GO LOG MODEM STATUS CHANGE
 044650  013737  011472  007556         MOV     CMODS,MODS      ;MOVE CURRENT TO OLD

                                ;;TEST FOR DATA
```

```
 7640
 7641  044656  032737  000200  011474  PRIN21: BIT     #RXDONE,IRXCSR  ;RX DONE ?
 7642  044664  001002                          BNE     10$             ;YES,BRANCH
 7643  044666  000137  045332                  JMP     PRINEX          ;EXIT
 7644  044672  017737  144556  011476  10$:    MOV     @RXDBUF,IRXDBUF ;READ DATA
 7645  044700  032737  100000  011476          BIT     #RERR,IRXDBUF   ;OVERRUN ERROR ?
 7646  044706  001414                          BEQ     PRIN4           ;NO,BRANCH
 7647
 7648                                  ;;IF AN OVERRUN THEN LOG ERROR,SET NAK REASON,TURN OFF RX & EXIT
 7649  044710  052737  000010  037052          BIS     #SNAK,PRSTAT    ;TELL MAIN CODE ABOUT OVERRUN ERROR
 7650  044716  105237  037106                  INCB    LOSTER          ;LOG LOCAL STATION ERROR
 7651  044722  152737  000001  037107          BISB    #RXOVMSK,LSMASK ;SET RX OVERRUN MASK BIT
 7652  044730  112737  000011  037070          MOVB    #RXOVRUN,REANAK ;SET REASON FOR SENDING NAK
 7653  044736  000570                          BR      PRIN8           ;GO TURN OFF RX AND EXIT
 7654
 7655
 7656                                  ;;;IF IN MULTI-POINT MODE AND NOT MY ADDRESS THEN JUST BUMP CHAR COUNT
 7657
 7658                                  ;;STORE AWAY DATA
 7659  044740  032737  001000  037052  PRIN4:  BIT     #MYDATA,PRSTAT  ;STORE THIS DATA ?
 7660  044746  001406                          BEQ     10$             ;NO,BRANCH
 7661  044750  013702  011510                  MOV     RMSGPT,R2       ;SET RX MESSAGE POINTER
 7662  044754  113722  011476                  MOVB    IRXDBUF,(R2)+   ;STORE DATA AWAY
 7663  044760  010237  011510                  MOV     R2,RMSGPT       ;SAVE UPDATED MESSAGE POINTER
 7664
 7665                                  ;;DECREMENT CHARACTER COUNT
 7666  044764  005337  011512          10$:    DEC     RMSGCC          ;ALL DATA RECEIVED ?
 7667  044770  001160                          BNE     PRINEX          ;NO,BRANCH
 7668  044772  032737  000400  037140          BIT     #BCC,PRFLAG     ;CHECK CRC ?
 7669  045000  001410                          BEQ     PRIN6           ;YES,BRANCH
 7670  045002  032737  010000  011476          BIT     #CRCOK,IRXDBUF  ;CRC GOOD ?
 7671  045010  001016                          BNE     PRIN5           ;YES,BRANCH
 7672  045012  052737  000002  037052          BIS     #BCCBAD,PRSTAT  ;TELL MAIN CODE ABOUT CRC ERROR
 7673  045020  000537                          BR      PRIN8           ;DISABLE INTERRUPTS AND EXIT
 7674
 7675                                  ;;;IN ORDER TO CHECK CRC, WE MUST READ 2 MORE CHARACTERS(CRC)
 7676  045022  052737  000400  037140  PRIN6:  BIS     #BCC,PRFLAG     ;SET CRC ALREADY CHECKED FLAG
 7677  045030  012737  000002  011512          MOV     #2,RMSGCC       ;COUNT TWO CHARACTERS
 7678  045036  012737  011514  011510          MOV     #BCCW,RMSGPT    ;CRC STORAGE ADDRESS
 7679  045044  000532                          BR      PRINEX          ;EXIT
 7680
 7681  045046  052737  000001  037052  PRIN5:  BIS     #BCCOK,PRSTAT   ;TELL MAIN CODE CRC HAS BEEN CHECKED
 7682  045054  123737  037065  002664          CMPB    TRIBN,RHDADR    ;MY MESSAGE
 7683  045062  001404                          BEQ     5$              ;YES,BRANCH
 7684  045064  042737  001000  037052          BIC     #MYDATA,PRSTAT  ;DON'T STORE IT
 7685  045072  000407                          BR      7$              ;BRANCH
 7686                                  ;; SELECT BIT SET ?
 7687  045074  032737  100000  002660  5$:     BIT     #BIT15,RHDMCC   ;SELECT BIT SET?
 7688  045102  001403                          BEQ     7$              ;NO,BRANCH
 7689  045104  012737  000001  037134          MOV     #1,SELECT       ;WE NOW HAVE THE RIGHT TO TRANSMIT,IF HALF-DUPL
 7690
 7691  045112  032737  000040  037140  7$:     BIT     #RXM,PRFLAG     ;READ DATA MESSAGE ?
 7692  045120  001071                          BNE     PRIN7           ;NO,BRANCH
 7693
 7694                                  ;;SET UP TO READ IN DATA PART OF MESSAGE
 7695  045122  042737  000003  037052          BIC     #BCCOK!BCCBAD,PRSTAT ;CLEAR FLAGS (USED IN PROTOCOL CODE)
```

```
                                        RECEIVER PROTOCOL INTERRUPT ROUTINE

7696  045130  052737  000040  037140        BIS     #RXM,PRFLAG     ;SET DATA MESSAGE READ FLAG
7697  045136  042737  000400  037140        BIC     #BCC,PRFLAG     ;CLEAR CRC CHECKED FLAG(USED BY THIS ROUTINE)
7698  045144  042737  140000  002660        BIC     #BIT15!BIT14,RHDMCC ;CLEAR SELECT & QS BITS
7699
7700                                  ;;IS ALLOCATED BUFFER SPACE LARGE ENOUGH FOR MESSAGE?
7701  045152  023727  002660  001000        CMP     RHDMCC,#512.    ;WILL MESSAGE FIT IN ALLOCATED BUFFER?
7702  045160  003414                        BLE     10$             ;YES,BRANCH
7703
7704                                  ;;MESSAGE TOO LONG !! LOG ERROR
7705  045162  105237  037100              INCB    LBUFER          ;LOG LOCAL BUFFER ERROR
7706  045166  152737  000002  037101        BISB    #MTLMSK,LBMASK  ;SET MESSAGE TOO LONG BIT
7707  045174  112737  000020  037070        MOVB    #MESLONG,REANAK ;SET REASON FOR NAK
7708  045202  152737  000010  037052        BISB    #SNAK,PRSTAT    ;SET SEND NAK FLAG
7709  045210  000443                        BR      PRIN8           ;TURN OFF RX & EXIT
7710
7711                                  ;; IF A NEW BUFFER IS AVAILABLE
7712                                  ;;SET BUFFER AND CHARACTER COUNT FOR MESSAGE
7713  045212  005737  037120        10$:    TST     RXPRC           ;NEW BUFFER AVAILABLE ?
7714  045216  001420                        BEQ     15$             ;YES,BRANCH
7715  045220  105237  037100              INCB    LBUFER          ;LOCAL BUFFER ERROR
7716  045224  001003                        BNE     12$             ;OVERFLOW?
7717  045226  012737  000377  037100        MOV     #377,LBUFER     ;LATCH A 255.
7718  045234  152737  000001  037101  12$:  BISB    #BNAMSK,LBMASK  ;SET MASK
7719  045242  112737  000010  037070        MOVB    #BUFFNA,REANAK  ;SET NAK REASON
7720  045250  152737  000010  037052        BISB    #SNAK,PRSTAT    ;SET "SEND NAK FLAG"
7721  045256  000412                        BR      PRIN7           ;EXIT
7722
7723  045260  013737  006470  011510  15$:  MOV     DVRXA,RMSGPT    ;MESSAGE BUFFER ADDRESS
7724  045266  013737  002660  011512        MOV     RHDMCC,RMSGCC   ;CHARACTER COUNT OF MESSAGE
7725  045274  013737  002660  006472        MOV     RHDMCC,DVRCC    ;TELL MAIN CODE HOW LARGE MESSAGE IS
7726  045302  000413                        BR      PRINEX          ;EXIT
7727
7728                                  ;;MESSAGE COMPLETE
7729  045304  052737  000004  037140  PRIN7: BIS     #QRX,PRFLAG     ;SET MESSAGE COMPLETE FLAG(USED BY MAIN CODE)
7730  045312  052737  000100  037052        BIS     #RXD,PRSTAT     ;MESSAGE COMPLETE(USED BY PROTOCOL MODULE)
7731
7732  045320  005037  037146        PRIN8:  CLR     TURNON          ;RX NOT ON
7733  045324  042777  000120  144116        BIC     #RINTEN+RXENA,@RXCSR    ;TURN OFF RECEIVER
7734
7735  045332  012602                  PRINEX: MOV     (SP)+,R2        ;RESTORE R2
7736  045334                                  ENDSRV
7737  045334                                                                  L10022:
7738  045334  000002                                                                  RTI
7739
7740
7741
```

```
 7742
 7743                                   .SBTTL  PROTOCOL TRANSMIT ROUTINE
 7744
 7745                                   ;++
 7746                                   ;; FUNCTIONAL DESCRIPTION:
 7747                                   ;;      THIS ROUTINE IS USED TO SETUP EITHER CONTROL MESSAGES OR
 7748                                   ;;      DATA MESSAGES FOR TRANSMISSION.
 7749                                   ;;      IF THE SEND ACK(SACK) IS SET AN 'ACK' MESSAGE WILL BE SETUP
 7750                                   ;;      AND TRANSMITTED.
 7751                                   ;;      IF THE SEND NAK(SNAK) IS SET A 'NAK' MESSAGE WILL BE SETUP
 7752                                   ;;      AND TRANSMITTED.
 7753                                   ;;      ELSE A DATA MESSAGE WILL BE SETUP AND SENT.
 7754                                   ;;      IF THE NETWORK IS HALF-DUPLEX THEN REQUEST TO SEND(RTS) WILL
 7755                                   ;;      BE ASSERTED BEFORE TRANSMISSION.
 7756
 7757
 7758
 7759  045336  013737  037052  037136  TXPROT: MOV     PRSTAT,IMPRSTAT ;SAVE A COPY OF FLAGS
 7760  045344  032737  000100  006576          BIT     #PRORUN,PARAM   ;PROTOCOL RUNNING ?
 7761  045352  001407                          BEQ     7$              ;NO,BRANCH
 7762  045354  022737  000003  006570          CMP     #ACT,MODTYP     ;ACTIVE MODE?
 7763  045362  001003                          BNE     7$              ;NO,BRANCH
 7764  045364  005737  037126          5$:     TST     TXREADY         ;TRANSMITTER READY FOR MESSAGE ?
 7765  045370  001775                          BEQ     5$              ;NO,BRANCH
 7766
 7767                                   ;;IF HALF DUPLEX OR MULTI-POINT LINK, WE NEED THE SELECT BIT
 7768                                   ;;; BEFORE WE CAN SEND.
 7769  045372  005737  037142          7$:     TST     HDXMTP          ;FULL DUPLEX AND PT TO PT ?
 7770  045376  001406                          BEQ     8$              ;YES,BRANCH
 7771  045400  005737  037134          6$:     TST     SELECT          ;OK TO SEND ?
 7772  045404  001003                          BNE     8$              ;YES,BRANCH
 7773  045406  004737  042104                  CALL    RXPROTO         ;GO WAIT ON SELECT BIT
 7774  045412  000772                          BR      6$              ;TRY AGAIN
 7775
 7776                                   ;; DETERMINE WHAT TO SEND
 7777  045414  013737  037136  037052  8$:     MOV     IMPRSTAT,PRSTAT ;RESTORE ORIGINAL FLAGS
 7778  045422  113737  037065  002652          MOVB    TRIBN,HDMADR    ;SET TRIB ADDRESS
 7779  045430  042737  001000  006602          BIC     #PAD,FLAG       ;THIS BIT USED IN TX INTER ROUTINE
 7780  045436  005037  037126                  CLR     TXREADY         ;TRANSMITTER BUSY
 7781  045442  005037  037134                  CLR     SELECT          ;IF HALF DUPLEX/MTP MODE
 7782  045446  032737  000004  037052          BIT     #SACK,PRSTAT    ;SEND ACK ?
 7783  045454  001021                          BNE     10$             ;YES,BRANCH
 7784  045456  032737  000010  037052          BIT     #SNAK,PRSTAT    ;SEND NAK ?
 7785  045464  001062                          BNE     50$             ;YES, BRANCH
 7786  045466  032737  004000  037052          BIT     #SSTART,PRSTAT  ;SEND START ?
 7787  045474  001106                          BNE     60$             ;YES,BRANCH
 7788  045476  032737  002000  037052          BIT     #SSTACK,PRSTAT  ;SEND START ACKNOWLEDGE ?
 7789  045504  001126                          BNE     70$             ;YES,BRANCH
 7790  045506  032737  000020  037052          BIT     #SDATA,PRSTAT   ;SEND DATA MESSAGE ?
 7791  045514  001151                          BNE     100$            ;YES,BRANCH
 7792  045516  000000                          HALT                    ;FATAL ERROR
 7793
 7794
 7795
 7796                                   ;; SETUP TO SEND AN 'ACK'
 7797  045520  052737  000020  006602  10$:     BIS     #TXM,FLAG       ;SEND HEADER ONLY(USED IN TX INTER. ROUTINE)
```

```
 7798  045526  112737  000005  002645          MOVB    #ENQ,HDMID      ;CONTROL MESSAGE
 7799  045534  012737  000001  002646          MOV     #ACK,HDMTYP     ;ACK CONTROL MESSAGE
 7800  045542  052737  140000  002646          BIS     #BIT15!BIT14,HDMTYP ;SET SELECT & QS FLAG
 7801  045550  005737  037142                  TST     HDXMTP          ;HALF DUPLEX OR MULTI - POINT
 7802  045554  001415                          BEQ     20$             ;NO,BRANCH
 7803  045556  005737  037122                  TST     TXPRC           ;ANY THING TO SENT ?
 7804  045562  001012                          BNE     20$             ;NO,BRANCH
 7805  045564  032737  000100  006576          BIT     #PRORUN,PARAM   ;PROTOCOL RUNNING?
 7806  045572  001406                          BEQ     20$             ;NO,BRANCH
 7807  045574  042737  100000  002646          BIC     #BIT15,HDMTYP   ;CLEAR SELECT BIT
 7808  045602  012737  000001  037134          MOV     #1,SELECT       ;WE HAVE SOMETHING TO SEND, SO KEEP THE LINE
 7809  045610  113737  037064  002650  20$:    MOVB    R,HDMREP        ;SET RESPONSE NUMBER
 7810  045616  105037  002651                  CLRB    HDMNUM          ;FILLER
 7811  045622  042737  000004  037052          BIC     #SACK,PRSTAT    ;CLEAR SEND ACK FLAG
 7812  045630  000526                          BR      200$            ;GO SEND IT
 7813
 7814
 7815                                  ;; SETUP TO SEND A 'NAK'
 7816  045632  052737  000020  006602  50$:    BIS     #TXM,FLAG       ;TELL TX INTERRUPT TO SEND HEADER ONLY
 7817  045640  112737  000005  002645          MOVB    #ENQ,HDMID      ;CONTROL MESSAGE
 7818  045646  012737  000002  002646          MOV     #NAK,HDMTYP     ;'NAK'
 7819  045654  113737  037070  002647          MOVB    REANAK,HDMTYP+1 ;REASON FOR NAK
 7820  045662  052737  140000  002646  55$:    BIS     #BIT15!BIT14,HDMTYP ;SET SELECT & QS FLAGS
 7821  045670  105037  002651                  CLRB    HDMNUM          ;FILLER
 7822  045674  113737  037064  002650          MOVB    R,HDMREP        ;LAST MESSAGE RECEIVED CORRECTLY
 7823  045702  042737  000010  037052          BIC     #SNAK,PRSTAT    ;CLEAR SEND NAK FLAG
 7824  045710  000476                          BR      200$            ;GO SEND IT
 7825
 7826
 7827                                  ;; SETUP TO SEND START MESSAGE
 7828  045712  052737  000020  006602  60$:    BIS     #TXM,FLAG       ;TELL TX INT. ROUTINE TO SEND HEADER ONLY
 7829  045720  112737  000005  002645          MOVB    #ENQ,HDMID      ;CONTROL MESSAGE
 7830  045726  012737  000006  002646          MOV     #STRT,HDMTYP    ;START MESSAGE
 7831  045734  052737  140000  002646          BIS     #BIT15!BIT14,HDMTYP ;SET SELECT & QS FLAGS
 7832  045742  105037  002650                  CLRB    HDMREP          ;FILLER
 7833  045746  105037  002651                  CLRB    HDMNUM          ;FILLER
 7834  045752  042737  004000  037052          BIC     #SSTART,PRSTAT  ;CLEAR SEND START FLAG
 7835  045760  000452                          BR      200$            ;GO SEND IT
 7836
 7837                                  ;; SETUP TO SEND STACK MESSAGE
 7838  045762  052737  000020  006602  70$:    BIS     #TXM,FLAG       ;TELL TX INT. TO SEND HEADER ONLY
 7839  045770  112737  000005  002645          MOVB    #ENQ,HDMID      ;CONTROL MESSAGE
 7840  045776  012737  000007  002646          MOV     #STACK,HDMTYP   ;START ACKNOWLEDGE MESSAGE
 7841  046004  052737  140000  002646          BIS     #BIT15!BIT14,HDMTYP     ;SET SELECT &QS FLAGS
 7842  046012  105037  002650                  CLRB    HDMREP          ;FILLER
 7843  046016  105037  002651                  CLRB    HDMNUM          ;FILLER
 7844  046022  012737  177777  037124          MOV     #-1,ASTRT       ;START HAS BEEN ACKNOWLEDGED
 7845  046030  042737  002000  037052          BIC     #SSTACK,PRSTAT  ;CLEAR SEND STACK FLAG
 7846  046036  000423                          BR      200$            ;GO SEND IT
 7847
 7848
 7849                                  ;; SETUP TO SEND DATA
 7850  046040  042737  000020  006602  100$:   BIC     #TXM,FLAG       ;TELL TX INTERRUPT TO SEND HEADER + DATA
 7851  046046  112737  000201  002645          MOVB    #SOH,HDMID      ;DATA MESSAGE
 7852  046054  013737  006456  002646          MOV     DVTCC,HDMCC     ;CHARACTERS COUNT
 7853  046062  052737  140000  002646          BIS     #BIT15!BIT14,HDMCC  ;SET SELECT & QS FLAGS
```

```
 7854  046070  113737  037064  002650          MOVB    R,HDMREP        ;LAST MESSAGE RECEIVED CORRECTLY
 7855  046076  113737  037062  002651          MOVB    T,HDMNUM        ;THIS MESSAGE NUMBER
 7856  046104  000400                          BR      200$            ;GO SEND IT
 7857
 7858                                  ;; GO SET 'REQUEST TO SEND'
 7859  046106  004737  036576          200$:   JSR     PC,CTSSR        ;GO SET REQUEST TO SEND
 7860  046112  052737  004000  037116          BIS     #FIRST,IMFLAG   ;TELL THE CTSSR SUBROUTINE TO SKIP DELAY
 7861
 7862                                  ;; SETUP TO TRANSMIT HEADER PORTION OF MESSAGE
 7863  046120  012737  002645  011500  210$:   MOV     #HDMID,MSGPTR   ;HEADER MESSAGE ADDRESS
 7864  046126  012737  000006  011502          MOV     #6,MSGCC        ;CHARACTER COUNT OF HEADER = 6
 7865  046134  012737  000020  011504          MOV     #20,SYNCC       ;NUMBER OF SYNCS TO TRANSMIT
 7866
 7867                                  ;; SEND THE DATA
 7868  046142  005737  006574                  TST     FHDPLX          ;FULL DUPLEX?
 7869  046146  001004                          BNE     215$            ;YES,BRANCH
 7870  046150  052777  000130  143300          BIS     #SEND!TINTEN!HDPLX,@TXCSR ;ENABLE FOR HALF DUPLEX
 7871  046156  000403                          BR      217$            ;
 7872
 7873  046160  052777  000120  143270  215$:   BIS     #SEND!#TINTEN,@TXCSR  ;TURN ON TRANSMITTER
 7874
 7875
 7876                                  ;;IF ACTIVE MODE, TURN ON TX AND GET OUT IN A HURRY
 7877                                  ;; NOTE: START UP SEQUENCE OPERATES LIKE HALF-DUPLEX
 7878
 7879  046166  005737  037142          217$:   TST     HDXMTP          ;FULL DUPLEX PT-PT?
 7880  046172  001005                          BNE     220$            ;NO,BRANCH
 7881  046174  022737  000003  006570          CMP     #ACT,MODTYP     ;ACTIVE MODE ?
 7882  046202  001001                          BNE     220$            ;NO,BRANCH
 7883  046204  000406                          BR      TXPREX          ;EXIT
 7884
 7885  046206                          220$:   BREAK
 7886  046206  104422                                                          TRAP    C$BRK
 7887  046210  005737  037126                  TST     TXREADY         ;TX FINISHED ?
 7888  046214  001774                          BEQ     220$            ;NO, BRANCH
 7889
 7890                                  ;; IF HALF-DUPLEX OR MULTI-POINT REQUEST TO SEND WILL BE DROPPED
 7891  046216  004737  037010          230$:   JSR     PC,CLRRTS       ;DROP RTS IF HALF DUPLEX
 7892
 7893  046222  000207                  TXPREX: RETURN                  ;WE ARE DONE !
 7894
 7895
 7896                                          .EVEN
 7897
 7898  046224                                  ENDTST
 7899  046224                                                          L10017:
 7900  046224  104401                                                          TRAP    C$ETST
 7901
 7902
 7903
 7904
```

```
                        .SBTTL  HARDWARE PARAMETER CODING SECTION


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

046226                          BGNHRD
046226  000030                                                    .WORD L10023-L$HARD/2
046230                                                      L$HARD::


                        .SBTTL          DEVICE INDEPENDENT SECTION

046230                          GPRML   DPLX,0,1,YES
046230  000130                                                  .WORD   T$CODE
046232  046310                                                  .WORD   DPLX
046234  000001                                                  .WORD   1


                        .SBTTL          DEVICE DEPENDENT SECTION

046236                          GPRMA   CSRADR,2,0,160000,177776,YES
046236  001031                                                  .WORD   T$CODE
046240  046341                                                  .WORD   CSRADR
046242  160000                                                  .WORD   T$LOLIM
046244  177776                                                  .WORD   T$HILIM
046246                          GPRMA   VECTOR,4,0,300,776,YES
046246  002031                                                  .WORD   T$CODE
046250  046367                                                  .WORD   VECTOR
046252  000300                                                  .WORD   T$LOLIM
046254  000776                                                  .WORD   T$HILIM
046256                          GPRML   RNODM,14,1,YES
046256  006130                                                  .WORD   T$CODE
046260  046422                                                  .WORD   RNODM
046262  000001                                                  .WORD   1
046264                          XFERT   ENDHWL
046264  012024                                                  .WORD   T$CODE
046266                          GPRML   PTPMLP,10,1,YES
046266  004130                                                  .WORD   T$CODE
046270  046446                                                  .WORD   PTPMLP
046272  000001                                                  .WORD   1
046274                          XFERF   ENDHWL
046274  006044                                                  .WORD   T$CODE
046276                          GPRMD   TRIBNQ,12,D,-1,1,255.,YES
046276  005052                                                  .WORD   T$CODE
046300  046504                                                  .WORD   TRIBNQ
046302  177777                                                  .WORD   -1
046304  000001                                                  .WORD   T$LOLIM
```

```
 7961  046306  000377                                                                   .WORD   T$HILIM
 7962  046310                                 ENDHWL: ENDHRD
 7963                                                                                   .EVEN
 7964  046310                                                                 L10023:
 7965
 7966                                         .NLIST  BEX

                                              ;DEVICE INDEPENDENT QUESTIONS

       046310  052506  046114  042040  DPLX:    .ASCIZ  /FULL DUPLEX OPERATION : /

                                              ;DEVICE DEPENDENT QUESTION

       046341     104  053105  041511  CSRADR: .ASCIZ  /DEVICE CSR ADDRESS : /
       046367     111  052116  051105  VECTOR: .ASCIZ  /INTERRUPT VECTOR ADDRESS: /
       046422  042522  047515  042524  RNODM:  .ASCIZ  /REMOTE NODE ''ITEP'':/
       046446  051511  052040  044510  PTPMLP: .ASCIZ  /IS THIS A MULTIPOINT NETWORK:/
       046504  042101  051104  051505  TRIBNQ: .ASCIZ  /ADDRESS THIS STATION:/

                                              .LIST   BEX
 7967                                                 .EVEN
 7968
 7969
```

```
                        ;.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
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
046532                  $PATCH:
046532  000030                  .BLKW   30
046612                          LASTAD
                                                                        .EVEN
046612  000000                                                          .WORD   0
046614  000000                                                          .WORD   0
046616                  L$LAST::
046616                          ENDMOD
                        .END    000001
```

```
A       037061            6706#   7247*   7346*   7415*
ABO   = 000026            2263#   4693
ABORT = 000200            2241#   4837    5580    7298
ACK   = 000001            6806#   7233    7799
ACT   = 000003            2214#   4819    5267    5347    5600    6954    7250    7425    7762    7881
ACTATV  031312            5042    5267#
ACTBCR  031116            5060    5229#
ACTCHK  031526            5022    5313#
ACTCLB  030440            5118    5132#
ACTCLP  031640            5056    5341#
ACTCLR  030130            5020    5073#
ACTCOP  030736            5030    5192#
ACTCRC  031542            5051    5319#
ACTCSE  030264            5025    5104#
ACTCST  030356            5026    5120#
ACTDLL  031360            5046    5281#
ACTDME  030664            5062    5173    5176#
ACTDMQ  030656            5063    5175#
ACTDMS  030634            5061    5170#
ACTDMX  030672            5177#
ACTECH  031436            5050    5297#
ACTEQO  031060            5034    5218#
ACTERR  021414            3491    3531#
ACTEXT  030214            5066    5092#
ACTFUL  021372            3492    3525#
ACTHLP  030150            5024    5079#
ACTLIS  031350            5045    5278#
ACTLLP  031650            5057    5343#
ACTLPX  031666            5338    5340    5342    5344    5347#
ACTLXX  031730            5311    5332    5335    5348    5357#
ACTMEX  031304            5211    5227    5249    5254    5259    5262    5264#
ACTME1  031240            5238    5240    5242    5244    5246    5253#
ACTMOP  031620            5054    5337#
ACTMOS  031550            5065    5322#
ACTMS0  031140            5035    5237#
ACTMS1  031146            5036    5239#
ACTMS2  031156            5037    5241#
ACTMS3  031166            5038    5243#
ACTMS4  031176            5039    5245#
ACTMS5  031206            5040    5247#
ACTMS6  031224            5041    5250#
ACTM2X  031406            5268    5276    5279    5282    5285    5289#
ACTNO   031426            5049    5294#
ACTNUF  030120            5059    5070#
ACTNUL  030126            5019    5071#
ACTNUM  030746            5031    5195#
ACTOPM  031040            5032    5213#
ACTPAS  031322            5043    5270#
ACTPRO  031556            5052    5325#
ACTPRT  030224            5064    5095#
ACTQFG  031562            5314    5317    5320    5323    5327#
ACTREC  031342            5044    5275#
ACTREX  021346            3489    3516#
ACTRHL  021302            3488    3501#
ACTRLG  021356            3490    3520#
ACTRLP  031660            5058    5345#
```

```
ACTRNF  021272            3493    3497#
ACTRNL  021300            3487    3498#
ACTRPS  031610            5053    5334#
ACTRSO  021436            3494    3538#
ACTRUN  030240            5023    5099#
ACTSEX  031250            5067    5256#
ACTSHO  030140            5021    5076#  .
ACTSHW  030500            5110    5124   5143#  5162
ACTSTE  030700            5027    5182#
ACTSTS  031534            5033    5316#
ACTSTT  030710            5028    5185#
ACTSTX  030716            5183    5186#
ACTSZE  030726            5029    5189#
ACTTAL  031400            5048    5287#
ACTTLP  031630            5055    5339#
ACTTRA  031370            5047    5284#
ADDCC   023266            3934#   4951   4993
ADDC1   023362            3936    3951#
ADR   = 000020 G          2187#
ALCK    032336            2680    5499#
ALCK1   032460            5554    5564   5567#  5620
ALCK2   032524            5568    5577#  5652
ALCK2A  032774            5617    5622#
ALCK3   033020            5587    5624   5627#
ALCK3A  033112            5634    5640#
ALCK3B  033124            5641    5643#
ALCK3C  033120            5638    5642#  5649
ALCK4   033146            5628    5647#
ALCK4A  033154            5646    5650#
ALCK4B  033166            5651    5653#
ALCK5   032400            5553#   5642
ALLTR   032400            5431    5454   5478   5504   5552#
ASSEMB= 000010            1959
ASTRT   037124            6764#   6936*  7403   7844*
ATVMOD= 000027            2325#   2822
BAD     006555            2647#   3040   5728*
BADCHR= 000051            2343#   2918
BASM1   015507            3035#   3878
BASM2   015500            3035#   3897
BASM3   015471            3035#   3889
BCC   = 000400            2279#   6199   6202   6398   6416   6439   6554   7118   7668   7676   7697
BCCBAD= 000002            6816#   7168   7183   7672   7695
BCCOK = 000001            6815#   7168   7207   7386   7681   7695
BCCW    011514            2989#   6418   7678
BDCLK   014210            3035#   4564
BIT0  = 000001 G          2160#   2234   2267   3539   3610   6792   6795   6815
BIT00 = 000001 G          2149#   2160
BIT01 = 000002 G          2148#   2159
BIT02 = 000004 G          2147#   2158
BIT03 = 000010 G          2146#   2157
BIT04 = 000020 G          2145#   2156
BIT05 = 000040 G          2144#   2155
BIT06 = 000100 G          2143#   2154
BIT07 = 000200 G          2142#   2153
BIT08 = 000400 G          2141#   2152
BIT09 = 001000 G          2140#   2151
```

```
BIT1   = 000002 G          2159#   2235    2268    2377    3613    6793    6794    6816
BIT10  = 002000 G          2139#   2283    6823
BIT11  = 004000 G          2138#   2285    2371    2383    2387    6824
BIT12  = 010000 G          2137#   2368    2388    2395
BIT13  = 020000 G          2136#   2366
BIT14  = 040000 G          2135#   2370    7511    7698    7800    7820    7831    7841    7853
BIT15  = 100000 G          2134#   2393    2394    2396    7511    7687    7698    7800    7807    7820    7831    7841    7853
BIT2   = 000004 G          2158#   2236    2269    2369    3616    6791    6817
BIT3   = 000010 G          2157#   2237    2270    2378    6818
BIT4   = 000020 G          2156#   2238    2275    2379    2389    6819
BIT5   = 000040 G          2155#   2239    2277    2380
BIT6   = 000100 G          2154#   2240    2271    2381    2390    6820
BIT7   = 000200 G          2153#   2241    2272    2382    2386    5393    5396
BIT8   = 000400 G          2152#   2279    2384    2391    2397    6821
BIT9   = 001000 G          2151#   2281    2367    2385    2392    6077    6822
BLDBEX   023474            3995    3999#
BLDBUF   023364            3977#   4810    4817    4952    4994    5139    5405
BLDB1    023374            3982#
BLDB2    023436            3990#   3997
BLDB3    023454            3993#   3998
BNAMSK=  000001            6795#   7535    7718
BOE    = 000400 G          2191#
BUFEX    014355            3035#   4924    4965
BUFFNA=  000010            6783#   7530    7719
BUFLIM=  001000            2202#   2591    2592    2593    3935    3947    4066    4921    4962    5132    5401
BYTBIT   006514            2626#   3883    5171*   5175*
CABLE  = 000002            2221#   6046
CABLOP=  010000            2388#   6048
CBLLOP=  000045            2339#   2939
CCURAD   006452            2606#   4078*   4086    4090*   4806*   4814    4992    4997*   5117*
CHECK  = 000003            2305#   2870
CLEAR  = 000001            2303#   2808    4914    5073
CLIACT   027740            4872    5011#
CLIALN=  000007            2296#   2916
CLIALP=  000006            2295#   2856    2866    2921
CLIBCR   012025            3035#   5230
CLIBDL   011720            3035#   5352
CLIBIF=  000003            2292#
CLIBR  = 000002            2291#   2805    2807    2809    2812    2814    2823    2825    2827    2829    2831    2834    2836
                           2851    2853    2863    2871    2875    2877    2881    2883    2886    2889    2896    2898    2900
                           2902    2904    2906    2908    2910    2914    2917    2924    2926    2931    2938    2940    2942
                           2944    2949    3470
CLIBRX   011700            3035#   4323
CLIDEC=  000011            2298#   2930    2948
CLIERM   011603            3035#   3434    4878
CLIERR=  000000            2289#   2815    2918    3480
CLIEXI=  000001            2290#   2797    2799    2801    2803    2842    2844    2954    3462    3464    3466    3468    3473
                           3475    3479    3481
CLINBG   011656            3035#   4366
CLINPS   011762            3035#   5303
CLINUF   011633            3035#   3444    4887
CLINUM=  000005            2294#
CLIOCT=  000010            2297#   2858    2860    3478
CLIPW    012123            3035#   5369
CLIRAC   021234            3428    3482#
CLIRT    021032            3427    3459#
```

```
CLISEO  012072          3035#  5200
CLISPA= 000004          2293#  2795  2819  2840  2849  2857  2915  3460
CLISTR= 000012          2299#  2798  2800  2802  2804  2806  2808  2811  2813  2820  2822  2824  2826
                        2828   2830  2833  2835  2841  2843  2850  2852  2868  2870  2874  2876  2880
                        2882   2885  2888  2895  2897  2899  2901  2903  2905  2907  2909  2923  2925
                        2937   2939  2941  2943  3463  3465  3467  3469  3472  3474  3476
CLITRE  007714          2791#  4871
CLI$PM  011570          3035#  4865
CLI$RP  011576          3035#  3421
CLKBR   006624          2688#  4549*
CLKCSR  006622          2687#  3221*  3242*  4520  4538*  4539*  4540*  4550*  4685*  4773*
CLKEN   006632          2691#  3242   4528*  4541*  4550   4773
CLKHZ   006630          2690#  3224   3239   3376   4554   4572
CLKINT  017756  G       3219#  4623
CLKSET  017732          3174#  4527   4537
CLKVEC  006626          2689#  4518*  4548*  4624
CLRRTS  037010          6144   6298   6310   6579   6664#  7891
CMDBUF  002666          2528#  3419   3426   4863   4870
CMODS   011472          2980#  6367*  6368*  6369   6379   7625*  7626*  7627   7637
CMPBUF  005150          2593#  4064*  4078   4806   5116
CMPPTR  006444          2603#  4072*  4073*  4086*  4087*  4088*  4092*  4801*  4813   4976*  4990*  4991   4995*
                        5104*  5105   5114*  5115   5379*  5692
CMPSEX  033442          5690   5778#
CMPSR   033166          5639   5689#
CMPS1   033326          5721#  5739
CMPS2   033332          5724#
CMPS3   033220          5695#  5751
CMPS5   033430          5748#
CMPS5A  033434          5741   5749#
CMPS6   033376          5722   5726   5737#
CMPS7   033314          5710   5719#
CMPTOT  006446          2604#  4081*  4084   4093*  4818*  4974*  4977   4996*  5107   5111*  5366   5382
CMSG0 = 000020          2318#  2897
CMSG1 = 000021          2319#  2895
CMSG2 = 000022          2320#  2899
CMSG3 = 000023          2321#  2901
CMSG4 = 000024          2322#  2905
CMSG5 = 000025          2323#  2903
CMSG6 = 000026          2324#  2907
CONOTM  006550          2644#  3122
CONTIN  006552          2645#
CPTR    006524          2634#  3980   3984*  4809*  4813*  4949*  4953   4991*  4995   5115*  5128*  5403*  5430*
                        5450*  5475*  5500*  5569   5574*  5593   5643*  5650   5691*  5696   5700*  5946*
CPTRR   006522          2633#  5427*  5453*  5476*  5502*  5555   5561*  5599*  5604   5613*  5626*  5647   5692*
                        5702   5705*  5894*
CR      016534          3035#  3360
CRC   = 000040          2334#
CRCB  = 000020          2238#  5319
CRCOK = 010000          2395#  6400   7670
CSHEXP= 000006          2308#  2841
CSHTRN= 000007          2309#  2843
CSRADR  046341          7936   7966#
CTOTCC  006450          2605#  4080*  4088   4090   4091*  4096*  4796*  4961   4972   4998*  5112*
CTS   = 020000          2366#  2714   6635
CTSSR   036576          6093   6104   6255   6504   6606#  7859
CTSS3   036646          6613   6621#
```

```
CTSS4   036624          6615#  6619
CTSS7   036706          6623   6626   6631#
CURADD  006526          2635#  3982   3987   3989   3999*  4808*  4814*  4950*  4955   4992*  4997   5133*  5402*
CURCC   006520          2632#  3934   3946   3949*  3950   3983   3988   3999   4804*  4816*  5135*  5207*  5220*
                        5248*  5251*  5253*  5401*
C$AU  = 000052          1959#  4753
C$AUTO= 000061          1959#  4674
C$BRK = 000022          1959#  6020   6075   6114   6278   6574   6617   6634   7153   7886
C$BSEG= 000004          1959#
C$BSUB= 000002          1959#
C$CEFG= 000045          1959#
C$CLCK= 000062          1959#  4523   4533
C$CLEA= 000012          1959#  4710
C$CLOS= 000035          1959#
C$CLP1= 000006          1959#
C$CVEC= 000036          1959#
C$DCLN= 000044          1959#  4491
C$DODU= 000051          1959#
C$DRPT= 000024          1959#
C$DU  = 000053          1959#  4731
C$EDIT= 000003          1959#  2023
C$ERDF= 000055          1959#
C$ERHR= 000056          1959#
C$ERRO= 000060          1959#
C$ERSF= 000054          1959#
C$ERSO= 000057          1959#  5713   5731   5744   5787   6031   6069   6137   6290   6408   6430   6648   7468
                        7486
C$ESCA= 000010          1959#
C$ESEG= 000005          1959#
C$ESUB= 000003          1959#
C$ETST= 000001          1959#  7900
C$EXIT= 000032          1959#  4650   4702   4851   4911
C$GETB= 000026          1959#
C$GETW= 000027          1959#
C$GMAN= 000043          1959#  3417   4552   4861   5872
C$GPHR= 000042          1959#  4587
C$GPLO= 000030          1959#
C$GPRI= 000040          1959#
C$INIT= 000011          1959#  4658
C$INLP= 000020          1959#
C$MANI= 000050          1959#  4845
C$MEM = 000031          1959#
C$MSG = 000023          1959#  3051   3064   3078   3092   3112   3132
C$OPEN= 000034          1959#
C$PNTB= 000014          1959#  3047   3060   3074   3088   3108   3128
C$PNTF= 000017          1959#  3363   3371   3409   3437   3447   3507   3567   3881   3892   3900   3944   4326
                        4369   4567   4786   4828   4881   4890   4928   4943   4969   4985   5085   5158   5203
                        5233   5306   5355   5372   5842   5939   5966   7305
C$PNTS= 000016          1959#  3553   3588   3602   3631   3652   3675   3689   3702   3715   3723   3750   3764
                        3779   3794   3802   3822   3842   4143   4180
C$PNTX= 000015          1959#
C$QIO = 000377          1959#
C$RDBU= 000007          1959#  4545
C$REFG= 000047          1959#  4496   4501   4506   4512
C$RESE= 000033          1959#  4699
C$REVI= 000003          1959#  2022
```

```
C$RFLA= 000021           1959#
C$RPT = 000025           1959#   4459
C$SEFG= 000046           1959#
C$SPRI= 000041           1959#   4648    4688
C$SVEC= 000037           1959#   4626    4636    4643    4835    6923
C$TPRI= 000013           1959#
DATABC= 000002           6781#   7394    7521    7526
DATAWO  006556           2648#   6505*   6507*   6509
DATCKB= 000002           2235#   4152    5313    5364    5449    5596    5645    5689    5866    5934
DCD   = 010000           2368#   2716    6625
DCK   = 000014           2256#   3317
DCLFLG  006564           2654#   4487    4489*   4909*
DDE   = 000022           2261#   3325
DER   = 000010           2254#   3303
DERIN   037076           6731#   7188*   7190*   7280*   7282*   7391*   7393*
DEROUT  037074           6725#   7514*   7516*   7523*   7525*
DEV1    007704           2771#   3708*   3719    3727*   3729*   4124    4838*   5163*
DEV2    007706           2772#   3709*   3718    3728*   3730*   4127    4839*   5164*
DEV3    007710           2773#   3731*   4134    4840*   5165*
DEV4    007712           2774#   3732*   4148    4152    4156    4161    4166    4841*   5166*
DFPTBL  002130 G         2088#
DIAGMC= 000000           1959
DIVN91  034772           6111#   6116
DLE   = 000020           2260#   3321
DLL     033570           2681    5837#
DLLCM   014130           3035#   5839
DLLMOD= 000033           2329#   2830
DMPE  = 000053           2345#   2860
DMPQ  = 000054           2346#   2862
DMPS  = 000052           2344#   2858    4903    5172
DMSGAD  002172           2423#   3991
DMSGCT  002150           2410#   3990
DOW   = 000004           2215#   5281    6051
DPLX    046310           7926    7966#
DPV   = 000000           2246#
DSCA  = 100000           2396#
DSITEN= 000040           2380#   6212    6560    7126
DSR   = 001000           2367#   2715
DTR   = 000002           2377#   6055
DUMEX   023264           3903    3909#
DUMPSR  023130           3874#   4905
DUM1    023222           3884    3895#
DUM2    023244           3894    3902#
DUM3    023160           3883#   3907
DUM4    023134           3875#   3906
DUPPAR  011470           2973#   5393*   5396*   6079
DVEM0   016747           3035#   6025    6033
DVEM1   017026           3035#   6642    6650
DVEM2   017110           3035#   6284    6292    7480    7488
DVEM3   017205           3035#   6404    6410
DVEM4   017246           3035#   6424    6432
DVEM5   017313           3035#   6131    6139    7462    7470
DVEM6   017377           3035#   6063    6071
DVI   = 000012           2255#   3308
DVINEX  035130           6053    6081    6083    6085    6112    6124    6144#
DVINIT  034306           5399    6010#   6035    6142
```

```
DVIN1    034404            6018    6039#
DVIN3    034440            6042    6046#
DVIN3A   034474            6052    6055#   6073
DVIN31   036432            6095    6120    6554#
DVIN32   036462            6564#   6584
DVIN34   036574            6576    6585#
DVIN35   036524            6573#   6578
DVIN36   036556            6581#   6582
DVIN37   034602            6044    6049    6079#
DVIN38   034510            6058#   6078
DVIN39   034562            6059    6074#
DVIN4    034702            6099#
DVIN41   034662            6094#   6102
DVIN8    035014            6100    6119#   6127
DVIN81   035056            6097    6117    6122    6131#
DVIN9    034722            6104#   6126
DVMEX    035160            6172#
DVMODS   035152            3340    6170#   6624*
DVRCC    006472            2616#   5560*   5589    5943*   5958    5960*   5961*   6205*   6442*   7725*
DVRCT    006474            2617#   5428*   5501*   5598*   5602*   5622*   5623    5693*   5750*
DVREX    035270            6198    6213#
DVRXA    006470            2615#   5558*   5588    5941*   5957    5960    6203    6440    7723
DVRXI    035556  G         4633    4832    6358#
DVRXQ    035170            5565    5952    6196#
DVRX2    035246            6201    6210#
DVRX3    035262            6206    6212#
DVTCC    006456            2609#   5573*   5630    5891*   5905    6260    6266    6517    7852
DVTCT    006460            2610#   5451*   5474*   5499*   5632*   5637    5640
DVTREX   035554            6300    6306    6311#
DVTR1    035346            6257    6264#
DVTR2    035376            6261    6268#
DVTR3    035412            6253    6275#   6294
DVTR4    035520            6297    6302#
DVTR5    037050            6668    6673#
DVTR8    035420            6277#   6303    6309
DVTXA    006454            2608#   5571*   5629    5888*   5904    6259    6516
DVTXI    036230  G         4640    6489#
DVTXRX   035272            5584    5899    5954    6252#
DVTXR1   037006            6636    6653#
DVTXR2   036722            6633#   6638
DVTXR3   036606            6612#   6652
DVTXR4   036742            6628    6642#
DVTXR9   037006            6608    6654#
ECHO   = 000037            2333#   2876
ECHOB  = 000004            2236#   4156    5289    5299    5309    5591
EDABO    015460            2745    3035#
EDDCK    015252            2740    3035#
EDDDE    015365            2743    3035#   5733    5746
EDDER    015235            2738    3035#
EDDLE    015330            2742    3035#   5715
EDDVI    015302            2739    3035#
EDEOP    015420            2744    3035#
EDMOS    015434            2741    3035#
EDRXC    015207            2737    3035#
EDRXQ    015162            2736    3035#
EDTXC    015133            2735    3035#
```

```
EDTXQ   015107            2734    3035#
EFM11   014707            3035#   3071
EFM13   015036            3035#   3085
EFM2    014612            3035#   3057
EF.CON= 000036 G          2167#   4505
EF.NEW= 000035 G          2168#   4511
EF.PWR= 000034 G          2169#
EF.RES= 000037 G          2166#   4500
EF.STA= 000040 G          2165#   4495
EMSG0   002215            2411    2435#
EMSG1   002216            2412    2437#
EMSG2   002217            2413    2439#
EMSG3   002220            2414    2441#
EMSG4   002320            2415    2455#
EMSG5   002412            2416    2469#
EMSG6   002513            2417    2483#
EMSG8   002643            2419    2497#
ENADD   006512            2625#   3902    5176*
ENDEVT  023006            3654    3664    3668    3809#
ENDHWL  046310            7949    7955    7962#
ENDIT   026120            4509    4617    4620#
ENDPS   033502            5781    5811#
ENDQO = 000017            2317#   2913
ENQ   = 000005            6801#   7163    7216    7798    7817    7829    7839
EOP   = 000024            2262#   3329
ERRCNT  006506            2623#   3339    3342    4696    5388*   5711*   5729*   5742*   5785*   5815    6029*   6067*   6135*
                          6288*   6406*   6428*   6646*   7466*   7484*
ERREXT  044506            7474    7493    7518    7527    7536    7545    7554    7563    7569    7574    7578    7582#
ERRPRC  043732            7158    7172    7178    7191    7227    7284    7325    7373    7381    7395    7454#
ERR1    017464 G          3037#   5734
ERR10   017554 G          3067#   5716
ERR13   017636 G          3101#   6034    6072    6140    6293    6411    6433    6651    7471    7489
ERR14   017670 G          3120#
ERR2    017526 G          3054#   5747
ERR4    017604 G          3081#   5790
ERX   = 000100            2271#   5429    5477    5503    5618    5636    5945    6299
ETX   = 000200            2272#   5452    5503    5619    5635    5893    5911    6305
EVL   = 000004 G          2185#
EVMCTS  016502            2724    3035#
EVMDCD  016512            2726    3035#
EVMDSR  016506            2725    3035#
EVMOCG  016365            3035#   3799
EVMOHD  016410            3035#   3819
EVMOST  016465            3035#   3839
EVMRI   016522            2728    3035#
EVMRTS  016516            2727    3035#
EVMSRD  016526            2729    3035#
EVTADD  007646            2752#   3693*   3698    3740*   3746    3754*   3760    3769*   3775
EVTBCT  007650            2753#   3694*   3697    3741*   3745    3755*   3759    3770*   3774
EVTEND  007554            2704#   3384    3661    4576
EVTF0   015555            3035#   3672
EVTF1   015651            3035#   3686
EVTF2   015700            3035#   3699
EVTF3   015752            3035#   3712    3791
EVTF3C  015764            3035#   3105    3720
EVTF3D  016001            3035#   3125
```

```
EVTF4   016023            3035#  3761
EVTF4A  016125            3035#  3776
EVTF4B  016223            3035#  3747
EVTF5A  016302            3035#  3044
EVTLOG  006652            2702   2703#  3388   3646   3659   4573
EVTLST  007610            2734#  3682
EVTMIN  007642            2750#  3680*  3685
EVTPTR  006650            2702#  3374   3389*  3645   3666   4574*
EVTSEC  007640            2749#  3679*  3684
EVTTCK  007644            2751#  3678*  3683
EVTTMP  007652            2754#  3707*  3711   3742*  3744   3756*  3758   3771*  3773   3788*  3790
EXIT  = 000057            2349#  2802   4689   4907   5092
EXITCL  026316            4690   4698#
E$END = 002100            1959#
E$LOAD= 000035            1959#  2046
FACSIM  023510            4060#  5260
FHDPLX  006574            2662#  4592*  6082   6622   6667   6926   7004   7254   7868
FIR     037254            3525*  3531*  3545*  3570   6875#
FIRST = 004000            2285#  6612   6620   7621   7860
FLAG    006602            2674#  5385*  5429*  5452*  5477*  5503*  5553   5567   5586   5618*  5619*  5625*  5627
                          5635*  5636*  5644*  5893*  5901*  5911*  5945*  6039*  6091*  6105*  6111   6145*  6146*
                          6197   6199*  6202*  6252   6254*  6258*  6296   6299   6302   6305   6308   6363   6398
                          6416*  6436   6438*  6439*  6445*  6496   6501*  6513   6515*  6519*  6554*  6565*  6577
                          6583   6612   6620*  6909   7078*  7779*  7797*  7816*  7828*  7838*  7850*
FMTMSK= 000002            6793#  7562
FORMER= 000021            6786#  7324   7557
F$AU  = 000015            1959#  4741   4752
F$AUTO= 000020            1959#  4669   4673
F$BGN = 000040            1959#  1961   3037   3054   3067   3081   3101   3120   3137   3219   4448   4468   4484
                          4650   4669   4683   4702   4719   4725   4741   4747   4768   4851   4911   6358   6489
                          7614   7899   7918   8003
F$CLEA= 000007            1959#  4683   4709
F$DU  = 000016            1959#  4719   4730
F$END = 000041            1959#  1961   3052   3065   3079   3093   3113   3133   3135   3246   4460   4650   4659
                          4675   4702   4711   4723   4732   4745   4754   4768   4851   4911   6453   6525   7739
                          7899   7901   7965   8003
F$HARD= 000004            1959#  7918   7949   7955   7963
F$HW  = 000013            1959#  2086   2114
F$INIT= 000006            1959#  4484   4657
F$JMP = 000050            1959#  3135   4650   4702   4723   4745   4851   4911
F$MOD = 000000            1959#  1961   8003
F$MSG = 000011            1959#  3037   3050   3054   3063   3067   3077   3081   3091   3101   3111   3120   3131
F$PROT= 000021            1959#  4468   4475
F$PWR = 000017            1959#
F$RPT = 000012            1959#  4448   4458
F$SEG = 000003            1959#
F$SOFT= 000005            1959#
F$SRV = 000010            1959#  3219   3244   6358   6451   6489   6523   7614   7737
F$SUB = 000002            1959#
F$SW  = 000014            1959#
F$TEST= 000001            1959#  4769   7899
GETCL   026764            4847   4858#  4883   4892   4895   4897   4899   4906   4915   4917   4930   4945   4959
                          4971   4987   5001   5374
GETPRM  025714            4515   4581#  4590
GETRCL  020640            3412#  3439   3449   3453
GLINC   044632            7630   7634#
```

```
GLMSG   016671          3035#   6377    7635
GLOBCC  037067          6717#   7209*
GOOD    006554          2646#   3042    5727*
GTRA2   026350          4776#
GTRA3   026412          4779    4789#
GTRA4   026420          4794#
GTRA5   026644          4790    4830#   5582    5824    5844
GTREX   032040          5385#
GTRX2   032142          5401#   5822    5913    5973
GTR9    031736          4854    4902    5362#
GTXRXB  026350          4775#
G$CNTO= 000200          1959#
G$DELM= 000372          1959#
G$DISP= 000003          1959#
G$EXCP= 000400          1959#
G$HILI= 000002          1959#
G$LOLI= 000001          1959#
G$NO  = 000000          1959#   3420    4864    5875
G$OFFS= 000400          1959#   3420    4555    4864    5875    7925    7935    7940    7945    7951    7957
G$OFSI= 000376          1959#   3420    4555    4864    5875    7925    7935    7940    7945    7951    7957
G$PRMA= 000001          1959#   7935    7940
G$PRMD= 000002          1959#   3420    4555    4864    5875    7957
G$PRML= 000000          1959#   7925    7945    7951
G$RADA= 000140          1959#   3420    4864    5875
G$RADB= 000000          1959#
G$RADD= 000040          1959#   4555    7957
G$RADL= 000120          1959#   7925    7945    7951
G$RADO= 000020          1959#   7935    7940
G$XFER= 000004          1959#   7949    7955
G$YES = 000010          1959#   4555    7925    7935    7940    7945    7951    7957
HDMADR  002652          2511#   7778*
HDMC    002654          2514#   6108    6211    6267    6557    6567    7121
HDMCC   002646          2508#   6094*   6106*   6119*   6133    6266*   7464    7852*   7853*
HDMID   002645          2506#   7798*   7817*   7829*   7839*   7851*   7863
HDMNUM  002651          2510#   7810*   7821*   7833*   7843*   7355*
HDMREP  002650          2509#   7809*   7822*   7832*   7842*   7854*
HDMSG   002644          2505#   6090*   6107    6264*   6265    6566
HDMTYP  002646          2507#   7799*   7800*   7807*   7818*   7819*   7820*   7830*   7831*   7840*   7841*
HDPLX = 000010          2378#   7870
HDXMTP  037142          6771#   6925*   6930*   6950    7241    7423    7769    7801    7879
HEADBC= 000001          6780#   7186    7187    7512    7517
HELP  = 000000            1#    1959    1973    2064    2076    2090    2118    2122    2129    2993    3016    3030    3035
                        3146    4450    4454    4456    4476    4486    4653    4671    4685    4705    4721    4726    4743
                        4748    4765    4766    4770    6675    7902    7921    7969    7983    7988    7996
HELPDC= 000000            1#    1975    2099    2117    2245    2258    2274    2360    2374    2955    2994    3009    3035
                        3095    3333    3642    3786    4594    4631    5836    5978    5983    6152    6177    6217    6315
                        6320    6454    7930    7966
HLP   = 000005          2307#   2796    2798    4896    5089
HLPEND  003036          2541#   5087
HLPF    012271          3035#   3504    5082
HLPTAB  003016          2533#   5079
HLP0    012213          3035#   4825
HLP1    012276          2533    3035#
HLP2    012311          2534    3035#
HLP3    012426          2535    3035#
HLP3A   012513          2536    3035#
```

```
HLP4    012540            2537    3035#
HLP4A   012617            2538    3035#
HLP5    012675            2539    3035#
HLP6    012776            2540    3035#
HOE   = 100000 G          2198#
HRDMSG  016717            3035#   6374    7632
HSMSE   002653            2512#
IBE   = 010000 G          2195#
IDU   = 000040 G          2188#
IER   = 020000 G          2196#
IMFLAG  037116            6761#   6909*   6959    6970    7006    7022    7041    7046    7078    7619    7621    7860*
IMPRST  037136            6769#   7759*   7777
ININT = 000001            2267#
INIT1   025372            4488    4493#
INMASK  037077            6732#   7186*   7283*   7394*
INOVR = 002000            2283#   6146    6363    7619
INTPRI  011466            2970#   4612*
INVEC   011462            2968#   4609*   4634    4833    6921
IRXCSR  011474            2981#   6076*   6077    6360*   6365    6367    6383    6403    6426    7616*   7623    7625    7641
IRXDBU  011476            2982#   6385*   6386    6392    6400    6402    6425    7644*   7645    7662    7670
ISR   = 000100 G          2189#
IXE   = 004000 G          2194#
I$AU  = 000041            1959#   4741#   4754#
I$AUTO= 000041            1959#   4669#   4675#
I$CLN = 000041            1959#   4683#   4702    4711#
I$DU  = 000041            1959#   4719#   4732#
I$HRD = 000041            7918#   7965#
I$INIT= 000041            1959#   4484#   4650    4659#
I$MOD = 000041            1959#   1961#   8003#
I$MSG = 000041            1959#   3037#   3052#   3054#   3065#   3067#   3079#   3081#   3093#   3101#   3113#   3120*   3133#
I$PROT= 000040            1959#   4468#
I$PTAB= 000041            1959#
I$PWR = 000041            1959#
I$RPT = 000041            1959#   4448#   4460#
I$SEG = 000041            1959#   4768
I$SETU= 000041            1959#
I$SRV = 000041            1959#   3219#   3246#   6358#   6453#   6489#   6525#   7614#   7739#
I$SUB = 000041            1959#   4768
I$TST = 000041            1959#   4768#   4851    4911    7899#   7901#
J$JMP = 000167            1959#   3135    4723    4745
KEYWD1  003010            2529#   3451    3511*   3516*   3521*   4486*   4689    4894    4896    4898    4900    4903    4907
                          4914    4916    4918    5073*   5076*   5089*   5092*   5095*   5099*   5109    5123    5172*   5182*
                          5185*   5256    5261*
LAST    037252            3526*   3532*   3546*   3575    6874#
LBMASK  037101            6738#   7706*   7718*
LBUFER  037100            6737#   7705*   7715*   7717*
LCLKEN= 000100            2228#   4528
LGDVE   020162            3301#   6028    6066    6134    6287    6405    6427    6645    7465    7483
LIS   = 000006            2217#   5278    5912
LISCK   034060            2683    5934#
LISCKA  034106            5941#   5969    5971
LISMOD= 000032            2328#   2828
LISP    014054            3035#   5936
LMDLOP= 000046            2340#   2941
LNCNT   006500            2620#   3355    3357*   3366*   5391*
LOE   = 040000 G          2197#
```

```
LOGCMD  020302            3323#   5748
LOGCML  020264            3319#   5717
LOGCMP  020246            3315#   5707
LOGDVI  020200            3306#   5398
LOGEOP  020320            3327#   5816
LOGEX   020610            3344    3391#
LOGMSC  020336            3333#   6378    7636
LOGRXC  020152            3298#   5590    5959
LOGRXQ  020134            3293#   5562    5953
LOGS1   020354            3286    3291    3296    3300    3339#
LOGS2   020602            3385    3389#
LOGS3   020406            3304    3313    3318    3322    3326    3330    3336    3348#
LOGS4   020462            3356    3365#
LOGS5   020506            3350    3352    3373#   4697
LOGTXC  020116            3288#   5631    5906
LOGTXQ  020100            3283#   5575    5892
LOGUNT  006560            2652#   4579*   4581*   4582    4586
LOOPS   003120            2572#   4133
LOSTER  037106            6752#   7323*   7650*
LOT   = 000010 G          2186#
LP0     013723            2572    3035#   4132
LP00    013724            3035#   4129
LP1     013733            2573    3035#
LP2     013744            2574    3035#
LP3     013752            2575    3035#
LP4     013765            2576    3035#
LSMASK  037107            6753#   7324*   7651*
L$ACP   002110 G          2053#
L$APT   002036 G          2011#
L$AU    026334 G          2038    4741#
L$AUT   002070 G          2037#
L$AUTO  026236 G          2054    4669#
L$CCP   002106 G          2051#
L$CLEA  026240 G          2052    4683#
L$CO    002032 G          2007#
L$DEPO  002011 G          1989#
L$DESC  011534 G          2044    3021#
L$DESP  002076 G          2043#
L$DEVP  002060 G          2029#
L$DISP  002124 G          2014    2073#
L$DLY   002116 G          2059#
L$DTP   002040 G          2013#
L$DTYP  002034 G          2009#
L$DU    026326 G          2040    4719#
L$DUT   002072 G          2039#
L$DVTY  011524 G          2030    3011#
L$EF    002052 G          2024#
L$ENVI  002044 G          2017#
L$ETP   002102 G          2047#
L$EXP1  002046 G          2019#
L$EXP4  002064 G          2033#
L$EXP5  002066 G          2035#
L$HARD  046230 G          1996    7918    7919#
L$HIME  002120 G          2061#
L$HPCP  002016 G          1995#
L$HPTP  002022 G          1999#
```

```
L$HW    002130 G        2000    2086    2087#
L$ICP   002104 G        2049#
L$INIT  025352 G        2050    4484#
L$LADP  002026 G        2003#
L$LAST  046616 G        2004    8001#
L$LOAD  002100 G        2045#
L$LUN   002074 G        2041#
L$MREV  002050 G        2021#
L$NAME  002000 G        1978#
L$PRIO  002042 G        2015#
L$PROT  025344 G        2056    4468#
L$PRT   002112 G        2055#
L$REPP  002062 G        2031#
L$REV   002010 G        1987#
L$RPT   025336 G        2032    4448#
L$SPC   002056 G        2027#
L$SPCP  002020 G        1997#
L$SPTP  002024 G        2001#
L$STA   002030 G        2005#
L$TEST  002114 G        2057#
L$TIML  002014 G        1993#
L$UNIT  002012 G        1991#   4582
L10000  002150          2086    2114#
L10001  017524          3050#
L10002  017552          3063#
L10003  017602          3077#
L10004  017634          3091#
L10005  017666          3111#
L10006  017724          3131#   3136
L10007  020076          3244#
L10010  025342          4458#
L10012  026234          4651    4657#
L10013  026236          4673#
L10014  026324          4703    4709#
L10015  026332          4724    4730#
L10016  026340          4746    4752#
L10017  046224          4852    4912    7899#
L10020  036226          6451#
L10021  036430          6523#
L10022  045334          7737#
L10023  046310          7918    7964#
L5060   014072          3035#   4556
MAINT = 000144          6800#
MCREP   033456          5779    5785#
MES     037256          3571*   3585    3599    3628    6876#
MESDAT  037260          3572*   3584    3596    3598    3610    3613    3616    3627    6877#
MESLON= 000020          6785#   7548    7707
MGLCNT  011516          2990#   3083    5389*   5778    5791*   6376*   7634*
MHRCNT  011520          2991#   3084    5390*   5780    5792*   6373*   7631*
MLTYP   006572          2659#   3310    4820*   4839    4853*   5164    5272*   5291*   ;7*   5339*   5341*   5343*   5345*
                        5350*   5362    6041    6046    6606
MOBITE  007574          2720#   3836
MOBITS  007560          2714#   3824
MOCHK = 000010          2237#   4166    5322    6361    7122    7617
MODE    006604          2677#   5408
MODES   003102          2564#   4126
```

```
MODLOC= 000003           2222#
MODREM= 000004           2223#
MODS    007556           2708#  3346  6170*  6172*  6370  6379*  6625  7628  7637*
MODTYP  006570           2656#  3309  4819*  4838  5163  5267*  5270*  5275*  5278*  5281*  5284*  5287*  5300
                         5347  5406  5600  5616  5633  5912*  5972*  6051  6954  7250  7425  7762  7881
MOMSGS  007574           2724#  3825
MOP   = 000043           2224#  2337#
MOSC  = 000056           2348#  2885
MO0     013632           2564  3035#
MO1     013642           2565  3035#
MO2     013653           2566  3035#
MO3     013663           2567  3035#
MO4     013672           2568  3035#
MO5     013707           2569  3035#
MO6     013714           2570  3035#
MPPTP   037132           4614*  6665  6767#  6928  6939
MSC   = 000016           2258#  3335
MSCMS   014764           3035#  5789
MSGCC   011502           2984#  6108*  6260*  6267*  6503  6510*  6517*  6567*  7864*
MSGLIM= 000017           2204#  2594  2595  2596  4069  4936  4978
MSGPTR  011500           2983#  6107*  6259*  6265*  6506  6508*  6516*  6566*  7863*
MSGTRN  014373           3035#  4925  4940  4966  4982
MSGTRU  014424           3035#  3941
MSGTYP  006516           2630#  3985  4803*  4815*  5134*  5213*  5237*  5239*  5241*  5243*  5245*  5247*  5250*
                         5404*
MSG0    002214           2411  2424  2434#
MSG0C   002150           2411#
MSG1    002215           2412  2425  2436#
MSG1C   002152           2412#
MSG2    002216           2413  2426  2438#
MSG2C   002154           2413#
MSG3    002217           2414  2427  2440#
MSG3C   002156           2414#
MSG4    002220           2415  2428  2442#
MSG4C   002160           2415#
MSG5    002320           2416  2429  2456#
MSG5C   002162           2416#  4804  4816  5135  5248
MSG6    002412           2417  2430  2470#
MSG6C   002164           2417#  5251
MSG8    002642           2419  2432  2496#
MSG8C   002170           2419#
MTLMSK= 000002           6794#  7553  7706
MYDATA= 001000           6822#  7119  7201  7338  7355  7659  7684
N       037060           6704#  7245*  7344*  7413*
NAK   = 000002           6807#  7221  7818
NAKRX = 000400           6821#  7226  7497
NEW     025706           4514  4579#  4583
NO    = 000036           2332#  2868  5294  5297  5328
NOCLK   014234           3035#  4783
NOD0    007714           2795#
NOD1    007720           2796#
NOD10   007776           2803#
NOD100  010624           2875#
NOD101  010630           2876#
NOD102  010644           2877#
NOD103  010650           2879#
```

```
NOD104  010654          2880#
NOD105  010672          2881#
NOD106  010676          2882#
NOD107  010710          2883#
NOD11   010000          2804#
NOD110  010714          2885#
NOD111  010730          2886#
NOD112  010734          2888#
NOD113  010750          2889#
NOD114  010754          2892#
NOD115  010760          2895#
NOD116  010774          2896#
NOD117  011000          2897#
NOD12   010012          2805#
NOD120  011016          2898#
NOD121  011022          2899#
NOD122  011036          2900#
NOD123  011042          2901#
NOD124  011056          2902#
NOD125  011062          2903#
NOD126  011076          2904#
NOD127  011102          2905#
NOD13   010016          2806#
NOD130  011116          2906#
NOD131  011122          2907#
NOD132  011136          2908#
NOD133  011142          2909#
NOD134  011162          2910#
NOD135  011166          2912#
NOD136  011172          2913#
NOD137  011176          2914#
NOD14   010032          2807#
NOD140  011202          2915#
NOD141  011206          2916#
NOD142  011212          2917#
NOD143  011216          2918#
NOD144  011220          2921#
NOD145  011224          2922#
NOD146  011230          2923#
NOD147  011244          2924#
NOD15   010036          2808#
NOD150  011250          2925#
NOD151  011264          2926#
NOD152  011270          2929#
NOD153  011274          2930#
NOD154  011300          2931#
NOD155  011304          2934#
NOD156  011310          2937#
NOD157  011332          2938#
NOD16   010052          2809#
NOD160  011336          2939#
NOD161  011352          2940#
NOD162  011356          2941#
NOD163  011400          2942#
NOD164  011404          2943#
NOD165  011426          2944#
```

```
NOD166  011432            2947#
NOD167  011436            2948#
NOD17   010056            2810#
NOD170  011442            2949#
NOD171  011446            2954#
NOD172  021032            3460#
NOD173  021036            3461#
NOD174  021042            3462#
NOD175  021044            3463#
NOD176  021060            3464#
NOD177  021062            3465#
NOD2    007724            2797#
NOD20   010062            2811#
NOD200  021076            3466#
NOD201  021100            3467#
NOD202  021112            3468#
NOD203  021114            3469#
NOD204  021134            3470#
NOD205  021140            3471#
NOD206  021144            3472#
NOD207  021160            3473#
NOD21   010074            2812#
NOD210  021162            3474#
NOD211  021176            3475#
NOD212  021200            3476#
NOD213  021216            3477#
NOD214  021222            3478#
NOD215  021226            3479#
NOD216  021230            3480#
NOD217  021232            3481#
NOD22   010100            2813#
NOD23   010112            2814#
NOD24   010116            2815#
NOD25   010120            2819#
NOD26   010124            2820#
NOD27   010140            2821#
NOD3    007726            2798#
NOD30   010144            2822#
NOD31   010162            2823#
NOD32   010166            2824#
NOD33   010204            2825#
NOD34   010210            2826#
NOD35   010226            2827#
NOD36   010232            2828#
NOD37   010250            2829#
NOD4    007742            2799#
NOD40   010254            2830#
NOD41   010300            2831#
NOD42   010304            2832#
NOD43   010310            2833#
NOD44   010326            2834#
NOD45   010332            2835#
NOD46   010344            2836#
NOD47   010350            2840#
NOD5    007744            2800#
NOD50   010354            2841#
```

```
NOD51   010376            2842#
NOD52   010400            2843#
NOD53   010424            2844#
NOD54   010426            2849#
NOD55   010432            2850#
NOD56   010450            2851#
NOD57   010454            2852#
NOD6    007760            2801#
NOD60   010474            2853#
NOD61   010500            2856#
NOD62   010504            2857#
NOD63   010510            2858#
NOD64   010514            2859#
NOD65   010520            2860#
NOD66   010524            2861#
NOD67   010530            2862#
NOD7    007762            2802#
NOD70   010534            2863#
NOD71   010540            2866#
NOD72   010544            2867#
NOD73   010550            2868#
NOD74   010562            2869#
NOD75   010566            2870#
NOD76   010602            2871#
NOD77   010606            2874#
NONE  = 000000            2219#
NOTNUF= 000050            2342#   2806    2809    2810    2820    2859    2861    2867    2922
NULEVT  015515            3035#   3649
NULL  = 000000            2302#
NUM   = 000014            2314#   2930
N10$    007720            2795#
N100$   010350            2809    2812    2839#
N102$   010354            2840#
N104$   010400            2841    2842#
N110$   010426            2814    2848#
N111$   010432            2849#
N112$   010454            2850    2851#
N114$   010544            2866#
N115$   010540            2823    2825    2827    2829    2831    2834    2836    2865#   2871    2875    2877    2881    2886
                          2938    2940    2942    2944    2949
N116$   010562            2868#
N117$   010606            2869    2870    2873#
N118$   010630            2874    2875#
N119$   010650            2876    2878#
N120$   010754            2851    2853    2891#
N121$   011220            2896    2898    2900    2902    2904    2906    2908    2914    2920#   2931
N122$   011250            2923    2924#
N123$   011224            2921#
N125$   011446            2859    2861    2863    2867    2910    2922    2953#
N126$   011270            2924    2926    2928#
N130$   010676            2880    2881#
N131$   010734            2885    2887#
N132$   010714            2879    2884#
N140$   011304            2889    2933#
N141$   011310            2936#
N142$   011336            2937    2938#
```

```
N143$   011356          2939    2940#
N144$   011404          2941    2942#
N150$   011432          2883    2946#
N20$    010056          2808    2809#
N25$    010100          2811    2812#
N30$    010116          2810    2813    2814#   2819    2820    2821    2832    2835    2840    2843    2849    2852    2858
                        2860    2862    2882    2888    2892    2912    2925    2929    2930    2934    2943    2947    2948
N40$    010036          2806    2807#
N42$    007726          2796    2797#
N43$    007744          2798    2799#
N44$    007762          2800    2801#
N45$    010000          2802    2803#
N46$    010016          2804    2805#
N50$    010500          2807    2855#
N51$    010504          2856#
N52$    010510          2857#
N60$    010760          2894#
N61$    011000          2895    2896#
N62$    011022          2897    2898#
N63$    011042          2899    2900#
N64$    011062          2901    2902#
N65$    011102          2903    2904#
N66$    011122          2905    2906#
N67$    011142          2907    2908#
N68$    011166          2909    2911#
N70$    011172          2912#   2917
N71$    011202          2913    2914#
N72$    011206          2915#
N73$    011216          2916    2917#
N80$    010120          2805    2818#
N81$    010124          2819#
N82$    010166          2822    2823#
N83$    010210          2824    2825#
N84$    010232          2826    2827#
N85$    010254          2828    2829#
N86$    010304          2830    2831#
N87$    010332          2833    2834#
OFSET   006532          2637#   3043    5720*   5737*
OPBFPT  002514          2489#   5963
OPBUF   002520          2431    2490#   5223    5867    5874    5882    5888    5889    5907    5909    5941    5942    5968
                        5970
OPCNT   002166          2418#   5221*   5886*   5890    5891
OPEND   002642          2491#   5869
OPRAM   014065          3035#   5876
OPRMSG= 000015          2315#   2912
OPVAR   006502          2621#   4694    5386*   5813    6141*   7472*
OTINT = 000002          2268#
OUTMAS  037075          6726#   7517*   7526*
OUTVEC  011464          2969#   4610*   4611*   4641
O$APTS= 000000          1959#   2005
O$AU  = 000001          1959#   1972#   2037
O$BGNR= 000001          1959#   1972#   2031
O$BGNS= 000000          1959#   1997
O$DU  = 000001          1959#   1972#   2039
O$ERRT= 000000          1959#   2047
O$GNSW= 000000          1959#   2001
```

```
O$POIN= 000001            1959#   1972#   2063
O$SETU= 000000            1959#   1991    7999
PAD   = 001000            2281#   6105    6254    6496    6519    6565    7779
PARAM   006576            2663#   3312    3351    3561    4822*   4837*   4841    5166    5299*   5299*   5309*   5327*   5330*
                          5364    5449*   5563    5577    5580    5591    5596    5645    5689    5866*   5897    5934*   5949
                          6084    6361    6914    6944    7122    7223    7235    7237*   7265    7291    7298*   7317*   7401
                          7408*   7456    7617    7760    7805
PARCSR  011452            2962#   4600*   4601*   6079*
PAS   = 000002            2213#   5270    5300    5616    5633
PASC  = 000042            2336#   2948
PASMOD= 000030            2326#   2824
PCADD   006562            2653#
PCK     014014            3035#   4151
PCLKCT= 001600            2230#   4539
PCLKEN= 000111            2229#   4541
PCPM    014655            3035#
PEC     014024            3035#   4155
PLCK    032302            2679    5473#
PLCK2   032302            5474#
PLCK3   032316            5476#
PMS     014033            3035#   4165
PNCK    014012            3035#   4154
PNEC    014022            3035#   4158
PNMS    014031            3035#   4168
PNPR    014041            3035#   4163
PNST    014001            3035#   4150
PNT   = 001000 G          2192#
PPR     014043            3035#   4160
PRFLAG  037140            6770#   7118*   7165*   7668    7676*   7691    7696*   7697*   7729*
PRI   = 002000 G          2193#
PRIBB   021670            3594#   6860    6861    6862    6863    6864    6865
PRIBS   021726            3607#   6866    6867    6868    6869    6870    6871
PRINEX  045332            7643    7667    7679    7726    7735#
PRIN1   044644            7633    7636#
PRIN2   044572            7627#
PRIN21  044656            7618    7620    7622    7624    7641#
PRIN4   044740            7646    7659#
PRIN5   045046            7671    7681#
PRIN6   045022            7669    7676#
PRIN7   045304            7692    7721    7729#
PRIN8   045320            7653    7673    7709    7732#
PRIW    021642            3583#   6857    6858    6859    6872
PRI00 = 000000 G          2181#   4647
PRI01 = 000040 G          2180#
PRI02 = 000100 G          2179#
PRI03 = 000140 G          2178#
PRI04 = 000200 G          2177#
PRI05 = 000240 G          2176#   4632    4639    4831    6919
PRI06 = 000300 G          2175#
PRI07 = 000340 G          2016    2174#   4687
PRNT  = 000055            2347#   2800    4898    5095
PROEND  037114            6760#   7099
PROFDX  041526            7005    7041#
PROHDX  041334            6951    7003#
PROINT  041734            6938    7095#
PRORUN= 000100            2240#   4837    6914    6944    7223    7235    7237    7265    7291    7317    7401    7408    7456
```

```
                    7760  7805
PROTEX  041674      6972  6978  6998  7024  7037  7072#
PROTO = 000041      2335# 2880
PROTOB= 000040      2239# 3561  4161  5325  5563  5577  5897  5949  6084
PROTOC  040706      5579  5902* 5956* 6909#
PRRXI   044510 G    6920  7614#
PRSTAT  037052      3572  6688# 6937* 6941* 6961* 6984* 7013* 7031* 7064* 7097  7116* 7119* 7168
                    7176  7183  7185* 7201* 7202  7207  7226* 7228* 7270* 7278* 7293* 7315* 7322*
                    7338* 7355* 7356  7362* 7367  7377  7386  7390* 7418* 7497  7568  7649* 7659
                    7672* 7681* 7684* 7695* 7708* 7720* 7730* 7759  7777* 7782  7784  7786  7788
                    7790  7811* 7823* 7834* 7845*
PRTEMP  037144      6772#
PRUN    037130      6766# 6935* 6946* 7421
PSCNT   006504      2622# 4695  5387* 5811* 5814
PST     014003      3035# 4147
PTPMLP  046446      7952  7966#
PTREND  006434      2597#
PTRTAB  006150      2594# 4072  4077  4797  4934  5120  5127  5378
PTR13   006244      2595# 4801  4976  5104  5114  5379
PTR23   006340      2596# 5380
P$ACT   003136      2582# 3428* 4249  4872*
P$BUFA  003132      2580# 3426* 4204  4870*
P$CNT   003140      2583# 4346* 4353  4415* 4422* 4427
P$EXIT  024364      4207  4242# 4266  4271
P$GDBD  003147      2587# 3412* 3431  3555* 4264* 4269* 4277  4328* 4371* 4858* 4875  5205* 5258*
                    5308* 5349*
P$NNUF  003146      2586# 3413* 3441  3497* 3538* 4859* 4884  5070* 5100* 5108* 5122* 5177* 5264*
                    5271* 5290* 5357*
P$NUM   003142      2584# 3539  3541  3543  3545  3546  4341* 4344* 4346  4347* 4349* 4353* 4355*
                    4361* 5170  5176  5197  5207  5209  5334
P$RADX  003144      2585# 4301* 4304* 4311* 4318  4351  4359
P$TREE  003134      2581# 3427* 4205  4871*
P$TRV   024246      3430  4203# 4874
P$TR5   024256      4206# 4215  4235  4240
QCOPY = 000013      2313# 2925  5192
QRX   = 000004      2269# 5429  5477  5503  5553  5586  5618  5625  5636  5945  6197  6199  6302
                    6445  6554  6583  6970  7022  7046  7118  7729
QTX   = 000010      2270# 5452  5503  5567  5619  5627  5635  5644  5893  5911  6105  6111  6252
                    6254  6296  6308  6501  6565  6577  6959  7006  7041
QUALFG  003012      2531# 3429* 4873* 5189* 5192* 5195  5294* 5297  5310* 5328  5331*
QUALVL  003014      2532# 4957* 4999* 5186* 5209* 5210*
R       037064      6712# 7268  7351* 7352  7354* 7809  7822  7854
RBMASK  037103      6743# 7535* 7553*
RBUFER  037102      6742# 7532* 7534* 7550* 7552*
REANAK  037070      6719# 7187* 7279* 7652* 7707* 7719* 7819
REC   = 000000      2211#
RECMOD= 000031      2327# 2826
REMTMO  037066      6716# 7271* 7273*
REP   = 000003      6808# 7261
REPLOG  022060      3520  3638#
REPMSK= 000004      6791# 7283
REPORT  020612      3400# 4451  5096
REPSEN= 000003      6782# 7279
RERR  = 100000      2394# 6386  7645
RESET = 000400      2384# 6013  6016
RESFLG  006566      2655# 4493* 4517* 4789
```

```
RESTRT  025646          4503    4529    4542    4561    4570#
RHDADR  002664          2523#   7197    7682
RHDMCC  002660          2520#   6099    6101    6123    6125    6132    6441    6442    7463    7687    7698*   7701    7724
                        7725
RHDMID  002657          2518#   7120    7154    7163    7216
RHDMSG  002656          2517#   6210    6556    7117*
RHDNUM  002663          2522#   7268    7352
RHDREP  002662          2521#   7239    7247    7342    7411
RHDTYP  002660          2519#   7221    7233    7261    7289    7313    7511*   7512    7521    7530    7539    7548    7557
RHLPEN  003052          2549#   3509
RHLPTB  003036          2543#   3501
RHLP0   013075          3035#   3406
RHLP1   013133          2543    3035#
RHLP2   013155          2544    3035#
RHLP3   013210          2545    3035#
RHLP4   013241          2546    3035#
RHLP5   013273          2547    3035#
RHLP6   013336          2548    3035#
RI    = 040000          2370#   2718
RINTEN= 000100          2381#   5823    6212    6447    6560    7124    7126    7733
RMDLOP= 000047          2341#   2943
RMMASK  037105          6748#   7544*   7562*
RMSGCC  011512          2988#   6204*   6211*   6396*   6417*   6441*   6557*   7121*   7666*   7677*   7724*
RMSGPT  011510          2987#   6203*   6210*   6391    6393*   6418*   6440*   6556*   7120*   7661    7663*   7678*   7723*
RMSTER  037104          6747#   7541*   7543*   7559*   7561*
RNODE   011522          2992#   4613*   5394    6080    6200    6256
RNODM   046422          7946    7966#
RNOTNF= 000006          2357#   3469    3471    3476
RPASS   006600          2673#   3311    4821*   4840    4848    5101*   5165    5334*   5818    5820*
RPERR = 000004          2355#   3472
RPEXT = 000002          2353#   3451    3465    3516
RPFUL = 000005          2356#   3474
RPHLP = 000001          2352#   3461    3463    3511
RPLOG = 000003          2354#   3467    3521
RPSWO = 000007          2358#   3478
RPT     022126          3656#   3705    3725    3736    3752    3766    3782    3806
RPTAA   022176          3671#
RPTABO  022512          2769    3739#
RPTDCK  022642          2764    3769#
RPTDDE  022566          2767    3754#
RPTDER  022352          2762    3707#
RPTDLE  022642          2766    3768#
RPTDSP  007654          2758#   3691
RPTDVI  022446          2763    3727#
RPTEOP  022512          2768    3740#
RPTIV   013405          3035#   3550
RPTMSB  023016          3704    3797    3805    3818#
RPTMSC  022716          2765    3788#
RPTMSE  023002          3806#
RPTNV   013430          3035#   3564
RPTTXQ  022300          2758    2759    2760    2761    3693#
RPT0    022172          3647    3663    3667    3670#
RPT1    022160          3660    3666#
RTS   = 000004          2369#   2717    6631    6672
RUN   = 000004          2306#   2804    4900    5099
RXACT = 004000          2383#
```

```
RXBUF   004150          2592#  5402
RXC   = 000006          2253#  3299  3349
RXCSR   011450          2961#  4597*  5823*  6017  6026  6055*  6064  6076  6170  6212*  6285  6360  6447*
                        6560*  6631*  6635  6643  6672*  7124*  7126*  7481  7616  7733*
RXD   = 000100          6820#  7202  7356  7367  7730
RXDBUF  011454          2963#  4602*  4603*  6385  7644
RXDONE= 000200          2382#  6383  7641
RXENA = 000020          2379#  5823  6212  6447  6560  7124  7126  7733
RXINEX  036224          6384  6397  6419  6443  6449#
RXIN1   035702          6375  6378#
RXIN2   035630          6369#
RXIN21  035714          6362  6364  6366  6383#
RXIN3   036074          6387  6422#
RXIN4   035742          6391#
RXIN5   036140          6401  6436#
RXIN6   036050          6399  6416#
RXIN7   036210          6434  6437  6445#
RXIN8   036216          6413  6447#
RXM   = 000040          2277#  6091  6199  6202  6436  6438  7118  7165  7691  7696
RXMTOT  006476          2618#  5382*  5428  5501  5693
RXON    042000          6983*  6997*  7010*  7028*  7036*  7061*  7070*  7114#  7148*
RXONEX  042102          7115  7128#
RXONLY  032210          2677  5426#
RXON2   032210          5427#
RXOVMS= 000001          6792#  7544  7651
RXOVRU= 000011          6784#  7539  7652
RXPRC   037120          6762#  6911*  6968  6989  7020  7044  7048*  7051  7252  7299*  7336  7420*  7713
RXPREX  043730          7159  7258  7307  7318  7349  7405  7422  7424  7426  7428  7431#
RXPROT  042104          6943  6963*  6975*  6986*  7015*  7033*  7050*  7066*  7147#  7173  7180  7194  7210
                        7225  7230  7243  7256  7267  7275  7286  7295  7327  7364  7374  7383  7397
                        7429  7773*
RXPTR   006440          2601#  4799*  5380*  5403  5427  5476  5502  5599  5691
RXQ   = 000004          2252#  3295  5901
RXTHER  037072          6722#  7215*  7571  7573*  7577*
RXTXTE  037110          6757#  7505*  7576*
RXWAIT  042110          7011*  7029*  7062*  7151#
R10$    021036          3460#
R11$    021044          3461  3462#
R12$    021062          3463  3464#
R125$   021232          3480#
R13$    021100          3465  3466#
R14$    021114          3467  3468#
R20$    021140          3470#
R21$    021162          3472  3473#
R22$    021200          3474  3475#
R30$    021230          3469  3471  3476  3477  3478  3479#
SACK  = 000004          6817#  7031  7270  7315  7362  7418  7782  7811
SCM     016603          3035#  3316
SCMD    016636          3035#  3324
SCML    016625          3035#  3320
SDATA = 000020          6819#  6961  6984  7013  7064  7228  7790
SDVE    016572          3035#  3302
SDVI    016614          3035#  3307
SELECT  037134          6768#  6933*  7008  7026  7059  7192*  7689*  7771  7781*  7808*
SELTHE  037071          6720#
SEND  = 000020          2389#  6110  6273  6499  6569  7870  7873
```

```
SEOP    016647          3035#   3328
SETET = 000060          2350#   2909    4894    5261
SETEXP= 000010          2310#   2850    4918    5182    5256
SETTRN= 000011          2311#   2852    5185
SHF0    014507          3035#   4140
SHF1    014545          3035#   4177
SHMSG   013516          3035#   5155
SHOW  = 000002          2304#   2811    4916    5076    5109    5123
SHTAB   003072          2558#   5143    5149
SHTEND  003101          2561#   5146
SHTYP0  013552          2551    3035#
SHTYP1  013561          2551    3035#
SHTYP2  013566          2551    3035#
SHTYP3  013573          2551    3035#
SHTYP4  013600          2551    3035#
SHTYP5  013606          2551    3035#
SHTYP6  013613          2551    3035#
SHTYP7  013621          2551    3035#
SHTYTB  003052          2551#   5154
SHWOP   023714          3734    4124#   4842    5167
SIZE  = 000012          2312#   2923    5189    5195
SMSC    016660          3035#   3334
SNAK  = 000010          6818#   7176    7185    7278    7322    7377    7390    7568    7649    7708    7720    7784    7823
SOH   = 000201          6799#   7851
SPARE0  037112          6758#
SPARE1  037113          6759#
SRD   = 004000          2371#   2719
SRXQ    016561          3035#   3294
SSTACK= 002000          6823#   7293    7788    7845
SSTART= 004000          6824#   6941    7786    7834
STACK = 000007          6810#   7313    7840
STADD   006510          2624#   3874    5170*
STAIND  037212          3573    6857#
STALST  037152          3571    6834#
STAPRI  021542          3527    3533    3547    3560#
START   025446          4498    4517#
STATB = 000001          2234#   3351    4148    5316
STATUS= 000016          2316#   2874
STA0A   037262          6834    6883#
STA1A   037311          6835    6883#
STA10A  040007          6842    6883#
STA11A  040071          6843    6883#
STA12A  040166          6844    6883#
STA13A  040262          6845    6883#
STA14A  040360          6846    6883#
STA15A  040456          6847    6883#
STA16A  040544          6848    6883#
STA17A  040631          6849    6883#
STA2A   037347          6836    6883#
STA3A   037403          6837    6883#
STA4A   037475          6838    6883#
STA5A   037561          6839    6883#
STA6A   037641          6840    6883#
STA7A   037725          6841    6883#
STRIP = 000400          2397#   6212    6560    7124    7126
STRT  = 000006          6809#   7289    7830
```

```
STXC    016550  3035#  3289
STXQ    016537  3035#  3284
SVCGBL= 000000  1959#  1978  1987  1989  1991  1993  1995  1997  1999  2001  2003  2005  2007
                2009   2011  2013  2015  2017  2019  2021  2024  2027  2029  2031  2035  2035
                2037   2039  2041  2043  2045  2047  2049  2051  2053  2055  2057  2059  2061
                2073   2087  2088  3011  3021  3037  3054  3067  3081  3101  3120  3219  4448
                4468   4484  4669  4683  4719  4741  6358  6489  7614  7919  8001#  8002
SVCINS= 000001  1959#  1979  1980  1981  1982  1983  1984  1985  1986  1988  1990  1992  1994
                1996   1998  2000  2002  2004  2006  2008  2010  2012  2014  2016  2018  2020
                2022   2023  2025  2026  2028  2030  2032  2034  2036  2038  2040  2042  2044
                2046   2048  2050  2052  2054  2056  2058  2060  2062  2072  2074  2086  3012
                3014   3022  3027  3039  3040  3041  3042  3043  3044  3045  3046  3047  3048
                3051   3056  3057  3058  3059  3060  3061  3064  3069  3070  3071  3072  3073
                3074   3075  3078  3083  3084  3085  3086  3087  3088  3089  3092  3103  3104
                3105   3106  3107  3108  3109  3112  3122  3123  3124  3125  3126  3127  3128
                3129   3132  3135  3136  3245  3360  3361  3362  3363  3364  3368  3369  3370
                3371   3372  3406  3407  3408  3409  3410  3417  3418  3419  3420  3421  3422
                3423   3424  3434  3435  3436  3437  3438  3444  3445  3446  3447  3448  3503
                3504   3505  3506  3507  3508  3550  3551  3552  3553  3554  3564  3565  3566
                3567   3568  3584  3585  3586  3587  3588  3589  3595  3596  3597  3598  3599
                3600   3601  3602  3603  3620  3621  3622  3623  3624  3625  3626  3627  3628
                3629   3630  3631  3632  3649  3650  3651  3652  3653  3672  3673  3674  3675
                3676   3682  3683  3684  3685  3686  3687  3688  3689  3690  3697  3698  3699
                3700   3701  3702  3703  3711  3712  3713  3714  3715  3716  3718  3719  3720
                3721   3722  3723  3724  3744  3745  3746  3747  3748  3749  3750  3751  3758
                3759   3760  3761  3762  3763  3764  3765  3773  3774  3775  3776  3777  3778
                3779   3780  3790  3791  3792  3793  3794  3795  3799  3800  3801  3802  3803
                3819   3820  3821  3822  3823  3839  3840  3841  3842  3843  3877  3878  3879
                3880   3881  3882  3887  3888  3889  3890  3891  3892  3893  3896  3897  3898
                3899   3900  3901  3941  3942  3943  3944  3945  4136  4137  4138  4139  4140
                4141   4142  4143  4144  4172  4173  4174  4175  4176  4177  4178  4179  4180
                4181   4323  4324  4325  4326  4327  4366  4367  4368  4369  4370  4459  4491
                4495   4496  4498  4500  4501  4503  4505  4506  4508  4511  4512  4514  4522
                4523   4524  4526  4532  4533  4534  4536  4545  4547  4552  4553  4554  4555
                4556   4557  4558  4559  4564  4565  4566  4567  4568  4586  4587  4588  4590
                4622   4623  4624  4625  4626  4627  4632  4633  4634  4635  4636  4637  4639
                4640   4641  4642  4643  4644  4647  4648  4650  4651  4658  4674  4687  4688
                4699   4702  4703  4710  4723  4724  4731  4745  4746  4753  4783  4784  4785
                4786   4787  4825  4826  4827  4828  4829  4831  4832  4833  4834  4835  4836
                4845   4847  4851  4852  4861  4862  4863  4864  4865  4866  4867  4868  4878
                4879   4880  4881  4882  4887  4888  4889  4890  4891  4911  4912  4924  4925
                4926   4927  4928  4929  4939  4940  4941  4942  4943  4944  4965  4966  4967
                4968   4969  4970  4981  4982  4983  4984  4985  4986  5081  5082  5083  5084
                5085   5086  5153  5154  5155  5156  5157  5158  5159  5200  5201  5202  5203
                5204   5230  5231  5232  5233  5234  5303  5304  5305  5306  5307  5352  5353
                5354   5355  5356  5369  5370  5371  5372  5373  5713  5714  5715  5716  5731
                5732   5733  5734  5744  5745  5746  5747  5787  5788  5789  5790  5839  5840
                5841   5842  5843  5872  5873  5874  5875  5876  5877  5878  5879  5936  5937
                5938   5939  5940  5963  5964  5965  5966  5967  6020  6031  6032  6033  6034
                6069   6070  6071  6072  6075  6114  6137  6138  6139  6140  6278  6290  6291
                6292   6293  6408  6409  6410  6411  6430  6431  6432  6433  6452  6524  6574
                6617   6634  6648  6649  6650  6651  6919  6920  6921  6922  6923  6924  7153
                7302   7303  7304  7305  7306  7468  7469  7470  7471  7486  7487  7488  7489
                7738   7886  7900  7918  7925  7926  7927  7935  7936  7937  7938  7940  7941
                7942   7943  7945  7946  7947  7949  7951  7952  7953  7955  7957  7958  7959
                7960   7961  7963  7998  7999  8000
```

```
SVCSUB= 000001  1959#
SVCTAG= 000001  1959#  2114  3050  3063  3077  3091  3111  3131  3244  3425  4458  4560  4657
                4673  4709  4730  4752  4869  5880  6451  6523  7737  7899  7964
SVCTST= 000001  1959#  4768
SYN   = 000226  2398#
SYNCC   011504  2985#  6109*  6268*  6491  6494*  6568*  7865*
SYNCW   011506  2986#  5392*  5397*  6493
S$LSYM= 010000  1959#  2115#  3051#  3064#  3078#  3092#  3112#  3132#  3245#  3418  3425  3426#  4459#
                4553  4560  4561#  4658#  4674#  4710#  4731#  4753#  4862  4869  4870#  5873  5880
                5881#  6452#  6524#  7738#  7900#  7965#
S1      025434  4508  4510#
S2      025510  4526  4531#
S3      025560  4536  4544#
S4      025626  4547  4563#
T       037062  6709#  7102*  7239  7245  7246  7248*  7342  7344  7345  7346  7347*  7411  7413
                7414  7415  7416*  7855
TABEX   014315  3035#  4939  4981
TAL   = 000005  2216#  5287  5972
TALCK   033614  2682  5865#  5908  5910
TALMOD= 000035  2331#  2835
TCURAD  006466  2613#  4805*  4808  4950  4955*  5130*
TEMP    006534  2638#  3285*  3290*  3295*  3299*  3303*  3308*  3317*  3321*  3325*  3329*  3335*  3349
                3375  3376*  3377*  3378  3885*  3888  3947*  3948*  3949  3987*  3988*  3994  4126*
                4139  4147*  4150*  4176  4693*  5151*  5153  5214*  5215*  5219  5222  5605*  5606*
                5610  6614*  6615*  6618  7096*  7101
TEMP1   006536  2639#  3284*  3289*  3294*  3302*  3307*  3316*  3320*  3324*  3328*  3334*  3368  3607*
                3612*  3625  4133*  4137  4151*  4154*  4175  6580*  6581*
TEMP2   006540  2640#  3309*  3310*  3381  3608*  3615*  3623  3990*  3992*  3996  4134*  4136  4155*
                4158*  4174  4695*  5557*  5570*  5588*  5594  5608*  5609  5629*  5697*  5814*  5889*
                5904*  5942*  5957*  6025*  6063*  6131*  6284*  6374*  6377*  6404*  6424*  6642*  6669*
                6670  7462*  7480*  7632*  7635*
TEMP3   006542  2641#  3069  3104  3124  3311*  3382  3609*  3618*  3621  4129*  4132*  4138  4696*
                5559*  5572*  5589*  5595  5606  5607  5608  5630*  5699*  5709  5815*  5890*  5905*
                5944*  5958*  6017*  6026*  6064*  6132*  6285*  6369*  6371  6402*  6425*  6643*  7460*
                7463*  7481*  7627*  7629
TEMP4   006544  2642#  3056  3103  3123  3312*  3346*  3383  4160*  4163*  4172  4694*  5706*  5719*
                5724*  5725  5740  5813*  6027*  6065*  6133*  6286*  6370*  6371  6403*  6426*  6644*
                7461*  7464*  7482*  7628*  7629
TEMP5   006546  2643#  4165*  4168*  4173
TEOM  = 001000  2392#  6512
TERR  = 100000  2393#
TIMEOU  037150  6774#  6913*  7477*  7478  7490*
TIMERS  006646  2699#  3237  3241*  6092*  6096  6115  6121  6275*  6279  6575  6932*  6949*  7156
                7170  7204  7358  7371  7454  7473*  7492*
TIMER1  006642  2697#  3231  3233*  4777*  4778  6057*  6058  6621*  6627  6632*  6637
TIMER2  006644  2698#  3234  3236*
TIMMIN  006634  2693#  3228*  3380  4570*
TIMSEC  006636  2694#  3225*  3226  3229*  3379  4571*
TIMTCK  006640  2695#  3222*  3224*  3239  3377  4572*
TINTEN= 000100  2390#  6110  6273  6499  6569  7870  7873
TMESRX  037056  6702#  6977*  6991*  7035*  7053*  7072  7074*
TMESTX  037054  6701#  6966*  6992*  7018*  7069*  7075  7077*
TOINDT  035474  6280  6296#
TOTCC   006530  2636#  3934*  3935  3946*  3948  3950*  4794*  4920*  4921  4954  4961*  4962  4998
TRA   = 000001  2212#  5284
TRAMOD= 000034  2330#  2833
```

```
TRIBN   037065          4616* 4618* 6713# 7096  7101* 7197  7682  7778
TRIBNQ  046504          7958  7966#
TRVACT  024366          4236  4247# 4263  4268  4273  4276  4296  4362  4385  4406  4430
TRVALN  025160          4225  4389#
TRVALP  025114          4224  4375#
TRVBIF  024472          4221  4276#
TRVBR   024462          4220  4273#
TRVBRC  024406          4234  4254# 4274  4279  4298  4372  4387  4408  4434
TRVDEC  024566          4227  4301#
TRVERR  024424          4218  4263#
TRVEXI  024444          4219  4268#
TRVNMA  024606          4302  4305#
TRVNOB  024416          4259# 4280  4297  4363  4386  4407
TRVNUM  024600          4223  4304#
TRVOCT  024600          4226  4303#
TRVSPA  024514          4222  4282#
TRVSTR  025246          4228  4412#
TSOM  = 000400          2391#
TTL   = 000001          2220# 4853  6041
TTLL  = 004000          2387# 6040  6043
TTLLOP= 000044          2338# 2937
TTOTCC  006464          2612# 4096  4795* 4920  4931  4954* 5126*
TURNON  037146          6773# 6934* 7114  7127* 7732*
TXACT = 001000          2385# 6670
TXBUF   003150          2591# 4064  4805  5129
TXC   = 000002          2251# 3290
TXCSR   011456          2964# 4604* 4605* 6013* 6016* 6027  6040* 6043* 6048* 6065  6110* 6273* 6286
                        6499* 6569* 6644  6669  7482  7870* 7873*
TXDBUF  011460          2965# 4606* 4607* 6493* 6498* 6509* 6512*
TXDONE= 000200          2386#
TXINEX  036426          6495  6502  6511  6518  6521#
TXIN1   036254          6492  6496#
TXIN2   036314          6497  6503#
TXIN3   036420          6514  6519#
TXIN4   036356          6504  6512#
TXM   = 000020          2275# 6091  6254  6258  6513  6515  7797  7816  7828  7838  7850
TXMTOT  006462          2611# 4084  4811* 4933* 4935  4956* 5121  5125* 5366  5451  5474  5499
TXONLY  032242          2678  5449#
TXON2   032250          5450#
TXPRC   037122          6763# 6912* 6964  6987  7016  7019* 7043* 7056  7067  7249* 7300* 7339  7348*
                        7417* 7427  7803
TXPREX  046222          7883  7893#
TXPROT  045336          6942  6962* 6985* 7014* 7032* 7065* 7179  7193  7229  7274  7285  7294  7316
                        7326  7363* 7382  7396  7419  7759#
TXPTR   006442          2602# 4077* 4079* 4080  4089* 4091  4798* 4809  4934* 4948* 4949  4953* 5127*
                        5128  5378* 5450  5475  5500
TXQ   = 000000          2250# 3285
TXREAD  037126          6500* 6765# 6910* 6995  7764  7780* 7887
TXTHER  037073          6723# 7244* 7501  7503* 7506*
T$ARGC= 000001          1979# 1980# 1981# 1982# 1983# 1984# 3039# 3048  3056# 3061  3069# 3075  3083#
                        3089  3103# 3109  3122# 3129  3360# 3364  3368# 3372  3406# 3410  3434# 3438
                        3444# 3448  3503# 3508  3550# 3554  3564# 3568  3584# 3589  3595# 3603  3620#
                        3632  3649# 3653  3672# 3676  3682# 3690  3697# 3703  3711# 3716  3718# 3724
                        3744# 3751  3758# 3765  3773# 3780  3790# 3795  3799# 3803  3819# 3823  3839#
                        3843  3877# 3882  3887# 3893  3896# 3901  3941# 3945  4136# 4144  4172# 4181
                        4323# 4327  4366# 4370  4564# 4568  4783# 4787  4825# 4829  4878# 4882  4887#
```

```
                    4891    4924#   4929    4939#   4944    4965#   4970    4981#   4986    5081#   5086    5153#   5159
                    5200#   5204    5230#   5234    5303#   5307    5352#   5356    5369#   5373    5839#   5843    5936#
                    5940    5963#   5967    7302#   7306
T$CODE= 005052      3420#   4555#   4864#   5875#   7925#   7935#   7940#   7945#   7949#   7951#   7955#   7957#
T$ERRN= 000007      1959#   5714#   5732#   5745#   5788#   6032#   6070#   6138#   6291#   6409#   6431#   6649#   7469#
                    7487#
T$EXCP= 000000      3420#   3425    4555#   4560    4864#   4869    5875#   5880    7935#   7939    7940#   7944    7957#
                    7962
T$FLAG= 000040      3135#   3137    4650#   4702#   4723#   4725    4745#   4747    4851#   4911#
T$GMAN= 000000      1959#   3417#   3420    3426#   4552#   4561#   4861#   4864    4870#   5872#   5875    5881#
T$HILI= 000377      3420#   3424    4555#   4559    4864#   4868    5875#   5879    7935#   7938    7940#   7943    7957#
                    7961
T$LAST= 000001      1959#   7999#
T$LOLI= 000001      3420#   3423    4555#   4558    4864#   4867    5875#   5878    7935#   7937    7940#   7942    7957#
                    7960
T$LSYM= 010000      1959#   2115    3051    3064    3078    3092    3112    3132    3245    4459    4658    4674    4710
                    4731    4753    6452    6524    7738    7900    7965
T$LTNO= 000001      8002#
T$NEST= 177777      1959#   1961#   2086#   2114#   3037#   3050#   3054#   3063#   3067#   3077#   3081#   3091#   3101#
                    3111#   3120#   3131#   3219#   3244#   4448#   4458#   4468#   4475#   4484#   4657#   4669#   4673#
                    4683#   4709#   4719#   4730#   4741#   4752#   4769#   6358#   6451#   6489#   6523#   7614#   7737#
                    7899#   7918#   7949    7955    7963#   8003#
T$NS0 = 000000      1961#   8003
T$NS1 = 000004      2086#   2114    3037#   3050    3054#   3063    3067#   3077    3081#   3091    3101#   3111    3120#
                    3131    3219#   3244    4448#   4458    4468#   4475    4484#   4657    4669#   4673    4683#   4709
                    4719#   4730    4741#   4752    4769#   7899    7918#   7949    7955    7963
T$NS2 = 000010      6358#   6451    6489#   6523    7614#   7737
T$PTNU= 000000      1959#
T$SAVL= 177777      1959#
T$SEGL= 177777      1959#
T$SUBN= 000000      1959#   4768#
T$TAGL= 177777      1959#
T$TAGN= 010024      1959#   2086#   3037#   3054#   3067#   3081#   3101#   3120#   3219#   4448#   4468#   4484#   4669#
                    4683#   4719#   4741#   4769#   6358#   6489#   7614#   7918#
T$TEMP= 000000      2074#   2075#   2114#   3050#   3063#   3077#   3091#   3111#   3131#   3135#   3136    3244#   3420#
                    4458#   4475#   4555#   4650#   4651    4657#   4673#   4702#   4703    4709#   4723#   4724    4730#
                    4745#   4746    4752#   4851#   4852    4864#   4911#   4912    5875#   6451#   6523#   7737#   7899#
                    7925#   7935#   7940#   7945#   7951#   7957#   7963#   8003#
T$TEST= 000001      1959#   4768#   8002
T$TSTM= 177777      1959#   3047    3051    3060    3064    3074    3078    3088    3092    3108    3112    3128    3132
                    3363    3371    3409    3417    3437    3447    3507    3553    3567    3588    3602    3631    3652
                    3675    3689    3702    3715    3723    3750    3764    3779    3794    3802    3822    3842    3881
                    3892    3900    3944    4143    4180    4326    4369    4459    4491    4496    4501    4506    4512
                    4523    4533    4545    4552    4567    4587    4626    4636    4643    4648    4650    4658    4674
                    4688    4699    4702    4710    4731    4753    4786    4828    4835    4845    4851    4861    4881
                    4890    4911    4928    4943    4969    4985    5085    5158    5203    5233    5306    5355    5372
                    5713    5731    5744    5787    5842    5872    5939    5966    6020    6031    6069    6075    6114
                    6137    6278    6290    6408    6430    6574    6617    6634    6648    6923    7153    7305    7468
                    7486    7886    7900
T$TSTS= 000001      1959#   4769#
T$$AU = 010016      4741#   4745    4752
T$$AUT= 010013      4669#   4673
T$$CLE= 010014      4683#   4702    4709
T$$DU = 010015      4719#   4723    4730
T$$HAR= 010023      7918#   7964
T$$HW = 010000      2086#   2114
```

```
T$$INI= 010012          4484#   4650    4657
T$$MSG= 010006          3037#   3050    3054#   3063    3067#   3077    3081#   3091    3101#   3111    3120#   3131    3135
T$$PRO= 010011          4468#
T$$RPT= 010010          4448#   4458
T$$SRV= 010022          3219#   3244    6358#   6451    6489#   6523    7614#   7737
T$$TES= 010017          4769#   4851    4911    7899
T1      026342 G        2074    4768#
UAM   = 000200 G        2190#
UPTABL  032610          5591#
UPTA1   032676          5597    5604#
UPTA3   032674          5601    5603#
UPTA4   032634          5592    5596#
UPTEX   032746          5603    5615#
VECTOR  046367          7941    7966#
X       037063          6710#   7246*   7345*   7414*
X$    = 000220          1962#   2795#   2796#   2797#   2798#   2799#   2800#   2801#   2802#   2803#   2804#   2805#   2806#
                        2807#   2808#   2809#   2810#   2811#   2812#   2813#   2814#   2815#   2819#   2820#   2821#   2822#
                        2823#   2824#   2825#   2826#   2827#   2828#   2829#   2830#   2831#   2832#   2833#   2834#   2835#
                        2836#   2840#   2841#   2842#   2843#   2844#   2849#   2850#   2851#   2852#   2853#   2856#   2857#
                        2858#   2859#   2860#   2861#   2862#   2863#   2866#   2867#   2868#   2869#   2870#   2871#   2874#
                        2875#   2876#   2877#   2879#   2880#   2881#   2882#   2883#   2885#   2886#   2888#   2889#   2892#
                        2895#   2896#   2897#   2898#   2899#   2900#   2901#   2902#   2903#   2904#   2905#   2906#   2907#
                        2908#   2909#   2910#   2912#   2913#   2914#   2915#   2916#   2917#   2918#   2921#   2922#   2923#
                        2924#   2925#   2926#   2929#   2930#   2931#   2934#   2937#   2938#   2939#   2940#   2941#   2942#
                        2943#   2944#   2947#   2948#   2949#   2954#   3460#   3461#   3462#   3463#   3464#   3465#   3466#
                        3467#   3468#   3469#   3470#   3471#   3472#   3473#   3474#   3475#   3476#   3477#   3478#   3479#
                        3480#   3481#
X$ALWA= 000000          1959#
X$FALS= 000040          1959#   7955
X$OFFS= 000400          1959#   7949    7955
X$TRUE= 000020          1959#   7949
$PATCH  046532          7993#
.     = 046616          1959#   2484#   2490#   2498#   2513#   2524#   2528#   2562#   2591#   2592#   2593#   2594#   2595#
                        2596#   2599#   2703#   2704#   2798#   2802#   2806#   2813#   2820#   2822#   2828#   2830#   2841#
                        2843#   2850#   2852#   2868#   2870#   2874#   2876#   2888#   2895#   2897#   2899#   2901#   2903#
                        2909#   2923#   2925#   2941#   3014#   3035#   3136    3463#   3465#   3469#   3474#   3476#   4651
                        4703    4724    4746    4852    4912    7309#   7949    7955    7994#
```

```
BCOMPL      1#  1959#  4497  450?  4513  4846
BERROR      1#  1959#
BGNAU       1#  1959#  4740
BGNAUT      1#  1959#  4668
BGNCLN      1#  1959#  4682
BGNDU       1#  1959#  4718
BGNHRD      1#  1959#  7917
BGNHW       1#  1959#  2085
BGNINI      1#  1959#  4483
BGNMOD      1#  1959#  1960
BGNMSG      1#  1959#  3036  3053  3066  3080  3100  3119
BGNPRO      1#  1959#  4467
BGNPTA      1#  1959#
BGNRPT      1#  1959#  4447
BGNSEG      1#  1959#
BGNSET      1#  1959#
BGNSFT      1#  1959#
BGNSRV      1#  1959#  3218  6357  6488  7613
BGNSUB      1#  1959#
BGNSW       1#  1959#
BGNTST      1#  1959#  4767
BNCOMP      1#  1959#  4507  4525  4535  4546  4589
BNERRO      1#  1959#
BREAK       1#  1959#  6019  6074  6113  6277  6573  6616  6633  7152  7885
BRESET      1#  1959#  4698
CKLOOP      1#  1959#
CLI      1963#  2794   2795  2796  2797  2798  2799  2800  2801  2802  2803  2804  2805  2806  2807
         2808   2809   2810  2811  2812  2813  2814  2818  2819  2820  2821  2822  2823  2824  2825
         2826   2827   2828  2829  2830  2831  2832  2833  2834  2835  2839  2840  2841  2842  2843
         2848   2849   2850  2851  2852  2855  2856  2857  2858  2859  2860  2861  2862  2865  2866
         2867   2868   2869  2870  2873  2874  2875  2876  2878  2879  2880  2881  2882  2884  2885
         2887   2888   2891  2894  2895  2896  2897  2898  2899  2900  2901  2902  2903  2904  2905
         2906   2907   2908  2909  2911  2912  2913  2914  2915  2916  2917  2920  2921  2922  2923
         2924   2925   2928  2929  2930  2933  2936  2937  2938  2939  2940  2941  2942  2943  2946
         2947   2948   2953  3459  3460  3461  3462  3463  3464  3465  3466  3467  3468  3469  3470
         3471   3472   3473  3474  3475  3476  3477  3478  3479  3480
CLOCK       1#  1959#  4521  4531
CLOSE       1#  1959#
CLRVEC      1#  1959#
COMMEN      1#  1959#
DELAY       1#  1959#
DESCRI      1#  1959#  3020
DEVTYP      1#  1959#  3010
DISPAT      1#  1959#  2071
DISPLA      1#  1959#
DOCLN       1#  1959#  4490
DODU        1#  1959#
DORPT       1#  1959#
ENDAU       1#  1959#  4751
ENDAUT      1#  1959#  4672
ENDCLN      1#  1959#  4708
ENDCOM      1#  1959#
ENDDU       1#  1959#  4729
ENDHRD      1#  1959#  7962
ENDHW       1#  1959#  2113
ENDINI      1#  1959#  4656
```

```
ENDMOD      1#  1959#  8002
ENDMSG      1#  1959#  3049  3062  3076  3090  3110  3130
ENDPRO      1#  1959#  4474
ENDPTA      1#  1959#
ENDRPT      1#  1959#  4457
ENDSEG      1#  1959#
ENDSET      1#  1959#
ENDSFT      1#  1959#
ENDSRV      1#  1959#  3243  6450  6522  7736
ENDSUB      1#  1959#
ENDSW       1#  1959#
ENDTST      1#  1959#  7898
EQUALS      1#  1959#  2130
ERRDF       1#  1959#
ERRHRD      1#  1959#
ERROR       1#  1959#
ERRSF       1#  1959#
ERRSOF      1#  1959#  5712  5730  5743  5786  6030  6068  6136  6289  6407  6429  6647  7467  7485
ERRTBL      1#  1959#
ESCAPE      1#  1959#
EXIT        1#  1959#  3134  4649  4701  4722  4744  4850  4910
FEQUAL      1#  1959#
GETBYT      1#  1959#
GETPRI      1#  1959#
GETWOR      1#  1959#
GMANIA      1#  1959#
GMANID      1#  1959#  3416  4551  4860  5871
GMANIL      1#  1959#
GPHARD      1#  1959#  4585
GPRMA       1#  1959#  7934  7939
GPRMD       1#  1959#  3417#  3420  4552#  4555  4861#  4864  5872#  5875  7956
GPRML       1#  1959#  7924  7944  7950
HEADER      1#  1959#  1977
INLOOP      1#  1959#
IOSETU      1#  1959#
IOSTAR      1#  1959#
KT11        1#  1959#
LASTAD      1#  1959#  7997
MANUAL      1#  1959#  4844
MEMORY      1#  1959#
MSBYTE      1#  1959#  1978#  1984  1985  1986
MSCHEC      1#  1959#  3135#  4650#  4702#  4723#  4745#  4851#  4911#
MSCNTO      1#  1959#  3420#  4555#  4864#  5875#  7925#  7935#  7940#  7945#  7951#  7957#
MSCOUN      1#  1959#  3039#  3056#  3069#  3083#  3103#  3122#  3360#  3368#  3406#  3434#  3444#  3503#  3550#
         3564#  3584#  3595#  3620#  3649#  3672#  3682#  3697#  3711#  3718#  3744#  3758#  3773#  3790#  3799#
         3819#  3839#  3877#  3887#  3896#  3941#  4136#  4172#  4323#  4366#  4564#  4783#  4825#  4878#  4887#
         4924#  4939#  4965#  4981#  5081#  5153#  5200#  5230#  5303#  5352#  5369#  5839#  5936#  5963#  7302#
MSDATA      1#  1959#  1978#  1987  1989  1991  1993  1995  1997  1999  2001  2003  2005  2007  2009
         2011  2013  2015  2017#  2019  2021  2024  2027  2029  2031  2033  2035  2037  2039  2041
         2043  2045  2047  2049  2051  2053  2055  2057  2059  2061  3011#  3021#
MSDECR      1#  1959#  2114#  3050#  3063#  3077#  3091#  3111#  3131#  3244#  4458#  4475#  4657#  4673#  4709#
         4730#  4752#  6451#  6523#  7737#  7899#  7963#  8003#
MSDEFA      1#  1959#  3420#  4555#  4864#  5875#  7925#  7935#  7940#  7945#  7951#  7957#
MSENDE      1#  1959#  2114#  3050#  3063#  3077#  3091#  3111#  3131#  3244#  4458#  4657#  4673#  4709#  4730#
         4752#  6451#  6523#  7737#  7899#  7963#  8003#
MSERRI      1#  1959#  5713#  5731#  5744#  5787#  6031#  6069#  6137#  6290#  6408#  6430#  6648#  7468#  7486#
```

```
MSESCA      1#   1959#
MSESCS      1#   1959#
MSEXCP      1#   1959#  3420#  4555#  4864#  5875#  7935#  7940#  7957#
MSEXIT      1#   1959#  3135#  4650#  4651   4702#  4703   4723#  4745#  4851#  4852   4911#  4912
MSEXSE      1#   1959#  3135#  4650#  4702#  4723#  4745#  4851#  4911#
MSEXTJ      1#   1959#  3135#  3136   4650#  4702#  4723#  4724   4745#  4746   4851#  4911#
MSGEN       1#   1959#  1978#  1987#  1989#  1991#  1993#  1995#  1997#  1999#  2001#  2003#  2005#  2007#  2009#
         2011#  2013#  2015#  2017#  2019#  2021#  2024#  2027#  2029#  2031#  2033#  2035#  2037#  2039#  2041#
         2043#  2045#  2047#  2049#  2051#  2053#  2055#  2057#  2059#  2061#  2073#  2087#  2088#  2114#  3011#
         3021#  3037#  3050#  3054#  3063#  3067#  3077#  3081#  3091#  3101#  3111#  3120#  3131#  3219#  3244#
         3425#  4448#  4458#  4468#  4484#  4560#  4657#  4669#  4673#  4683#  4709#  4719#  4730#  4741#  4752#
         4768#  4869#  5880#  6358#  6451#  6489#  6523#  7614#  7737#  7899#  7919#  7964#  8001#
MSGENB      1#   1959#  3417#  3418   4552#  4553   4861#  4862   5872#  5873
MSGETS      1#   1959#  2114#  3050#  3063#  3077#  3091#  3111#  3131#  3244#  4458#  4475#  4657#  4673#  4709#
         4730#  4752#  6451#  6523#  7737#  7899#  7949#  7955#  7963#  8003#
MSGETT      1#   1959#  3135#  4650#  4702#  4723#  4745#  4851#  4911#  7949#  7955#
MSGNGB      1#   1959#  1961#  1978#  1987#  1989#  1991#  1993#  1995#  1997#  1999#  2001#  2003#  2005#  2007#
         2009#  2011#  2013#  2015#  2017#  2019#  2021#  2024#  2027#  2029#  2031#  2033#  2035#  2037#  2039#
         2041#  2043#  2045#  2047#  2049#  2051#  2053#  2055#  2057#  2059#  2061#  2072#  2073   2086#  2087
         2088   3011#  3021#  3037#  3054#  3067#  3081#  3101#  3120#  3219#  4448#  4468#  4484#  4669#  4683#
         4719#  4741#  6358#  6489#  7614#  7918#  7919   7998#  8001
MSGNIN      1#   1959#  1978#  1979   1980   1981   1982   1983   1984#  1985#  1986#  1987#  1988   1989#  1990
         1991#  1992   1993#  1994   1995#  1996   1997#  1998   1999#  2000   2001#  2002   2003#  2004   2005#
         2006   2007#  2008   2009#  2010   2011#  2012   2013#  2014   2015#  2016   2017#  2018   2019#  2020
         2021#  2022   2023   2024#  2025   2026#  2027#  2028   2029#  2030   2031#  2032   2033#  2034   2035#
         2036   2037#  2038   2039#  2040   2041#  2042   2043#  2044   2045#  2046   2047#  2048   2049#  2050
         2051#  2052   2053#  2054   2055#  2056   2057#  2058   2059#  2060   2061#  2062   2072#  2074#  2086#
         3011#  3012   3014   3021#  3022   3027   3039#  3040   3041#  3042   3043#  3044#  3045#  3046   3047#
         3048   3051#  3056#  3057#  3058#  3059   3060#  3061   3064#  3069#  3070#  3071#  3072#  3073   3074#
         3075   3078#  3083#  3084#  3085#  3086#  3087   3088#  3089   3092#  3103#  3104#  3105#  3106#  3107
         3108#  3109   3112#  3122#  3123#  3124#  3125#  3126#  3127   3128#  3129   3132#  3135#  3136#  3244#
         3245   3360#  3361#  3362   3363#  3364   3368#  3369#  3370   3371#  3372   3406#  3407#  3408   3409#
         3410   3417#  3418#  3419#  3420#  3421   3422   3423   3424   3434#  3435#  3436   3437#  3438   3444#
         3445#  3446   3447#  3448   3503#  3504#  3505#  3506   3507#  3508   3550#  3551#  3552   3553#  3554
         3564#  3565#  3566   3567#  3568   3584#  3585#  3586#  3587   3588#  3589   3595#  3596   3597#  3598
         3599#  3600#  3601   3602#  3603   3620#  3621   3622#  3623   3624#  3625   3626#  3627   3628#  3629#
         3630   3631#  3632   3649#  3650#  3651   3652#  3653   3672#  3673#  3674   3675#  3676   3682#  3683#
         3684#  3685#  3686#  3687#  3688   3689#  3690   3697#  3698#  3699#  3700#  3701   3702#  3703   3711#
         3712#  3713#  3714   3715#  3716   3718#  3719#  3720#  3721#  3722   3723#  3724   3744#  3745#  3746#
         3747#  3748#  3749   3750#  3751   3758#  3759#  3760#  3761#  3762#  3763   3764#  3765   3773#  3774#
         3775#  3776#  3777#  3778   3779#  3780   3790#  3791#  3792#  3793   3794#  3795   3799#  3800#  3801
         3802#  3803   3819#  3820#  3821   3822#  3823   3839#  3840#  3841   3842#  3843   3877#  3878#  3879#
         3880   3881#  3882   3887#  3888   3889#  3890#  3891   3892#  3893   3896#  3897#  3898#  3899   3900#
         3901   3941#  3942#  3943   3944#  3945   4136#  4137#  4138#  4139#  4140#  4141#  4142   4143#  4144
         4172#  4173#  4174#  4175#  4176#  4177#  4178#  4179   4180#  4181   4323#  4324#  4325   4326#  4327
         4366#  4367#  4368   4369#  4370   4459#  4491#  4495#  4496#  4498#  4500#  4501#  4503#  4505#  4506#
         4508#  4511#  4512#  4514#  4522#  4523#  4524#  4526#  4532#  4533#  4534#  4536#  4545#  4547#  4552#
         4553#  4554#  4555#  4556   4557   4558   4559   4564#  4565#  4566   4567#  4568   4586#  4587#  4588#
         4590#  4622#  4623#  4624#  4625#  4626#  4627   4632#  4633#  4634#  4635#  4636#  4637   4639#  4640#
         4641#  4642#  4643#  4644   4647#  4648#  4650#  4651#  4658#  4674#  4687#  4688#  4699#  4702#  4703#
         4710#  4723#  4724#  4731#  4745#  4746#  4753#  4783#  4784#  4785   4786#  4787   4825#  4826#  4827
         4828#  4829   4831#  4832#  4833#  4834#  4835#  4836   4845#  4847#  4851#  4852#  4861#  4862#  4863#
         4864#  4865   4866   4867   4868   4878#  4879#  4880   4881#  4882   4887#  4888#  4889   4890#  4891
         4911#  4912#  4924#  4925#  4926#  4927   4928#  4929   4939#  4940#  4941#  4942   4943#  4944   4965#
         4966#  4967#  4968   4969#  4970   4981#  4982#  4983#  4984   4985#  4986   5081#  5082#  5083#  5084
         5085#  5086   5153#  5154#  5155#  5156#  5157   5158#  5159   5200#  5201#  5202   5203#  5204   5230#
```

```
                5231#   5232    5233#   5234    5303#   5304#   5305    5306#   5307    5352#   5353#   5354    5355#   5356    5369#
                5370#   5371    5372#   5373    5713#   5714#   5715#   5716#   5731#   5732#   5733#   5734#   5744#   5745#   5746#
                5747#   5787#   5788#   5789#   5790#   5839#   5840#   5841    5842#   5843    5872#   5873#   5874#   5875#   5876
                5877    5878    5879    5936#   5937#   5938    5939#   5940    5963#   5964#   5965    5966#   5967    6020#   6031#
                6032#   6033#   6034#   6069#   6070#   6071#   6072#   6075#   6114#   6137#   6138#   6139#   6140#   6278#   6290#
                6291#   6292#   6293#   6408#   6409#   6410#   6411#   6430#   6431#   6432#   6433#   6451#   6452    6523#   6524
                6574#   6617#   6634#   6648#   6649#   6650#   6651#   6919#   6920#   6921#   6922#   6923#   6924    7153#   7302#
                7303#   7304    7305#   7306    7468#   7469#   7470#   7471#   7486#   7487#   7488#   7489#   7737#   7738    7886#
                7900#   7918#   7925#   7926    7927    7935#   7936    7937    7938    7940#   7941    7942    7943    7945#   7946
                7947    7949#   7951#   7952    7953    7955#   7957#   7958    7959    7960    7961    7963#   7998#   7999#   8000#
MSGNLS            1#    1959#   3417#   3425    4552#   4560    4861#   4869    5872#   5880
MSGNSU            1#    1959#
MSGNTA            1#    1959#   2114#   3050#   3063#   3077#   3091#   3111#   3131#   3244#   4458#   4657#   4673#   4709#   4730#
                4752#   6451#   6523#   7737#   7899#   7963#   7964
MSGNTE            1#    1959#   4768#
MSHAPT            1#    1959#   1978#
MSHNAP            1#    1959#   1978#   2017
MSINCR            1#    1959#   1961#   2086#   3037#   3047#   3051#   3054#   3060#   3064#   3067#   3074#   3078#   3081#   3088#
                3092#   3101#   3108#   3112#   3120#   3128#   3132#   3219#   3363#   3371#   3409#   3417#   3426    3437#   3447#
                3507#   3553#   3567#   3588#   3602#   3631#   3652#   3675#   3689#   3702#   3715#   3723#   3750#   3764#   3779#
                3794#   3802#   3822#   3842#   3881#   3892#   3900#   3944#   4143#   4180#   4326#   4369#   4448#   4459#   4468#
                4484#   4491#   4496#   4501#   4506#   4512#   4523#   4533#   4545#   4552#   4561    4567#   4587#   4626#   4636#
                4643#   4648#   4650#   4658#   4669#   4674#   4683#   4688#   4699#   4702#   4710#   4719#   4731#   4741#   4753#
                4768#   4769#   4786#   4828#   4835#   4845#   4851#   4861#   4870    4881#   4890#   4911#   4928#   4943#   4969#
                4985#   5085#   5158#   5203#   5233#   5306#   5355#   5372#   5713#   5731#   5744#   5787#   5842#   5872#   5881
                5939#   5966#   6020#   6031#   6069#   6075#   6114#   6137#   6278#   6290#   6358#   6408#   6430#   6489#   6574#
                6617#   6634#   6648#   6923#   7153#   7305#   7468#   7486#   7614#   7886#   7900#   7918#
MSIOSE            1#    1959#
MSLDRO            1#    1959#   4495#   4500#   4505#   4511#   4522#   4532#   4586#   4647#   4687#
MSMASK            1#    1959#
MSMCHI            1#    1959#
MSMCLO            1#    1959#
MSMSK1            1#    1959#
MSPOP             1#    1959#   2114#   3050#   3063#   3077#   3091#   3111#   3131#   3244#   4458#   4475#   4657#   4673#   4709#
                4730#   4752#   6451#   6523#   7737#   7899#   7963#   8003#
MSPRIN            1#    1959#   3039#   3056#   3069#   3083#   3103#   3122#   3360#   3368#   3406#   3434#   3444#   3503#   3550#
                3564#   3584#   3595#   3620#   3649#   3672#   3682#   3697#   3711#   3718#   3744#   3758#   3773#   3790#   3799#
                3819#   3839#   3877#   3887#   3896#   3941#   4136#   4172#   4323#   4366#   4564#   4783#   4825#   4878#   4887#
                4924#   4939#   4965#   4981#   5081#   5153#   5200#   5230#   5303#   5352#   5369#   5839#   5936#   5963#   7302#
MSPUSH            1#    1959#   1961#   2086#   3037#   3054#   3067#   3081#   3101#   3120#   3219#   4448#   4468#   4484#   4669#
                4683#   4719#   4741#   4768#   4769    6358#   6489#   7614#   7918#
MSPUT             1#    1959#   3039#   3056#   3069#   3083#   3103#   3122#   3360#   3368#   3406#   3434#   3444#   3503#   3550#
                3564#   3584#   3595#   3620#   3649#   3672#   3682#   3697#   3711#   3718#   3744#   3758#   3773#   3790#   3799#
                3819#   3839#   3877#   3887#   3896#   3941#   4136#   4172#   4323#   4366#   4564#   4622#   4632#   4639#   4783#
                4825#   4831#   4878#   4887#   4924#   4939#   4965#   4981#   5081#   5153#   5200#   5230#   5303#   5352#   5369#
                5839#   5936#   5963#   6919#   7302#
MSPUT1            1#    1959#   3039#   3041    3043    3044    3045    3056#   3057    3058    3069#   3070    3071    3072    3083#
                3084    3085    3086    3103#   3104    3105    3106    3122#   3123    3124    3125    3126    3360#   3361    3368#
                3369    3406#   3407    3434#   3435    3444#   3445    3503#   3504    3505    3550#   3551    3564#   3565    3584#
                3585    3586    3595#   3597    3599    3600    3620#   3622    3624    3626    3628    3629    3649#   3650    3672#
                3673    3682#   3683    3684    3685    3686    3687    3697#   3698    3699    3700    3711#   3712    3713    3718#
                3719    3720    3721    3744#   3745    3746    3747    3748    3758#   3759    3760    3761    3762    3773#   3774
                3775    3776    3777    3790#   3791    3792    3799#   3800    3819#   3820    3839#   3840    3877#   3878    3879
                3887#   3889    3890    3896#   3897    3898    3941#   3942    4136#   4137    4138    4139    4140    4141    4172#
                4173    4174    4175    4176    4177    4178    4323#   4324    4366#   4367    4564#   4565    4622#   4623    4624
                4625    4632#   4633    4634    4635    4639#   4640    4641    4642    4783#   4784    4825#   4826    4831#   4832
```

```
                4833    4834    4878#   4879    4887#   4888    4924#   4925    4926    4939#   4940    4941    4965#   4966    4967
                4981#   4982    4983    5081#   5082    5083    5153#   5154    5155    5156    5200#   5201    5230#   5231    5303#
                5304    5352#   5353    5369#   5370    5839#   5840    5936#   5937    5963#   5964    6919#   6920    6921    6922
                7302#   7303
M$RADI            1#    1959#   3420#   4555#   4864#   5875#   7925#   7935#   7940#   7945#   7951#   7957#
M$RBR0            1#    1959#
M$RNR0            1#    1959#   4522#   4524    4532#   4534    4586#   4588
M$SETS            1#    1959#   1961#   2086#   3037#   3054#   3067#   3081#   3101#   3120#   3219#   4448#   4468#   4484#   4669#
                4683#   4719#   4741#   4769#   6358#   6489#   7614#   7918#
M$STAR            1#    1959#
M$SVC             1#    1959#   3039#   3047    3050#   3051    3056#   3060    3063#   3064    3069#   3074    3077#   3078    3083#
                3088    3091#   3092    3103#   3108    3111#   3112    3122#   3128    3131#   3132    3135#   3360#   3363    3368#
                3371    3406#   3409    3417#   3434#   3437    3444#   3447    3503#   3507    3550#   3553    3564#   3567    3584#
                3588    3595#   3602    3620#   3631    3649#   3652    3672#   3675    3682#   3689    3697#   3702    3711#   3715
                3718#   3723    3744#   3750    3758#   3764    3773#   3779    3790#   3794    3799#   3802    3819#   3822    3839#
                3842    3877#   3881    3887#   3892    3896#   3900    3941#   3944    4136#   4143    4172#   4180    4323#   4326
                4366#   4369    4458#   4459    4491#   4495#   4496    4500#   4501    4505#   4506    4511#   4512    4522#   4523
                4532#   4533    4545#   4552#   4564#   4567    4586#   4587    4622#   4626    4632#   4636    4639#   4643    4647#
                4648    4650#   4657#   4658    4673#   4674    4687#   4688    4699#   4702#   4709#   4710    4723#   4730#   4731
                4745#   4752#   4753    4783#   4786    4825#   4828    4831#   4835    4845#   4851#   4861#   4878#   4881    4887#
                4890    4911#   4924#   4928    4939#   4943    4965#   4969    4981#   4985    5081#   5085    5153#   5158    5200#
                5203    5230#   5233    5303#   5306    5352#   5355    5369#   5372    5713    5731    5744    5787    5839#   5842
                5872#   5936#   5939    5963#   5966    6020#   6031    6069    6075#   6114#   6137    6278#   6290    6408    6430
                6574#   6617#   6634#   6648    6919#   6923    7153#   7302#   7305    7468    7486    7886#   7899#   7900
M$TLAB            1#    1959#   3047#   3051#   3060#   3064#   3074#   3078#   3088#   3092#   3108#   3112#   3128#   3132#   3363#
                3371#   3409#   3417#   3437#   3447#   3507#   3553#   3567#   3588#   3602#   3631#   3652#   3675#   3689#   3702#
                3715#   3723#   3750#   3764#   3779#   3794#   3802#   3822#   3842#   3881#   3892#   3900#   3944#   4143#   4180#
                4326#   4369#   4459#   4491#   4496#   4501#   4506#   4512#   4523#   4533#   4545#   4552#   4567#   4587#   4626#
                4636#   4643#   4648#   4650#   4658#   4674#   4688#   4699#   4702#   4710#   4731#   4753#   4786#   4828#   4835#
                4845#   4851#   4861#   4881#   4890#   4911#   4928#   4943#   4969#   4985#   5085#   5158#   5203#   5233#   5306#
                5355#   5372#   5713#   5731#   5744#   5787#   5842#   5872#   5939#   5966#   6020#   6031#   6069#   6075#   6114#
                6137#   6278#   6290#   6408#   6430#   6574#   6617#   6634#   6648#   6923#   7153#   7305#   7468#   7486#   7886#
                7900#
M$TSTL            1#    1959#   3047#   3051#   3060#   3064#   3074#   3078#   3088#   3092#   3108#   3112#   3128#   3132#   3363#
                3371#   3409#   3417#   3437#   3447#   3507#   3553#   3567#   3588#   3602#   3631#   3652#   3675#   3689#   3702#
                3715#   3723#   3750#   3764#   3779#   3794#   3802#   3822#   3842#   3881#   3892#   3900#   3944#   4143#   4180#
                4326#   4369#   4459#   4491#   4496#   4501#   4506#   4512#   4523#   4533#   4545#   4552#   4567#   4587#   4626#
                4636#   4643#   4648#   4650#   4658#   4674#   4688#   4699#   4702#   4710#   4731#   4753#   4786#   4828#   4835#
                4845#   4851#   4861#   4881#   4890#   4911#   4928#   4943#   4969#   4985#   5085#   5158#   5203#   5233#   5306#
                5355#   5372#   5713#   5731#   5744#   5787#   5842#   5872#   5939#   5966#   6020#   6031#   6069#   6075#   6114#
                6137#   6278#   6290#   6408#   6430#   6574#   6617#   6634#   6648#   6923#   7153#   7305#   7468#   7486#   7886#
                7900#
M$WORD            1#    1959#   2017#   2026    2072#   2074    3135#   3417#   3419    3420#   4552#   4554    4555#   4650#   4702#
                4723#   4745#   4851#   4861#   4863    4864#   4911#   5713#   5714    5715    5716    5731#   5732    5733    5734
                5744#   5745    5746    5747    5787#   5788    5789    5790    5872#   5874    5875#   6031#   6032    6033    6034
                6069#   6070    6071    6072    6137#   6138    6139    6140    6290#   6291    6292    6293    6408#   6409    6410
                6411    6430#   6431    6432    6433    6648#   6649    6650    6651    7468#   7469    7470    7471    7486#   7487
                7488    7489    7925#   7935#   7940#   7945#   7949#   7951#   7955#   7957#   7999    8000
M$XFER            1#    1959#   7949#   7955#
NODCL           1965#   2795    2796    2797    2798    2799    2800    2801    2802    2803    2804    2805    2806    2807    2808
                2809    2810    2811    2812    2813    2814    2815    2819    2820    2821    2822    2823    2824    2825    2826
                2827    2828    2829    2830    2831    2832    2833    2834    2835    2836    2840    2841    2842    2843    2844
                2849    2850    2851    2852    2853    2856    2857    2858    2859    2860    2861    2862    2863    2866    2867
                2868    2869    2870    2871    2874    2875    2876    2877    2879    2880    2881    2882    2883    2885    2886
                2888    2889    2892    2895    2896    2897    2898    2899    2900    2901    2902    2903    2904    2905    2906
                2907    2908    2909    2910    2912    2913    2914    2915    2916    2917    2918    2921    2922    2923    2924
```

```
        2925    2926    2929    2930    2931    2934    2937    2938    2939    2940    2941    2942    2943    2944    2947
        2948    2949    2954    3460    3461    3462    3463    3464    3465    3466    3467    3468    3469    3470    3471
        3472    3473    3474    3475    3476    3477    3478    3479    3480    3481
OPEN       1#   1959#
POINTE     1#   1959#   1971
PRINTB     1#   1959#   3038    3055    3068    3082    3102    3121
PRINTF     1#   1959#   3359    3367    3405    3433    3443    3502    3563    3876    3886    3895    3940    4322    4365
        4563    4782    4824    4877    4886    4923    4938    4964    4980    5080    5152    5199    5229    5302    5351
        5368    5838    5935    5962    7301
PRINTS     1#   1959#   3549    3583    3594    3619    3648    3671    3681    3696    3710    3717    3743    3757    3772
        3789    3798    3818    3838    4135    4171
PRINTX     1#   1959#
READBU     1#   1959#   4544
READEF     1#   1959#   4494    4499    4504    4510
RFLAGS     1#   1959#
SETPRI     1#   1959#   4646    4686
SETVEC     1#   1959#   4621    4631    4638    4830    6918
SLASH      1#   1959#
STARS      1#   1959#
SVC        1#   1959#
XFER       1#   1959#   3135#   4650#   4702#   4723#   4745#   4851#   4911#
XFERF      1#   1959#   7954
XFERT      1#   1959#   7948

. ABS.  046616     000

ERRORS DETECTED:  0

CZDCLA,CZDCLA.LST/CRF/SOL=SVC34R.MLB,CZDCLA.P11
RUN-TIME: 27 34 4 SECONDS
RUN-TIME RATIO: 91/66=1.3
CORE USED:  22K  (43 PAGES)
```