

DMR11, DMC11

DMR/C11 DCLT  
CZCL KAO

AH-F593A-MC  
FICHE 1 OF 1

JUN 1980  
COPYRIGHT © 1980  
MADE IN USA



A large grid of approximately 100 small tables, each containing technical data, likely a parts list or assembly drawing. The data is organized in columns and rows, with various alphanumeric codes and symbols. The text is small and difficult to read, but the layout is consistent across the grid.



1  
2

.TITLE CZCLKAO DMR,DMC-11 DATA COMM. LINK TEST

.REM 8

IDENTIFICATION  
-----

PRODUCT CODE: AC-F591A-MC  
PRODUCT NAME: CZCLKAO DMR/C11 DCLT  
PRODUCT DATE: 17-APRIL-80  
MAINTAINER: MERRIMACK DIAGNOSTIC ENGINEERING  
AUTHOR: BRUCE LUHRS - BRUCE RIBOLINI

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1980 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

REVISION HISTORY:

REV ---	DATE ----	AUTHOR -----	REASON -----
A	23-APR-80	BRUCE LUHRS BRUCE RIBOLINI	ORIGINAL ISSUE, DCLT FOR THE DMC OR DMR-11

TABLE OF CONTENTS

- 1.0 GENERAL INFORMATION
  - 1.1 PROGRAM ABSTRACT
  - 1.2 SYSTEM REQUIREMENTS
  - 1.3 RELATED DOCUMENTS AND STANDARDS
  - 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
  - 1.5 ASSUMPTIONS - RESTRICTIONS
- 2.0 OPERATING INSTRUCTIONS
  - 2.1 COMMANDS
  - 2.2 SWITCHES
  - 2.3 FLAGS
  - 2.4 HARDWARE QUESTIONS
  - 2.5 DATA COMM. LINK TEST COMMANDS
    - 2.5.1 MESSAGE COMMANDS
    - 2.5.2 RUN COMMAND
    - 2.5.3 DEFAULTS
  - 2.6 QUICK STARTUP PROCEDURE
- 3.0 ERROR INFORMATION
  - 3.1 TYPES OF ERROR MESSAGES
  - 3.2 SPECIFIC ERROR MESSAGES
- 4.0 PERFORMANCE AND PROGRESS REPORTS
  - 4.1 PRINTING EVENT LOG
  - 4.2 OPERATOR STATUS MESSAGES
  - 4.3 PRINTING DMR,DMC-11 BASE TABLE
- 5.0 DEVICE INFORMATION TABLES
- 6.0 MODE AND MESSAGE DESCRIPTIONS
  - 6.1 MODE DESCRIPTIONS
  - 6.2 MESSAGE DESCRIPTIONS
  - 6.3 INTERFACING TO AN 'ITEP' NODE
  - 6.4 TROUBLESHOOTING HINTS

## 1.0 GENERAL INFORMATION

### 1.1 PROGRAM ABSTRACT

THIS DCLT (DATA COMMUNICATION LINK TEST) PROGRAM IS MEANT TO PROVIDE FIELD SERVICE WITH A TOOL TO MAINTAIN DMR/DMC-11 TO DMR/DMC-11 COMMUNICATION LINKS. THIS DCLT PROGRAM WILL PROVIDE THE COVERAGE NECESSARY TO DETECT FAILURES TO THE COMPUTER EQUIPMENT, THE COMMUNICATION LINK, OR THE MODEM.

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

### 1.2 SYSTEM REQUIREMENTS

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

- A PDP-11 CPU
- MINIMUM OF 24K WORDS OF MEMORY
- A WORKING, LINE OR REAL-TIME CLOCK
- A CONSOLE TERMINAL
- ANY XXDP+ SUPPORTED LOAD MEDIA
- ONE OF THESE DMR-11 OR DMC-11 CONFIGURATIONS:
  - DMC11-AL - LOCAL MICROPROCESSOR
  - DMC11-AL - REMOTE MICROPROCESSOR
  - DMC11-DA - E.I.A. LINE UNIT
  - DMC11-FA - CCITT V.35 LINE UNIT
  - DMC11-MA - 1M BPS LINE UNIT
  - DMC11-MD - 56K BPS LINE UNIT
  
  - DMR11-AA - E.I.A. (RS 232/423)
  - DMR11-AB - CCITT V.35
  - DMR11-AC - LOCAL
  - DMR11-AE - E.I.A. (RS 422)

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

### 1.3 RELATED DOCUMENTS AND STANDARDS

- XXDP+ USER'S MANUAL (CHQUS?.SEQ WHERE ? IS THE REV. LEVEL OF THE MANUAL - 'C' IS THE CURRENT REV.).

#### 1.4 DIAGNOSTIC HIERARCY PREREQUISITES

THE GOAL OF THE DATA COMM. LINK TEST PROGRAM IS TO TEST THE COMMUNICATION LINK AND THEREFORE ASSUMES THAT THE CPU'S, CLOCKS, AND DMR OR DMC-11'S AT EACH END OF THE LINK HAVE ALREADY BEEN TESTED.

IF NO LINE OR REAL-TIME CLOCK IS FOUND, THE PROGRAM WILL CONTINUE BUT ANY OF THE PROGRAM THAT TIMES THE DEVICE WILL HANG IF THE DEVICE TIMES OUT. ALSO, THE EVENT LOG WILL CONTAIN A ZERO EVENT TIME FOR ALL EVENTS LOGGED.

IT IS NOT THE INTENTION OF A DATA COMM. LINK TEST PROGRAM TO TEST THE DMR OR DMC-11, BUT TO TEST THE COMMUNICATION LINK TO WHICH THEY ARE CONNECTED.

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

DMR: CZDMIAO DMR-11 FCTNL DIAG  
CZDMPA1 M8207 STATIC DIAG #1  
CZDMQA2 M8207 STATIC DIAG #2  
CZDMRCO M8203 STATIC DIAG #1  
CZDMSCO M8203 STATIC DIAG #2

DMC: CZDMCCO BSC W/R MICRO-PROC TST  
CZDMECO DDCMP MDLN UNIT TST  
CZDMGDO DMC-11 CROM + JMUP TEST  
MD-11-DZDMHB1 DMC-11 FREE RUNNING TEST

#### 1.5 ASSUMPTIONS - RESTRICTIONS

IT IS ASSUMED THAT THE COMMUNICATIONS DEVICE (DMC OR DMR-11) HAS BEEN TESTED USING THE PREREQUISTE DIAGNOSTICS. THE OPERATOR SHOULD HAVE READ THE USER DOCUMENTATION PORTION OF THE LISTING TO FAMILIARIZE HIMSELF WITH THE COMMANDS AND CAPABILITIES AVAILABLE UNDER THE DIAGNOSTIC SUPERVISOR AND DCLT.

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

## 2.0 OPERATING INSTRUCTIONS

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

### 2.1 COMMANDS

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

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

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

### 2.2 SWITCHES

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

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:DDDD	EXECUTE DDDD PASSES (DDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN SECTION 2.3.
/EOP:DDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDD PASSES ONLY. (DDDD = 1 TO 64000)
/UNITS:LIST	TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10-12 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

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

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

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

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

### 2.3 FLAGS

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

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



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

\*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

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

/FLAGS:LOE:IER:BOE

#### 2.4 HARDWARE QUESTIONS

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

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

# UNITS (D) ? 1<CR>

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

## 2.5 DATA COMM. LINK TEST COMMANDS

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

### MESSAGE COMMANDS AVAILABLE:

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

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

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

### 2.5.1 MESSAGE COMMANDS

COMMAND	DESCRIPTION
CLEAR EXPECTLIST	ZEROES THE EXPECTLIST (000'S) AND THEN PUTS DEFAULT ITEP MSG IN SO NOT REALLY EMPTY
CLEAR TRANSMITLIST	ZEROES TRANSMITLIST (000'S) AND THEN PUTS DEFAULT ITEP MSG IN SO NOT REALLY EMPTY
HELP ?	TYPES HELP INFO FOR OPERATOR
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)
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
PRINT		PRINTS THE EVENT LOG AFTER ASKING THE OPERATOR IF HE WANTS THE DMR/DMC-11 BASE TABLE PRINTED
DUMP	SSSSSS-EEEEEE/B	PRINTS THE CONTENTS OF THE MEMORY LOCATIONS BETWEEN OCTAL ADDRESSES 'SSSSSS' AND 'EEEEEE' WHERE 'SSSSSS' IS THE START ADDRESS AND '-EEEEEE' IS THE END ADDRESS.  IF '-EEEEEE' IS NOT SPECIFIED THEN THE CONTENTS OF 'SSSSSS' IS PRINTED IN WORD FORMAT.

WHERE '/B' IS OPTIONAL:  
DEFAULT IS PRINT WORDS  
'/B' CAUSES PRINT BYTES

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

#### 2.5.2 RUN COMMAND

COMMAND

DESCRIPTION

-----  
RUN MODE=MTYPE/QUAL

-----  
STARTS DCLT EXECUTING IN THE  
MODE SPECIFIED

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

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

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

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

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

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

NOTE: IF BOTH MODES 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)  
/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:

=INTERNAL TTL

=CABLE

=LOCALMODEM (DMR IN DMR MODE AND RS449 MODEMS ONLY.  
CAUSES A 'WRITE MODEM' TO BE DONE TO SET UP  
LOCAL-LOOPBACK (MAINT1) . ALSO CALLED  
ANALOG-LOOPBACK.

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

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

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

### 2.5.3 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  
ZEROS - /SIZE=64/COPY=0  
OALT - /SIZE=64/COPY=0  
1ALT - /SIZE=64/COPY=0  
CCITT - /SIZE=64/COPY=0  
ALPHA - /SIZE=65/COPY=0  
ITEP - /SIZE=58/COPY=0  
OPER. SPEC'D - /SIZE=LENGTH-OF-TEXT-TYPED-BETWEEN-QUOTES/COPY=0

FOR THE RUN COMMAND THE DEFAULTS ARE:

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

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

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

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

### OTHER NOTES:

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

## 2.6 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

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

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

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

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

### 3.0 ERROR INFORMATION

#### 3.1 TYPES OF ERROR MESSAGES

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

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

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

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

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

#### 3.2 SPECIFIC ERROR MESSAGES

##### COMMAND LINE INTERPRETER ERRORS:

ERROR MESSAGE:	MEANING
?ILL (MD)-BAD SYNTAX?	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 REQUIRED 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.

DCLT OR DEVICE ERROR MESSAGES:

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

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

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

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

BUFFER FULL - MSG. NOT BUILT !!

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

CHAR. COUNT EXCEEDS BUFF LIMIT - MSG TRUNCATED



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

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 MISMATCHES 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 RECEIVE. WAS A DIFFENT LENGTH THEN THE MESSAGE THAT WAS EXPECTED.

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

TIME OUT WAITING FOR RDI TO CLEAR  
SELO SEL2  
XXXXXX XXXXXX

THIS MEANS THAT A SOFTWARE TIMER EXPIRED BEFORE  
THE DEVICE CLEARED RDI IN RESPONSE TO THE DROPPING  
OF RDI.

NOTE: PROGRAM RESETS TIMER AND WAITS AGAIN  
SO AN EFFECTIVE LOOP ON ERROR IS SETUP.

TIME OUT WAITING FOR RDI TO SET  
SELO SEL2  
XXXXXX XXXXXX

THIS MEANS THAT A SOFTWARE TIMER EXPIRED BEFORE  
THE DEVICE CAUSED AN INTERRUPT IN RESPONSE TO THE  
PROGRAM SETTING RDI.

NOTE: PROGRAM RESETS TIMER AND WAITS AGAIN  
SO AN EFFECTIVE LOOP ON ERROR IS SETUP.

TIME OUT WAITING FOR RUN TO SET  
SELO SEL2  
XXXXXX XXXXXX

THIS MEANS THAT A SOFTWARE TIMER EXPIRED BEFORE  
THE DEVICE SET THE RUN BIT IN RESPONSE TO THE  
PROGRAM SETTING MASTER CLEAR.

NOTE: PROGRAM RESETS TIMER AND ISSUES ANOTHER  
MASTER CLEAR AND WAITS AGAIN SO AN EFFECTIVE  
LOOP ON ERROR IS SETUP.  
THIS ERROR COULD INDICATE WRONG ADDRESS FOR  
DMR/DMC-11 WAS GIVEN IN HARDWARE P TABLE.

TIME OUT WAITING FOR OUTPUT INTERRUPT  
SELO SEL2  
XXXXXX XXXXXX

THIS MEANS THAT A SOFTWARE TIMER EXPIRED BEFORE  
THE DEVICE SET OUTPUT INTERRUPT IN RESPONSE TO  
PROGRAM REQUESTING DEVICE TO TRANSMIT OR RECIEVE.

NOTE: PROGRAM RESETS TIMER AND WAITS AGAIN SO AN  
EFFECTIVE LOOP ON ERROR IS SET UP.  
THIS ERROR WILL OCCUR WHEN ONE NODE IS STARTED  
IN RX OR TX MODE AND THE OTHER IS STILL BEING  
SET UP. IGNORE THIS ERROR IF PROGRAM CONTINUES  
WITHOUT FURTHER ERRORS.

INPUT INTERRUPT WHEN EXPECTING OUTPUT  
SELO SEL2  
XXXXXX XXXXXX

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

ILLEGAL OUTPUT INTERRUPT  
SEL2 SEL6  
XXXXXX XXXXXX

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

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

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

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

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

TX BUFF COMPLETED AND SHOULD BE RX  
SEL4 SEL6  
XXXXXX XXXXXX

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

RX BUFF COMPLETED AND SHOULD BE TX  
SEL4 SEL6  
XXXXXX XXXXXX

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

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

DOWN LINE LOAD ABORTED

THIS ERROR CAN ONLY OCCUR IN A NODE THAT

IS A DLL 'HOST' WHEN IT HAPPENS IT ALSO  
PRINTS ONE OF THE FOLLOWING QUALIFERS:

TX NOT COMPLETE

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

RX NOT COMPLETE

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

SEC REQ WORD1

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

SEC REQ WORD2

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

CALLED FROM PC. XXXXXX

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

#### 4.0 PERFORMANCE AND PROGRESS REPORTS

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

#### 4.1 PRINTING OF EVENT LOG

SIGNIFICANT EVENTS OR CHECK-POINTS WILL BE LOGGED IN A 'CIRCULAR QUEUE' STORAGE AREA CALLED THE EVENT LOG. THE LAST 'N' EVENTS ARE KEPT LOGGED AND CAN BE LISTED ON THE OPERATORS CONSOLE BY GIVING A 'PRINT' COMMAND AT THE 'DR>' (DIAGNOSTIC SUPERVISOR) OR 'DCLT>' (DCLT) LEVEL. 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 AVAILIABLE TO STORE RELATIVE MINUTES SO 'TIME' WILL WRAP TO 000:00:00 AFTER 256:59:59.

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

THE TYPES OF EVENTS KEPT IN THE EVENT LOG ARE:

TRANSMIT MESSAGE QUEUED:

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

TRANSMIT MESSAGE COMPLETED:

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

RECEIVE SPACE QUEUED:

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

RECEIVE MESSAGE COMPLETED:

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

DATA COMPARISON STARTED:

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

DATA COMPARISON DATA ERROR:

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

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

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

#### 4.2 OPERATOR STATUS MESSAGES

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

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

#### 4.3 PRINTING OF DMR/DMC-11 BASE TABLE

WHEN THE 'PRINT' COMMAND IS GIVEN, BEFORE THE EVENT LOG IS  
PRINTED, THE DCLT DMR/DMC-11 DCLT PROGRAM ASKS IF YOU WISH  
TO HAVE THE CONTENTS OF THE DEVICE'S BASE TABLE PRINTED:

BASE TABLE (L) N ?

IF A 'Y' IS TYPED AS AN ANSWER, THE 256. BYTES OF BASE TABLE  
WILL BE PRINTED IN THE FOLLOWING FORMAT:

017370:	000	001	002	003	004	005	006	007
017400:	010	011	012	013	014	015	016	017

```
  ::      ::      ::      ::      ::      ::      ::      ::      ::  
  ::      ::      ::      ::      ::      ::      ::      ::      ::  
017760:  570  371  372  373  374  375  376  377
```

## 5.0 DEVICE INFORMATION TABLES

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

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

.WORD 1	: [0] FULL OR HALF DUPLEX FLAG (BIT0=1 IF FULL)
.WORD 160170	: [2] CSR ADDRESS
.WORD 300	: [4] INTERRUPT VECTOR
.WORD 240	: [6] INTERRUPT PRIORITY (5)
.WORD 0	: [10] SPARE
.WORD 0	: [12] OPTION TYPE (0=DMC,5=DMR-DMC MODE,7=DMR)

## 6.0 MODE AND MESSAGE DESCRIPTIONS

### 6.1 MODE DESCRIPTIONS

#### TRANSMIT MODE

A LIST OF MESSAGES IS TRANSMITTED WITHOUT EXPECTING ANY DATA TO BE RECEIVED.

#### 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.

#### 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.



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!

DOWN-LINE-LOAD

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

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 SWITCHED

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'.

MAINTENANCE 'LOOP' MODES

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

INTERNALTTL THE 'LU LOOP' BIT IS SET SO THAT THE UNIT'S SERIAL LINE OUT IS LOOPED BACK TO THE SERIAL LINE IN AT THE TTL LEVEL BEFORE LEVEL CONVERSION.

CABLE NOT USED BY DMR OR DMC-11 CODE.

LOCALMODEM FOR DMR-11 IN DMR MODE AND RS449 MODEMS ONLY. CAUSES A 'WRITE MODEM' TO BE DONE TO SET UP LOCAL-LOOPBACK (MAINT1) . ALSO CALLED ANALOG-LOOPBACK.

REMOTEMODEM FOR DMR-11 IN DMR MODE AND RS449 MODEMS ONLY. CAUSES A 'WRITE MODEM' TO BE DONE TO SET UP REMOTE-LOOPBACK (MAINT2) . ALSO CALLED DIGITAL-LOOPBACK.

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

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

\*= MOST LIKELY TO BE IN THAT MODE

### 6.2 MESSAGE DESCRIPTIONS

NAME	DESCRIPTION
ZEROES	MESSAGE OF ALL 0'S (00000000,00000000,00000000,...)
ONES	MESSAGE OF ALL 1'S (11111111,11111111,11111111,...)
TALT	MESSAGE OF ALTERNATING 1'S (10101010,10101010,...)
OALT	MESSAGE OF ALTERNATING 0'S (01010101,01010101,...)
CCITT	'CCITT' 512-BIT (VS. 511 BITS) TEST PATTERN



### 6.3 INTERFACING TO AN 'ITEP' NODE

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

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

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

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

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

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

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

### 6.4 TROUBLESHOOTING HINTS

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

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

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

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 COMMAND SEQUENCE IS:

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

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 PREREQUISTE DIAGNOSTICS FOR THE COMM.  
DEVICE.

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

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

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

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

2.) TRANSMIT ON ONE NODE RECEIVE ON THE OTHER

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

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

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
```

NOW RUN THRU THE SEQUENCE AGAIN WITH NODE A RECEIVING  
AND NODE B RECEIVING.

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=1/NOCH/PAS=3

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.

#### 4.) BOTH NODES ACTIVE

NOW BOTH NODES CAN BE RUN IN ACTIVE WITH DATA CHECKING ON.  
STAT'S 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

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

#### 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

CZCLKAO DMR,DMC-11 DATA COMM. LINK TEST  
CZCLKA.P11 18-APR-80 09:24

MACY11 30A(1052) <sup>E 3</sup> 18-APR-80 09:24 PAGE 31

SEQ 0030

LIS>

TLK>

&

1401  
1402  
1403  
1404 002000  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415 002000  
1416  
1417  
1418  
1419 002000  
1420 002000  
1421 002000 103  
1422 002001 132  
1423 002002 103  
1424 002003 114  
1425 002004 113  
1426 002005 000  
1427 002006 000  
1428 002007 000  
1429 002010  
1430 002010 101  
1431 002011  
1432 002011 060  
1433 002012  
1434 002012 000000  
1435 002014  
1436 002014 003410  
1437 002016  
1438 002016 035764  
1439 002020  
1440 002020 000000  
1441 002022  
1442 002022 002130  
1443 002024  
1444 002024 000000  
1445 002026  
1446 002026 036346  
1447 002030  
1448 002030 000000  
1449 002032  
1450 002032 000000  
1451 002034  
1452 002034 000000  
1453 002036  
1454 002036 000000  
1455 002040  
1456 002040 002124

.SBTTL PROGRAM HEADER

BGNMOD

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

POINTER BGNRPT,BGNAU,BGN DU

HEADER CZCLK,A,0,1800.,0

LSNAME::  
    .ASCII /C/  
    .ASCII /Z/  
    .ASCII /C/  
    .ASCII /L/  
    .ASCII /K/  
    .BYTE 0  
    .BYTE 0  
    .BYTE 0  
LSREV::  
    .ASCII /A/  
LSDEPO::  
    .ASCII /O/  
LSUNIT::  
    .WORD 0  
LSTIML::  
    .WORD 1800.  
LSHPCP::  
    .WORD LSHARD  
LSSPCP::  
    .WORD 0  
LSHPTP::  
    .WORD LSHW  
LSSPTP::  
    .WORD 0  
LSLADP::  
    .WORD LSLAST  
LSSTA::  
    .WORD 0  
LSCO::  
    .WORD 0  
LSDTYP::  
    .WORD 0  
LSAPT::  
    .WORD 0  
LSDTP::  
    .WORD LSDISPATCH



1457 002042  
 1458 002042 000000  
 1459 002044  
 1460 002044 000000  
 1461 002046  
 1462 002046 000000  
 1463 002050  
 1464 002050 003  
 1465 002051 003  
 1466 002052  
 1467 002052 000000  
 1468 002054 000000  
 1469 002056  
 1470 002056 000000  
 1471 002060  
 1472 002060 012026  
 1473 002062  
 1474 002062 024364  
 1475 002064  
 1476 002064 000000  
 1477 002066  
 1478 002066 000000  
 1479 002070  
 1480 002070 025316  
 1481 002072  
 1482 002072 025310  
 1483 002074  
 1484 002074 000000  
 1485 002076  
 1486 002076 012042  
 1487 002100  
 1488 002100 104035  
 1489 002102  
 1490 002102 000000  
 1491 002104  
 1492 002104 024400  
 1493 002106  
 1494 002106 025266  
 1495 002110  
 1496 002110 025264  
 1497 002112  
 1498 002112 024372  
 1499 002114  
 1500 002114 000000  
 1501 002116  
 1502 002116 000000  
 1503 002120  
 1504 002120 000000  
 1505  
 1506

L\$PRIO::  
 .WORD 0  
 L\$ENVI::  
 .WORD 0  
 L\$EXP1::  
 .WORD 0  
 L\$MREV::  
 .WORD 0  
 .BYTE C\$REVISION  
 .BYTE C\$EDIT  
 L\$EF::  
 .WORD 0  
 .WORD 0  
 L\$SPC::  
 .WORD 0  
 L\$DEVP::  
 .WORD L\$DVTYP  
 L\$REPP::  
 .WORD L\$RPT  
 L\$EXP4::  
 .WORD 0  
 L\$EXP5::  
 .WORD 0  
 L\$AUT::  
 .WORD L\$AU  
 L\$DUT::  
 .WORD L\$DU  
 L\$!UN::  
 .WORD 0  
 L\$DESP::  
 .WORD L\$DESC  
 L\$LOAD::  
 EMT ESLOAD  
 L\$ETP::  
 .WORD 0  
 L\$ICP::  
 .WORD L\$INIT  
 L\$CCP::  
 .WORD L\$CLEAN  
 L\$ACP::  
 .WORD L\$AUTO  
 L\$PRT::  
 .WORD L\$PROT  
 L\$TEST::  
 .WORD 0  
 L\$DLY::  
 .WORD 0  
 L\$HIME::  
 .WORD 0

1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518

.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  
002122 000001  
002124  
002124 025324

.WORD 1  
LSDISPATCH::  
.WORD T1

1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560

```
.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.
:--

          BGNHW  DFPTBL

                                .WORD  L10000-LSHW/2
                                LSHW::
                                DFPTBL::

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

          .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'

          .WORD  160170           ;[2] CSR ADDRESS
          .WORD  300              ;[4] INTERRUPT VECTOR
          .WORD  240              ;[6] INTERRUPT PRIORITY (5)
          .WORD  0                ;[10] DEVICE PARAMETERS WORD
                                ; (ENABLE CRC, STRIP SYNC, COMPATIBLE MODE...)
          .WORD  0                ;[12] DEVICE OPTION TYPE(0=DMC,5=DMR-DMC MODE,
                                ; 7=DMR.
          .WORD  4                ;[14] BAUD RATE (0=2.4K, 1=4.8K, 2=9.6K, 3= 19.2K,
                                ; 4=56K, 5=250K, 6=500K, 7=1 MEGA-BAUD)
          .WORD  0                ;[16] LINE INTERFACE (422, V.35, INT, EIA...)

          ENDPHW

                                L10000:
```

002126  
002126 000010  
002130  
002130

002130 000001

002132 160170  
002134 000300  
002136 000240  
002140 000000  
002142 000000  
002144 000004  
002146 000000  
002150  
002150

1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616

002150

.SBTTL GLOBAL EQUATES SECTION

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

EQUALS

:  
: BIT DIFINITIONS

:  
BIT15== 100000  
BIT14== 40000  
BIT13== 20000  
BIT12== 10000  
BIT11== 4000  
BIT10== 2000  
BIT09== 1000  
BIT08== 400  
BIT07== 200  
BIT06== 100  
BIT05== 40  
BIT04== 20  
BIT03== 10  
BIT02== 4  
BIT01== 2  
BIT00== 1

:  
BIT9== BIT09  
BIT8== BIT08  
BIT7== BIT07  
BIT6== BIT06  
BIT5== BIT05  
BIT4== BIT04  
BIT3== BIT03  
BIT2== BIT02  
BIT1== BIT01  
BIT0== BIT00

:  
: EVENT FLAG DEFINITIONS  
: EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

:  
EF.START== 32. : START COMMAND WAS ISSUED  
EF.RESTART== 31. : RESTART COMMAND WAS ISSUED  
EF.CONTINUE== 30. : CONTINUE COMMAND WAS ISSUED  
EF.NEW== 29. : A NEW PASS HAS BEEN STARTED  
EF.PWR== 28. : A POWER-FAIL/POWER-UP OCCURRED  
:

```
1617  
1618  
1619  
1620          000340  
1621          000300  
1622          000240  
1623          000200  
1624          000140  
1625          000100  
1626          000040  
1627          000000  
1628  
1629  
1630  
1631          000004  
1632          000010  
1633          000020  
1634          000040  
1635          000100  
1636          000200  
1637          000400  
1638          001000  
1639          002000  
1640          004000  
1641          010000  
1642          020000  
1643          040000  
1644          100000  
1645
```

```
.;  
.; PRIORITY LEVEL DEFINITIONS  
.;  
PRI07== 340  
PRI06== 300  
PRI05== 240  
PRI04== 200  
PRI03== 140  
PRI02== 100  
PRI01== 40  
PRI00== 0  
.;  
.; OPERATOR FLAG BITS  
.;  
EVL==      4  
LOT==      10  
ADR==      20  
IDU==      40  
ISR==     100  
UAM==     200  
BOE==     400  
PNT==    1000  
PRI==    2000  
IXE==    4000  
IBE==   10000  
IER==   20000  
LOE==   40000  
HOE==  100000
```

```
1646          ;***** INDEPENDENT EQUATES
1647
1648          001000          BUFLIM=512.          ;MAX BUFFER SIZE IN BYTES
1649
1650          000017          MSGLIM=15.          ; APPLIES TO TX,RX AND CMP BUFFS
1651
1652
1653
1654
1655
1656          ;MODE OF OPERATION EQUATES
1657          000000          REC=0          ;RECEIVE MODE
1658          000001          TRA=1          ;TRANSMIT MODE
1659          000002          PAS=2          ;PASSIVE MODE
1660          000003          ACT=3          ;ACTIVE MODE
1661          000004          DOW=4          ;DOWN-LINE-LOAD MODE
1662          000005          TAL=5          ;TALK MODE
1663          000006          LIS=6          ;LISTEN MODE
1664
1665          ;MAINT LOOP TYPE EQUATES
1666          000000          NONE= 0          ;NO LOOP
1667          000001          TTL= 1          ;INTERNAL TTL
1668          000002          CABLE= 2          ;CABLE LOOP
1669          000003          MODLOC= 3          ;MODMEM LOCAL
1670          000004          MODREM= 4          ;MODEM REMOTE
1671          000005          MOP= 5          ;MOP
1672
1673          ;CLOCK ENABLE VALUES TO BE LOADED IN CLK'S CSR
1674          000100          LCLKEN= 100          ;L-CLOCK CSR VALUE TO ENABLE THE CLOCK
1675          000111          PCLKEN= 111          ;P-CLOCK CSR VALUE TO ENABLE THE CLOCK
1676          001600          PCLKCT= 1600          ;P-CLOCK COUNT SET REGISTER FOR COUNTER
1677
1678          ;PARAM WORD EQUATES
1679
1680          000001          STATB= BIT0          ;OPERATOR AWAKE ASKED FOR
1681          000002          DATCKB= BIT1          ;DATA CHECK BIT
1682          000004          ECHOB= BIT2          ;ECHO BIT
1683          000020          CRCB= BIT4          ;CRC CALCUALTE ASKED FOR
1684          000040          PROTOB= BIT5          ;PROTOCOL PROCESSING ASKED FOR
1685
1686          ;OPTION TYPE EQUATES
1687
1688          000000          DMC= 0          ;DMC
1689          000004          DMRC6= 4          ;8206 DMR IN DMC MODE
1690          000005          DMRC7= 5          ;8207 DMR IN DMC MODE
1691          000006          DMR6= 6          ;8206 DMR IN DMR MODE
1692          000007          DMR7= 7          ;8207 DMR IN DMR MODE
1693
1694          ;EVENT LOG MESSAGE TYPES (USED TO LOCATE EVENT DESCRIPTION IN EVENT TABLE
1695          ; AND DISPATCHING TO SEPERATE SECTIONS OF THE EVENT REPORTING SECTION)
1696          000000          TXQ= 0          ;TRANSMIT MESSAGE QUEUED
1697          000002          TXC= 2          ;TRANSMIT COMPLETE
1698          000004          RXQ= 4          ;RECEIVE BUFFER QUEUED
1699          000006          RXC= 6          ;RECEIVE COMPLETE
1700          000010          DER= 10          ;DEVICE INFORMATION
1701          000012          DVI= 12          ;DEVICE ABOUT TO INIT
```

```
1702      000014      DCK= 14      ;DATA COMPARISON RESULTS
1703
1704      000020      DLE= 20      ;DATA COMPARISON LENGH ERROR
1705      000022      DDE= 22      ;DATA COMPARISON DATA ERROR
1706      000024      EOP= 24      ;END OF PASS
1707
1708      ;;;;EQUATES FOR FLAG WORD;;;;;
1709
1710      000001      ININT= 1      ;INPUT INT. REC.
1711      000002      OTINT= 2      ;OUTPUT INT REC
1712      000004      QRX= 4      ;RX QUED /COMPL
1713      000010      QTX= 10     ;TX QUED/COMPL
1714      000020      CTX= 20     ;TX COMPL AND IN TXSEL4 AND TSEL6
1715      000040      CRX= 40     ;RX COMPL AND IN TSEL4 AND TSEL6
1716      000100      ERX= 100    ;EXPECT TO GET A RX COMPLETED
1717      000200      ETX= 200    ;EXPECT TO GET A TX COMPLETED
1718      000400      DLLGA= 400   ;DOWN LINE LOAD GO AHEAD BIT
1719      001000      DMRRUN= 1000  ;DMR RUN MODE EXPECTED
1720
1721      ; SPECIAL CLI CODES FOR 'CHAR' ARGUMENT IN CLI CALLS
1722      ; (COMMAND LINE INTERPRETER DEFINITIONS)
1723      000000      CLIERR= 0
1724      000001      CLIEXI= 1
1725      000002      CLIBR= 2
1726      000003      CLIBIF= 3
1727      000004      CLISPA= 4
1728      000005      CLINUM= 5
1729      000006      CLIALP= 6
1730      000007      CLIALN= 7
1731      000010      CLIOCT= 8.
1732      000011      CLIDEC= 9.
1733      000012      CLISTR= 10.
1734
1735      ; DEFS FOR COMMAND LINE INTERPRETATION ACTION VALUES
1736      000000      NULL=0
1737      000001      CLEAR=1
1738      000002      SHOW=2
1739      000003      CHECK=3
1740      000004      RUN=4
1741      000005      HLP=5
1742      000006      CSHEXP=6
1743      000007      CSHTRN=7
1744      000010      SETEXP=10
1745      000011      SETTRN=11
1746      000012      SIZE=12
1747      000013      QCOPY=13
1748      000014      NUM=14
1749      000015      OPRMSG=15
1750      000016      STATUS=16
1751      000017      ENDQO=17
1752      000020      CMSG0=20
1753      000021      CMSG1=21
1754      000022      CMSG2=22
1755      000023      CMSG3=23
1756      000024      CMSG4=24
1757      000025      CMSG5=25
```

1758	000026	CMSG6=26
1759	000027	ATVMOD=27
1760	000030	PASMOD=30
1761	000031	RECMOD=31
1762	000032	LISMOD=32
1763	000033	DLLMOD=33
1764	000034	TRAMOD=34
1765	000035	TALMOD=35
1766	000036	NO=36
1767	000037	ECHO=37
1768	000040	CRC=40
1769	000041	PROTO=41
1770	000042	PASC=42
1771	000043	MOP=43
1772	000044	TTLLOP=44
1773	000045	CBLLLOP=45
1774	000046	LMDLOP=46
1775	000047	RMDLOP=47
1776	000050	NOTNUF=50
1777	000051	BADCHR=51
1778	000052	DMP5=52
1779	000053	DMPE=53
1780	000054	DMPQ=54
1781	000055	PRNT=55

:\*\*\*\*\* DEVICE DEPENDENT EQUATES  
: MODEM SIGNAL BIT DEFINITIONS  
: IF SIGNAL AVAILABLE IN DEVICE, EQUATE NAME TO BIT POSITION,  
: ELSE EQUATE IT TO = 0

1788			
1789	000004	CTS= BIT2	:CLEAR TO SEND (CIRCUIT CB)
1790	000010	DSR= BIT3	:DATA SET READY (CIRCUIT CC)
1791	000001	DCD= BIT0	:DATA CARRIER DETECT (CIRCUIT CF)
1792	000040	RTS= BIT5	:REQUEST TO SEND (CIRCUIT CA)
1793	000200	RI= BIT7	:RING INDICATOR (CIRCUIT CE)
1794	040000	SQD= BIT14	:SIGNAL QUALITY DETECT (CIRCUIT CG)
1795	001000	TM= BIT9	:MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)

: DEVICE SIGNALS

1799			
1800	000040	RQI= BIT5	:REQUEST IN
1801	000200	RDI= BIT7	:READY IN
1802	000200	RDO= BIT7	
1803	000001	BACC= BIT0	:BUFFER ADDR. CHAR COUNT
1804	040000	MCLR= BIT14	:MASTER CLEAR
1805	004000	LULOOP= BIT11	:LINE UNIT LOOP(TTL)
1806	000400	MAINTB= BIT8	:MAINT MODE BIT
1807	002000	HALFDB= BIT10	:HALF DUPLEX BIT
1808	000004	RXBIT= BIT2	:RX BIT
1809	000100	IEO= BIT6	:ENABLE OUTPUT INTERRUPT BIT
1810			



1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866

002150  
002150 000001  
002152 000001  
002154 000001  
002156 000001  
002160 000100  
002162 000072  
002164 000101  
002166 000000  
002170 000001  
002172 000005  
002174 000206  
  
002176  
002176 002220  
002200 002221  
002202 002222  
002204 002223  
002206 002224  
002210 002324  
002212 002416  
002214 002524  
002216 002646  
  
002220 000  
002221  
002221 377  
002222  
002222 252  
002223  
002223 125  
002224  
002224 177603 157427 031011  
002232 047321 163715 105221  
002240 143325 142304  
002244 040041 014116 052606  
002252 172334 105025 123754  
002260 111337 111523  
002264 030030 145064 137642  
002272 143531 063617 135075  
002300 066730 026575  
002304 052012 053627 070071  
002312 151172 165044 031605

.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  
  
DMSGCT:  
MSG0C: .WORD EMSG0-MSG0 ;BYTE COUNT OF MESSAGE #0  
MSG1C: .WORD EMSG1-MSG1 ;BYTE COUNT OF MESSAGE #1  
MSG2C: .WORD EMSG2-MSG2 ;BYTE COUNT OF MESSAGE #2  
MSG3C: .WORD EMSG3-MSG3 ;BYTE COUNT OF MESSAGE #3  
MSG4C: .WORD EMSG4-MSG4 ;BYTE COUNT OF MESSAGE #4  
MSG5C: .WORD EMSG5-MSG5 ;BYTE COUNT OF MESSAGE #5  
MSG6C: .WORD EMSG6-MSG6 ;BYTE COUNT OF MESSAGE #6  
OPCNT: .WORD 0 ;BYTE COUNT FOR OPERATOR SPEC'D MSG.  
MSG8C: .WORD EMSG8-MSG8 ;BYTE COUNT OF RECEIVE BUFFER FILL PATTERN  
DLLM1C: .WORD DLLM1E-DLLM1 ;DLL MSG 1 COUNT  
DLLM2C: .WORD DLLM2E-DLLM2 ;DLL MSG 2 COUNT  
  
:MESSAGE ADDRESS TABLE  
  
DMSGAD:  
MSG0 ;ADDRESS OF MESSAGE #0  
MSG1 ;ADDRESS OF MESSAGE #1  
MSG2 ;ADDRESS OF MESSAGE #2  
MSG3 ;ADDRESS OF MESSAGE #3  
MSG4 ;ADDRESS OF MESSAGE #4  
MSG5 ;ADDRESS OF MESSAGE #5  
MSG6 ;ADDRESS OF MESSAGE #6  
OPBUF ;ADDRESS OF OPERATOR SPEC'D MSG.  
MSG8 ;ADDRESS OF RECEIVE BUFFER FILL PATTERN  
  
MSG0: .BYTE 000 ;MESSAGE OF ALL 0'S  
EMSG0:  
MSG1: .BYTE 377 ;MESSAGE OF ALL 1'S  
EMSG1:  
MSG2: .BYTE 252 ;MESSAGE OF ALTERNATING 1'S  
EMSG2:  
MSG3: .BYTE 125 ;MESSAGE OF ALTERNATING 0'S  
EMSG3:  
MSG4: .WORD 177603,157427,031011,047321,163715,105221,143325,142304 ;'CCITT' 512-BIT (VS. 511 BITS) TEST PATTERN  
  
MSG0: .WORD 040041,014116,052606,172334,105025,123754,111337,111523  
MSG1: .WORD 030030,145064,137642,143531,063617,135075,066730,026575  
MSG2: .WORD 052012,053627,070071,151172,165044,031605,166632,016741

1867 002320 166632 016741  
1868 002324  
1869 002324  
1870  
1871 002324 077577 040444 052040  
1872 002332 042510 050440 044525  
1873 002340 045503 041040 047522  
1874 002346 047127 043040 054117  
1875 002354 045040 046525 042520  
1876 002362 020104 053117 051105  
1877 002370 052040 042510 046040  
1878 002376 055101 020131 047504  
1879 002404 027107  
1880 002406 005015 077401 077577  
1881 002414 000177  
1882 002416  
1883 002416  
1884 002416 022043 021041 023040  
1885 002424 024047 025051 026053  
1886 002432 027055 030460 031462  
1887 002440 032464 033466 034470  
1888 002446 035472 036474 037476  
1889 002454 040500 041502 042504  
1890 002462 043506 044510 045512  
1891 002470 046514 047516 050520  
1892 002476 051522 052524 053526  
1893 002504 054530 132  
1894 002507 057 056133 057135  
1895 002514 022537 000  
1896 002517  
1897 002520  
1898  
1899  
1900  
1901  
1902 002520 047045 040445  
1903 002524 000122  
1904 002646  
1905  
1906  
1907  
1908  
1909 002646 033  
1910 002647  
1911  
1912  
1913  
1914 002647 006  
1915 002650 000  
1916 002651 000  
1917 002652 000  
1918 002653 000  
1919 002654  
1920 002654 000  
1921 002655 000  
1922 002656 006

EMSG4:  
MSG5: ;'INTERPROCESSOR TEST PROGRAM'S (ITEP)'' MESSAGE  
; #1, (DP1:)  
.ASCII <177><177>/SA THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG./

EMSG5:  
MSG6: ;ALPHA-NUMERICS (OR FUTURE COMM TURNAROUND MSG)  
.ASCII /#S!'' &'()\*+,-.0123456789:;<=>?@ABCDEFGHIJKLMN OPQRSTUVWXYZ/

EMSG6:  
.EVEN

; \*\*\*\*\*  
;THESE THREE STORAGE AREAS MUST NOT BE SEPERATED !!!!

OPBFPT: .ASCII /%N%  
OPBUF: .BLKB 82. ;BUFFER FOR OPERATOR SPEC'D MESSAGES  
OPEND:

; THE ABOVE THREE LINES MUST BE KEPT TOGETHER  
; \*\*\*\*\*

MSG8: .BYTE 33 ;RECEIVE BUFFER FILL PATTERN  
EMSG8:

; DOWN-LINE-LOAD MESSAGE DEFINITIONS

DLLM1: .BYTE 6  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0

DLLM1E:  
DLLM2: .BYTE 0 ;CODE  
.BYTE 0 ;LOAD NUMBER  
.BYTE 6 ;LOAD ADDRESS LSB

```
1923 002657 000 .BYTE 0
1924 002660 006 .BYTE 6
1925 002661 000 .BYTE 0 ;LOAD ADDRESS
1926
1927 :
1928 : IMAGE DATA
1929 :
1929 002662 005037 000006 CLR @#6
1930 002666 000005 RESET
1931 002670 012706 001000 MOV #1000,SP
1932 002674 012701 177560 MOV #177560,R1 ;SET UP TTY
1933 002700 010700 MOV PC,R0 ;MAKE ADDR.PIC
1934 002702 062700 000034 ADD #<MSG-.>,R0 ;ADDRESS MSG.
1935 002706 105761 000004 1$: TSTB 4(R1) ;TTY READY?
1936 002712 100375 BPL 1$ ;WAIT TIL YES
1937 002714 112061 000006 MOV#B (R0)+,6(R1) ;TYPE A CHAR
1938 002720 001372 BNE 1$ ;KEEP GOING
1939 002722 012737 000026 000024 MOV #26,@#24 ;SET UP POWER FAIL
1940 002730 005037 000026 CLR @#26 ;MAKE SURE T BIT CLAER
1941 002734 000777 BR ;JUMP ON YOURSELF
1942 002736 006412 047502 052117 MSG: .ASCII <12><15>/BOOT MESSAGE WAS RECEIVED SUCCESSFULLY -END OF TEST!!/
1943 002744 046440 051505 040523
1944 002752 042507 053440 051501
1945 002760 051040 041505 044505
1946 002766 042526 020104 052523
1947 002774 041503 051505 043123
1948 003002 046125 054514 026440
1949 003010 047105 020104 043117
1950 003016 052040 051505 020524
1951 003024 041
1952 003025 012 027015 027056 .ASCIZ <12><15>/....RELOAD PROGRAM..../
1953 003032 051056 046105 040517
1954 003040 020104 051120 043517
1955 003046 040522 027115 027056
1956 003054 000056
1957 003056 006 .BYTE 6
1958 003057 000 .BYTE 0
1959 003060 000 .BYTE 0
1960 003061 000 .BYTE 0
1961 003062 DLLM2E:
1962
1963 .EVEN
1964
```

```

1965 ;COMMAND LINE BUFFER, DATA LOCATIONS AND MESSAGES FOR ACTION ROUTINES
1966
1967 003062 000122 CMDBUF: .BLKB 82. ;BUFFER FOR OPERATOR COMMANDS
1968 003204 000000 KEYWD1: .WORD 0 ;THIS LOC WILL =1 IF CLEAR TYPED, 2 FOR SHOW,
1969 ; A 4 IF RUN WAS TYPED, 5 IF HELP WAS TYPED
1970 003206 000000 QUALFG: .WORD 0 ;THIS LOC HOLDS QUALIFIER VALUE (SIZE OR COPY)
1971 003210 000000 QUALVL: .WORD 0
1972 003212 012525 HLPTAB: .WORD HLP1
1973 003214 012540 .WORD HLP2
1974 003216 012646 .WORD HLP3
1975 003220 012733 .WORD HLP4
1976 003222 013012 .WORD HLP4A
1977 003224 013070 .WORD HLP5
1978 003226 013152 .WORD HLP6
1979 003230
1980 003230 013305 013314 013321 HLPEND:
1981 003236 013326 013333 013341 SHTYTB: .WORD SHTYP0,SHTYP1,SHTYP2,SHTYP3,SHTYP4,SHTYP5,SHTYP6,SHTYP7
1982 003244 013346 013354
1983
1984 ; THE LIST OF BYTES BELOW ARE THE FIRST BYTES OF THE PREDEFINED MESSAGES
1985 ; USED TO 'SHOW' THE TRANSMIT AND COMPARE BUFFER CONTENTS.
1986
1987 003250 000 377 252 SHTAB: .BYTE 0,377,252,125,203,177,043
1988 003253 125 203 177
1989 003256 043
1990 003257
1991 003260 SHTEND:
1992 .EVEN
1993 003260 013365 MODES: .WORD M00 ;ADDRESSES OF MODE TYPES IN ASCII
1994 003262 013375 .WORD M01
1995 003264 013406 .WORD M02
1996 003266 013416 .WORD M03
1997 003270 013425 .WORD M04
1998 003272 013442 .WORD M05
1999 003274 013447 .WORD M06
2000
2001 003276 013456 LOOPS: .WORD LP0 ;ADDRESSES OF LOOP TYPES IN ASCII
2002 003300 013466 .WORD LP1
2003 003302 013477 .WORD LP2
2004 003304 013505 .WORD LP3
2005 003306 013520 .WORD LP4
2006
2007 ;COMMAND LINE TRAVERSE LOCATIONS (USED BY 'P$TRV')
2008
2009 003310 000000 P$BUFA: .WORD 0 ;LOC. TO HOLD ADDR. OF CMD LINE BUFFER
2010 003312 000000 P$TREE: .WORD 0 ;LOC. TO HOLD ADDR. OF PARSING TREE
2011 003314 000000 P$ACT: .WORD 0 ;LOC. TO HOLD ADDR. OF ACTION ROUTINE
2012 003316 000000 P$CNT: .WORD 0 ;LOC. TO BE A COUNTER LOCATION
2013 003320 000000 P$NUM: .WORD 0 ;LOC. TO HOLD NUMERIC VALUE FROM PARSE
2014 003322 000000 P$RADX: .WORD 0 ;LOC. TO HOLD RADIX USED(LO) AND +/--(HI BYTE)
2015 003324 000 P$NUF: .BYTE 0 ;RETURN =0 IF ENOUGH OF COMMAND FOUND
2016 003325 000 P$GDBD: .BYTE 0 ;RETURN CODE 0 IF NO ERROR FOUND
2017
  
```

2018			.SBTTL	MESSAGE BUFFERS AND POINTER TABLES
2019				
2020	003326	001000	TXBUF: .BLKB	BUFLIM :TRANSMITTER BUFFERS
2021	004326	001000	RXBUF: .BLKB	BUFLIM :RECEIVER BUFFERS
2022	005326	001000	CMPBUF: .BLKB	BUFLIM :COMPARISON BUFFERS
2023	006326	000264	PTRTAB: .BLKW	180. :TABLE FOR MESSAGE ADDRS. & BYTE COUNTS
2024	007076		PTREND:	: END OF MSG. PTR. TABLE
2025				
2026	007076	000000	RXPTR: .WORD	0 :RECEIVER MESSAGE POINTER
2027	007100	000000	TXPTR: .WORD	0 :TRANSMITTER BUFFER POINTER
2028	007102	000000	CMPPTR: .WORD	0 :COMPARISON BUFFER POINTER
2029	007104	000000	CMPTOT: .WORD	0 :CMP MSG TOTAL
2030	007106	000000	CTOTCC: .WORD	0 :COMPARE BUFFER CHAR. COUNT
2031	007110	000000	CCURAD: .WORD	0 :CURRENT ADDR OF CMP BUFF TO ADD AT
2032				
2033	007112	000000	DVTXA: .WORD	0 :DEVICE TX ADDR
2034	007114	000000	DVTCC: .WORD	0 :DEVICE TX CHAR COUNT
2035	007116	000000	DVTCT: .WORD	0 :DEVICE TX MESSAGE COUNT
2036	007120	000000	TXMTOT: .WORD	0 :TX MSG TOTAL
2037	007122	000000	TTOTCC: .WORD	0 :TX BUFFER CHAR. COUNT
2038	007124	000000	TCURAD: .WORD	0 :CURRENT ADDR. OF TX BUFF TO ADD AT
2039				
2040	007126	000000	DVRXA: .WORD	0 :DEVICE RX ADDR
2041	007130	000000	DVRCC: .WORD	0 :DEVICE RX CHAR COUNT
2042	007132	000000	DVRCT: .WORD	0 :DEVICE RX MESSAGE COUNT
2043	007134	000000	RXMTOT: .WORD	0 :RX MSG TOTAL
2044				
2045	007136	000000	LNCNT: .WORD	0 :NUMBER OF OPERATOR AWAKE MSGS
2046	007140	000000	NOBUF: .WORD	0 :NUMBER OF NO BUFFS
2047	007142	000000	PSCNT: .WORD	0 :PASS COUNTER
2048	007144	000000	ERRCNT: .WORD	0 :ERROR COUNTER
2049	007146	000000	STADD: .WORD	0 :START ADDR.
2050	007150	000000	ENADD: .WORD	0 :END ADDR. FOR DUMP
2051	007152	000000	BYTBIT: .WORD	0 :BYTE BIT FOR DUMP ROUTINE
2052				
2053			:OTHER MESSAGE RELATED STORAGE LOCATIONS	
2054				
2055	007154	000000	MSGTYP: .WORD	0 :TYPE OF DATA 0=0'S,1=1'S,2=10'S,3=01'S
2056				:4=CCITT,5=QUICK FOX,6=ALPHA/NUM,7=OPER
2057	007156	000000	CURCC: .WORD	0 :TX/RX/CMP CHAR COUNT
2058	007160	000000	CPTRR: .WORD	0 :CURRENT RX POINTER
2059	007162	000000	CPTR: .WORD	0 :CURRENT POINTER
2060	007164	000000	CURADD: .WORD	0 :CURRENT TX/RX/CMP START ADDD
2061	007166	000000	TOTCC: .WORD	0 :TOTAL CHAR COUNT NOT MORE THEN 'BUFLIM'
2062	007170	000000	OFSET: .WORD	0 :OFFSET COUNT
2063	007172	000000	TEMP: .WORD	0 :TEMPORARY LOCATIONS (USED A LOT)
2064	007174	000000	TEMP1: .WORD	0
2065	007176	000000	TEMP2: .WORD	0
2066	007200	000000	TEMP3: .WORD	0
2067	007202	000000	TEMP4: .WORD	0
2068	007204	000000	CONOTM: .WORD	0 :CONTROL OUT ERROR MSG. ADDRESS
2069	007206	000000	CONTIN: .WORD	0 :WORD FOR CONTORL IN
2070	007210	000	GOOD: .BYTE	0 :BYTE TO HOLD EXPECTED MESSAGE DATA BYTE FOR ERR REPORT
2071	007211	000	BAD: .BYTE	0 :BYTE TO HOLD RECEIVED MESSAGE DATA BYTE FOR ERR REPORT
2072				

```

2073                ;MORE INDEPENDENT CODE STORAGE LOCATIONS
2074
2075 007212 000000  LOGUNT: .WORD 0      ;LOC. TO HOLD LOGICAL UNIT NUMBER
2076 007214 000000  PCADD:  .WORD 0      ;LOC. HOLD PC OF CALLIN ROUTINE
2077 007216 000000  RESFLG: .WORD 0      ;LOC TO HOLD FLAG (-1) THAT A RESTART WAS GIVEN
2078 007220 000000  MODTYP: .WORD 0      ;DCLT MODE OF OPERATION TYPE
2079                ; (0=REC-ONLY, 1=TX-ONLY, 2=PASSIVE-LOOPBK,
2080                ; 3=ACTIVE-LOOPBK, 4=DOWN L.L., 5=TALK, 6=LISTEN)
2081 007222 000000  MLTYP:  .WORD 0      ;MAINTENANCE LOOP TYPE (0=NONE, 1=INTERNAL TTL,
2082                ; 2=CABLE, 3=MODEM-ANALOG LOOPBK (LOCAL),
2083                ; 4=MODEM-DIGITAL LOOPBK (REMOTE), 5=MOP)
2084 007224 000000  FHDPLX: .WORD 0      ;FULL OR HALF DUPLEX FLAG (1=FULL FROM P-TABLE)
2085 007226 000002  PARAM:  .WORD 2      ;PROGRAM PARAMETERS
2086                ; BIT0= STATUS MSGS TO OPR PRINTED (1=YES)
2087                ; BIT1= DATA CHECKING DONE ON RCV'D MSGS (1=YES)
2088                ; BIT2= ECHO (TRANSMIT) RCV'D MSG.(PASSIVE)(1=YES)
2089                ; BIT3= SPARE
2090                ; BIT4= CRC CALC./CHECK DONE (1=YES)
2091                ; BIT5= PROTOCOL EMULATION (1=YES)
2092                ; BIT6= SPARE
2093 007230 000000  RPASS:  .WORD 0      ;PASS NUMBER FROM RUN COMMAND
2094 007232 000000  FLAG:   .WORD 0      ;DEVICE FLAG WORD
2095
2096                ;MODE DISPATCH TABLE
2097 007234 031014  MODE:   .WORD RYONLY ;RX ONLY DISPATCH
2098 007236 031046                .WORD TXONLY  ;TX ONLY DISPATCH
2099 007240 031106                .WORD PLCK   ;PASSIVE LOOP BACK DISP
2100 007242 031142                .WORD ALCK   ;ACTIVE LOOP BACK DISP
2101 007244 032262                .WORD DLL    ;DOWN LINE LOAD DISP
2102 007246 032750                .WORD TALCK  ;TALK MODE DISPATCH
2103 007250 033170                .WORD LISCK  ;LISTEN MODE DISPATCH
2104
2105
2106                .SBTTL      CLOCK TABLES, EVENT LOG AND POINTERS
2107 007252 000000  CLKCSR: .WORD 0      ;CLOCK CSR ADDRESS
2108 007254 000000  CLKBR:  .WORD 0      ;CLOCK INTERRUPT LEVEL
2109 007256 000000  CLKVEC: .WORD 0      ;CLOCK INTERRUPT VECTOR
2110 007260 000074  CLKHZ:  .WORD 0      ;CLOCK'S HERTZ RATE
2111 007262 000000  CLKEN:  .WORD 0      ;CLOCK'S CSR VALUE TO INTRPT. ENABLE IT
2112
2113 007264 000000  TIMMIN: .WORD 0      ;PLACE TO KEEP TIME-SINCE-START
2114 007266 000000  TIMSEC: .WORD 0
2115 007270 000000  TIMTCK: .WORD 0      ;PLACE TO KEEP # OF TICKS/SEC
2116
2117 007272 000000  TIMER1: .WORD 0      ;EVENT TIMER #1 (TICKS)
2118 007274 000000  TIMER2: .WORD 0      ;EVENT TIMER #2 (TICKS)
2119 007276 000000  TIMERS: .WORD 0      ;EVENT TIMER #3 (SECONDS)
2120

```

```
2121 ;EVENT LOG TABLE AND ITS NEXT ENTRY POINTER
2122 007300 007302 EVTPTR: .WORD EVTLOG ;POINTER TO NEXT FREE SPACE IN EVENT LOG
2123 007302 000341 EVTLOG: .BLKW 225. ;EVENT LOG BUFFER
2124 010204 000001 EVTEND: .BLKW 1. ;APPROXIMATE END OF EVENT TABLE (ALLOWS CIRCULAR QUE)
2125
2126 .SBTTL MODEM DATA SECTION
2127
2128 010206 000000 MODS: .WORD 0 ;MODEM STATUS
2129
2130 ;TABLE OF MODEM SIGNAL BIT DEFINITIONS
2131
2132 010210 000004 MOBITS: .WORD CTS ;CLEAR TO SEND (CIRCUIT CB)
2133 010212 000010 .WORD DSR ;DATA SET READY (CIRCUIT CC)
2134 010214 000001 .WORD DCD ;DATA CARRIER DETECT (CIRCUIT CF)
2135 010216 000040 .WORD RTS ;REQUEST TO SEND (CIRCUIT CA)
2136 010220 000200 .WORD RI ;RING INDICATOR (CIRCUIT CE)
2137 010222 040000 .WORD SQD ;SIGNAL QUALITY DETECT (CIRCUIT CG)
2138 010224 001000 .WORD TM ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)
2139 010226
2140
2141 ;TABLE OF ADDRESSES OF MODEM SIGNAL MESSAGE POSITIONS
2142
2143 010226 016004 MOMSGS: .WORD EVMCTS ;CLEAR TO SEND (CIRCUIT CB)
2144 010230 016010 .WORD EVMDSR ;DATA SET READY (CIRCUIT CC)
2145 010232 016014 .WORD EVMDCD ;DATA CARRIER DETECT (CIRCUIT CF)
2146 010234 016020 .WORD EVMRTS ;REQUEST TO SEND (CIRCUIT CA)
2147 010236 016024 .WORD EVMRI ;RING INDICATOR (CIRCUIT CE)
2148 010240 016030 .WORD EVMSQD ;SIGNAL QUALITY DETECT (CIRCUIT CG)
2149 010242 016034 .WORD EVMTM ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)
2150
2151 ;TABLE OF ADDRESSES OF EVENT DESCRIPTION MESSAGES
2152 ; ORDER CORRESPONDS TO MESSAGE TYPE VALUES
2153
2154 010244 014430 EVTLS: .WORD EDTXQ ;TRANSMIT MESSAGE QUEUED
2155 010246 014454 .WORD EDTXC ;TRANSMIT OF MESSAGE COMPLETE
2156 010250 014503 .WORD EDRXQ ;RECEIVE MESSAGE SPACE QUEUED
2157 010252 014530 .WORD EDRXC ;MESSAGE RECEIVED - RECEIVE COMPLETE
2158 010254 014556 .WORD EDDER ;DEVICE INFORMATION
2159 010256 014623 .WORD EDDVI ;DEVICE INITIALIZE STARTED
2160 010260 014573 .WORD EDDCK ;DATA COMPARISON DONE
2161 010262 013456 .WORD LPO ;NULL STRING
2162 010264 014651 .WORD EDDLE ;DATA COMPARE LENGTH ERROR
2163 010266 014706 .WORD EDDDE ;DATA COMPARE DATA ERROR
2164 010270 014741 .WORD EDEOP ;END OF PASS
2165
2166 ;LOCATIONS USED DURING EVENT REPORTING
2167
2168 010272 000000 EVTSEC: .WORD 0 ;TEMPORARY LOCS TO KEEP EVENT TIME WHILE REPORTING
2169 010274 000000 EVTMIN: .WORD 0
2170 010276 000000 EVTTC: .WORD 0
2171 010300 000000 EVTADD: .WORD 0 ;TEMP. LOC. TO HOLD ADDRESS DURING EVENT REPORTING
2172 010302 000000 EVTBC: .WORD 0 ; " " " " BYTE COUNT " " " "
2173 010304 000000 EVTTP: .WORD 0 ; " " " " OTHER DATA " " " "
2174
2175 ;REPORT CODING DISPATCH TABLE
2176
```

```
2177 010306 021702 RPTDSP: .WORD RPTTXQ ; TRANSMIT QUEUED ENTRY DECODING
2178 010310 021702 .WORD RPTTXQ ; TRANSMIT COMPLETE ENTRY DECODING
2179 010312 021702 .WORD RPTTXQ ; RECEIVER QUEUED ENTRY DECODING
2180 010314 021702 .WORD RPTTXQ ; RECEIVER COMPLETE ENTRY DECODING
2181 010316 021754 .WORD RPTDER ; DEVICE ERROR ENTRY DECODING
2182 010320 022050 .WORD RPTDVI ; DEVICE INIT ENTRY DECODING
2183 010322 022244 .WORD RPTDCK ; DATA COMPARISON ENTRY DECODING
2184 010324 021530 .WORD RPT ; PLACE HOLDER
2185 010326 022244 .WORD RPTDLE ; DATA COMPARISON LENGH ERROR
2186 010330 022170 .WORD RPTDDE ; DATA COMPARISON DATA ERROR
2187 010332 022114 .WORD RPTEOP ; END OF PASS
2188
2189
2190 010334 000000 DEV1: .WORD 0 ; TEMP LOCS TO HOLD DATA FOR EVENT REPORTING
2191 010336 000000 DEV2: .WORD 0 ; AND SHOW MODE,... SUBROUTINE
2192 010340 000000 DEV3: .WORD 0
2193 010342 000000 DEV4: .WORD 0
2194
```



2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
2234  
2235  
2236  
2237  
2238  
2239  
2240  
2241  
2242  
2243  
2244  
2245  
2246  
2247  
2248  
2249  
2250

010344  
  
  
010344  
010350  
010354  
010356  
010372  
010374  
010410  
010412  
010424  
010430  
010444  
010450  
010464  
010470  
010474  
010506  
010512  
010524  
010530  
  
  
010532  
010536  
010552  
010556  
010574  
010600  
010616  
010622  
010640  
010644  
010662  
010666  
010712  
010716  
010722  
010740

```

.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
:-----:
: ! ASCIIZ MATCH STRING !       ONLY IF 'ASCII' ARGUMENT DEFINED
:-----:
:          (.EVEN) !
:-----:

CLITRE:

: FIRST KEYWORD
N10$: CLI CLISPA,0,N10$          : SKIP ANY LEADIN SPACES
N42$: CLI <'?'>,HLP,N42$        : IS THE FIRST NON-SP CHAR A '?'
N43$: CLI CLISTR,HLP,N43$,<'HELP'> : IF YES DO 'HLP' AND EXIT
N45$: CLI CLISTR,PRNT,N45$,<'PRINT'> : ELSE, IS FIRST WORD A 'HELP'
N46$: CLI CLISTR,RUN,N46$,<'RUN'>   : IF YES DO 'HLP' AND EXIT
N40$: CLI CLISTR,CLEAR,N20$,<'CLEAR'> : ELSE, IS FIRST WORD A 'PRINT'
N20$: CLI <'S'>,NOTNUF,N30$      : IF YES DO 'PRINT' AND EXIT
N25$: CLI CLISTR,SHOW,N25$,<'HOW'>  : ELSE, IS FIRST WORD A 'RUN'
N30$: CLI CLISTR,ET,N30$,<'ET'>    : IF YES DO 'RUN' & GOTO N80$
      CLI CLISTR,ERR,N30$,<'ERR'>  : ELSE, IS FIRST WORD A 'DUMP'
      CLI CLISTR,SET,N30$,<'SET'>   : IF YES GOTO N80$
      CLI CLISTR,ILL,N30$,<'ILL'>  : ELSE, IS FIRST WORD A 'CLEAR'
      CLI CLISTR,MODE,N30$,<'MODE'> : IF YES DO 'CLR' & GOTO N100$
      CLI CLISTR,ACT,N30$,<'ACTIVE'> : ELSE, IS FIRST CHAR. A 'S'
      CLI CLISTR,PAS,N30$,<'PASSIVE'> : IF YES IS REST OF WORD 'HOW'
      CLI CLISTR,REC,N30$,<'RECEIVE'> : IF YES, DO 'SHOW',BR N100$
      CLI CLISTR,LIS,N30$,<'LISTEN'> : ELSE, IS REST OF WORD 'ET'
      CLI CLISTR,DOW,N30$,<'DOWNLINELOAD'> : IF YES, DO 'SET', BR N110$
      CLI CLISTR,TRM,N30$,<'TRANSMIT'> : OTHERWISE 'ILL CMD' - EXIT
      CLI CLISTR,ERR,N30$,<'ERR'>

: SECOND KEYWORD (MODE=) FOR RUN COMMAND
N80$: CLI CLISPA,0,N30$
N81$: CLI CLISTR,NOTNUF,N30$,<'MODE'>
      CLI <'='>,0,N30$
      CLI CLISTR,ATVMOD,N82$,<'ACTIVE'>
      CLI CLISTR,PASMOD,N83$,<'PASSIVE'>
      CLI CLISTR,RECMOD,N84$,<'RECEIVE'>
      CLI CLISTR,LISMOD,N85$,<'LISTEN'>
      CLI CLISTR,DLLOD,N86$,<'DOWNLINELOAD'>
      CLI <'T'>,0,N30$
      CLI CLISTR,TRAMOD,N87$,<'RANSMIT'>
      CLI CLISTR,ERR,N87$,<'ERR'>
      CLI CLISTR,MODE,N87$,<'MODE'>
      CLI CLISTR,ACT,N87$,<'ACTIVE'>
      CLI CLISTR,PAS,N87$,<'PASSIVE'>
      CLI CLISTR,REC,N87$,<'RECEIVE'>
      CLI CLISTR,LIS,N87$,<'LISTEN'>
      CLI CLISTR,DOW,N87$,<'DOWNLINELOAD'>
      CLI CLISTR,TRM,N87$,<'TRANSMIT'>
      CLI CLISTR,ERR,N87$,<'ERR'>

: SKIP LEADING SPS, IF NONE-ERR
: IS NEXT WORD 'MODE='
: IF NO, IT'S WRONG -ERR -EXIT
: IS NEXT WORD 'ACTIVE'
: IF YES, DO 'ACTIVE',BR N115$
: IS NEXT WORD 'PASSIVE'
: IF YES, DO 'PASSIVE',BR N115$
: IS NEXT WORD 'RECEIVE'
: IF YES, DO 'RECVE',BR N115$
: IS NEXT WORD 'LISTEN'
: IF YES, DO 'LISTEN',BR N115$
: IS NEXT WORD 'DOW'
: IF YES, DO 'DOWLL',BR N115$
: IS NEXT CHAR A 'T'
: IS REST OF WORD 'TRANSMIT'
: IF YES, DO 'TRANSM',BR N115$

```

```
2251 010744 N87$: CLI CLISTR,TALMOD,N30$,<'ALK'> : IS REST OF WORD 'ALK'
2252 010756 CLI CLIBR,0,N115$ : IF YES, DO 'TALK',BR N115$
2253 : IF NO, ERROR - EXIT
2254
2255 ;SECOND KEYWORD (FOR CLEAR OR SHOW)
2256 010762 N100$: CLI CLISPA,0,N30$ ;SKIP LEADING SPACES, NONE=ERR
2257 010766 N102$: CLI CLISTR,CSHEXP,N104$,<'EXPECTBUFF'> ;IS NEXT WORD 'EXPE...'
2258 011010 CLI CLIEXI,0 ; IF YES, DO CLR-EXP,EXIT
2259 011012 N104$: CLI CLISTR,CSHTRN,N30$,<'TRANSMITBUFF'> ;IS NEXT WORD 'TRANS...'
2260 011036 CLI CLIEXI,0 ; IF YES, DO CLR-TRN,EXIT
2261 :IF NO - ERROR - EXIT
2262
2263
2264 ;SECOND KEYWORD (FOR SET)
2265 011040 N110$: CLI CLISPA,0,N30$
2266 011044 N111$: CLI CLISTR,SETEXP,N112$,<'EXPECTMSG'>
2267 011064 CLI CLIBR,0,N120$
2268 011070 N112$: CLI CLISTR,SETTRN,N30$,<'TRANSMITMSG'>
2269 011112 CLI CLIBR,0,N120$
2270
2271 ;GET ADDRESSES FOR DUMP COMMAND
2272 011116 N50$: CLI CLIALP,0,N51$
2273 011122 N51$: CLI CLISPA,0,N52$
2274 011126 N52$: CLI CLIOCT,DMP$ ,N30$
2275 011132 CLI <'>,NOTNUF,N125$
2276 011136 CLI CLIOCT,DMP$,N30$
2277 011142 CLI <'>,NOTNUF,N125$
2278 011146 CLI <'B>,DMPQ,N30$
2279 011152 CLI CLIBR,0,N125$
2280
2281 ;QUALIFIERS FOR THE RUN COMMAND
2282 011156 N115$: CLI CLIALP,0,N114$
2283 011162 N114$: CLI <'>,NOTNUF,N125$
2284 011166 CLI CLISTR,NO,N116$,<'NO'>
2285 011200 N116$: CLI <'C>,0,N117$
2286 011204 CLI CLISTR,CHECK,N117$,<'HECK'>
2287 011220 CLI CLIBR,0,N115$
2288
2289
2290 ;N113$: CLI CLISTR,CRC,N30$,<'RC16'>
2291 : CLI CLIBR,0,N115$
2292
2293 011224 N117$: CLI CLISTR,STATUS,N118$,<'STATUS'>
2294 011242 CLI CLIBR,0,N115$
2295 011246 N118$: CLI CLISTR,ECHO,N130$,<'ECHO'>
2296 011262 CLI CLIBR,0,N115$
2297
2298
2299 011266 N130$: CLI CLISTR,0,N131$,<'PASS'>
2300 011302 CLI CLIBR,0,N150$
2301 011306 N131$: CLI CLISTR,0,N30$,<'LOOP'>
2302 011322 CLI CLIBR,0,N140$
2303
2304 ;GET MESSAGE TYPE FOR SET MESSAGE COMMANDS
2305 011326 N120$: CLI <'>,0,N30$
2306
```

```
2307 ; LOOK FOR DEFAULT MESSAGE NAME
2308 011332 N60$: CLI CLISTR,MSG1,N61$,<'ONES'>
2309 011346 CLI CLIBR,0,N121$
2310 011352 N61$: CLI CLISTR,MSG0,N62$,<'ZER0ES'>
2311 011370 CLI CLIBR,0,N121$
2312 011374 N62$: CLI CLISTR,MSG2,N63$,<'1ALT'>
2313 011410 CLI CLIBR,0,N121$
2314 011414 N63$: CLI CLISTR,MSG3,N64$,<'0ALT'>
2315 011430 CLI CLIBR,0,N121$
2316 011434 N64$: CLI CLISTR,MSG5,N65$,<'ITEP'>
2317 011450 CLI CLIBR,0,N121$
2318 011454 N65$: CLI CLISTR,MSG4,N66$,<'CCITT'>
2319 011470 CLI CLIBR,0,N121$
2320 011474 N66$: CLI CLISTR,MSG6,N67$,<'ALPHA'>
2321 011510 CLI CLIBR,0,N121$
2322
2323 ; LOOK FOR QUOTED MESSAGE
2324 011514 N67$: CLI <'>,OPRMSG,N30$
2325 011520 N70$: CLI <'>,ENDQ0,N71$
2326 011524 CLI CLIBR,0,N121$
2327 011530 N71$: CLI CLISPA,0,N72$
2328 011534 N72$: CLI CLIALN,0,N73$ ;ONLY A-Z,SP,TAB, OR 0-9 BETWEEN ''S
2329 011540 CLI CLIBR,0,N70$ ;PRINT ERROR IF NONE LEGAL CHAR FOR ''S
2330 011544 N73$: CLI CLIERR,BADCHR
2331
2332 ;GET QUALIFIERS (SIZE OR COPY) FOR SET MESSAGE COMMANDS
2333 011546 N121$: CLI CLIALP,0,N123$
2334 011552 N123$: CLI <'/'>,NOTNUF,N125$
2335 011556 CLI CLISTR,SIZE,N122$,<'SIZE'>
2336 011572 CLI CLIBR,0,N126$
2337 011576 N122$: CLI CLISTR,QCOPY,N30$,<'COPY'>
2338 011612 CLI CLIBR,0,N126$
2339
2340 ;NUMER FOR SIZE OR COPY
2341 011616 N126$: CLI <'=>,0,N30$
2342 011622 CLI CLIDEC,NUM,N30$
2343 011626 CLI CLIBR,0,N121$
2344
2345 ;GET MAINTENANCE LOOP TYPE FOR RUN 'LOOP' QUALIFIER
2346 011632 N140$: CLI <'=>,0,N30$
2347
2348
2349 011636 N141$: CLI CLISTR,TTLLOP,N142$,<'INTERNAL TTL'>
2350 011660 CLI CLIBR,0,N115$
2351 011664 N142$: CLI CLISTR,CBLLOP,N143$,<'CABLE'>
2352 011700 CLI CLIBR,0,N115$
2353 011704 N143$: CLI CLISTR,LMDLOP,N144$,<'LOCALMODEM'>
2354 011726 CLI CLIBR,0,N115$
2355 011732 N144$: CLI CLISTR,RMDLOP,N30$,<'REMOTEMODEM'>
2356 011754 CLI CLIBR,0,N115$
2357
2358 ;GET LINE NUMBER FOR 'PASS' RUN QUALIFIER
2359 011760 N150$: CLI <'=>,0,N30$
2360 011764 CLI CLIDEC,PASC,N30$
2361 011770 CLI CLIBR,0,N115$
2362
```

2363  
2364  
2365  
2366 011774  
2367

:END-OF-LINE  
N125\$: CLI CLIEXI,0

```
2368
2369
2370 ;DEVICE DEPENDENT STORAGE LOCATIONS FOR
2371 ; CURRENT DEVICE PARAMTERS
2372
2373 011776 SEL0:
2374 011776 000000 BSEL0: .WORD 0 ;ADDRESSES OF REGISTERS SEL0 THRU BSEL7
2375 012000 000000 BSEL1: .WORD 0
2376 012002 SEL2:
2377 012002 000000 BSEL2: .WORD 0
2378 012004 000000 BSEL3: .WORD 0
2379 012006 SEL4:
2380 012006 000000 BSEL4: .WORD 0
2381 012010 000000 BSEL5: .WORD 0
2382 012012 SEL6:
2383 012012 000000 BSEL6: .WORD 0
2384 012014 000000 BSEL7: .WORD 0
2385
2386
2387 012016 000000 INVEC: .WORD 0 ;INPUT INTERRUPT VECTOR ADDRESS
2388 012020 000000 OUTVEC: .WORD 0 ;OUTPUT INTERRUPT VECTOR ADDRESS
2389 012022 000000 INTPRI: .WORD 0 ;INTERRUPT PRIORITY
2390 012024 000000 OPTYP: .WORD 0 ;DEVICE OPTION TYPE(0=DMC,5=DMR-DMC MODE
2391 ;7=DMR).
2392
2393
2394
2395
2396 ; ERRTBL
```

2397  
2398  
2399  
2400  
2401  
2402  
2403  
2404  
2405  
2406  
2407  
2408  
2409  
2410  
2411  
2412  
2413  
2414  
2415  
2416  
2417  
2418  
2419  
2420  
2421  
2422  
2423  
2424  
2425  
2426  
2427  
2428  
2429  
2430  
2431  
2432  
2433  
2434

.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 <DMR,DMC-11>

012026  
012026  
012026 046504 026122 046504  
012034 026503 030461 000  
012042

LSDVTYP::  
.ASCIZ /DMR,DMC-11/  
.EVEN

.SBTTL PROGRAM IDENTIFICATION  
: TEST DESCRIPTION

012042  
012042  
012042 055103 046103 040513  
012050 020060 046504 026122  
012056 042040 041515 030455  
012064 020061 040504 040524  
012072 041440 046517 027115  
012100 046040 047111 020113  
012106 042524 052123 000  
012114

<CZCLKAO DMR, DMC-11 DATA COMM. LINK TEST>  
L\$DESC::  
.ASCIZ /CZCLKAO DMR, DM

.EVEN .EVEN

```
2435 .SBTTL GLOBAL FORMAT STATEMENTS, MESSAGES, AND ASCII INFO
2436
2437
2438 012114 041504 052114 000076 CLISPM: .ASCIZ /DCLT>/
2439 012122 047045 040445 044477 CLIERM: .ASCIZ /%N%A?ILL CMD-BAD SYNTAX?/
2440 012130 046114 041440 042115
2441 012136 041055 042101 051440
2442 012144 047131 054124 000077
2443 012152 047045 040445 044477 CLINUF: .ASCIZ /%N%A?INCMPLTE CMD?/
2444 012160 041516 050115 052114
2445 012166 020105 046503 037504
2446 012174 000
2447 012175 045 022516 037501 CLINBG: .ASCIZ /%N%A?NUM TOO BIG?/
2448 012202 052516 020115 047524
2449 012210 020117 044502 037507
2450 012216 000
2451 012217 045 022516 037501 CLIBRX: .ASCIZ /%N%A?BAD RADIX?/
2452 012224 040502 020104 040522
2453 012232 044504 037530 000
2454 012237 045 022516 037501 CLIBDL: .ASCIZ /%N%A?'LOOP' VALID ONLY IN ACTIVE?/
2455 012244 046042 047517 021120
2456 012252 053040 046101 042111
2457 012260 047440 046116 020131
2458 012266 047111 040440 052103
2459 012274 053111 037505 000
2460 012301 045 022516 037501 CLINPS: .ASCIZ /%N%A?'ECHO' VALID ONLY IN PASSIVE?/
2461 012306 042442 044103 021117
2462 012314 053040 046101 042111
2463 012322 047440 046116 020131
2464 012330 047111 050040 051501
2465 012336 044523 042526 000077
2466 012344 047045 040445 044477 CLIBCR: .ASCIZ /%N%A?ILL CHR- 'A-Z,0-9,SP,TAB' ONLY?/
2467 012352 046114 041440 051110
2468 012360 020055 040442 055055
2469 012366 030054 034455 051454
2470 012374 026120 040524 021102
2471 012402 047440 046116 037531
2472 012410 000
2473 012411 045 022516 037501 CLISEO: .ASCIZ /%N%A?'SIZE=0' NOT VALID?/
2474 012416 051442 055111 036505
2475 012424 021060 047040 052117
2476 012432 053040 046101 042111
2477 012440 000077
2478 012442 047045 040445 044124 HLPO: .ASCIZ /%N%A?THIS IS DCLT. TYPE 'H' OR '?' FOR DETAILS/
2479 012450 051511 044440 020123
2480 012456 041504 052114 020056
2481 012464 054524 042520 021040
2482 012472 021110 047440 020122
2483 012500 037442 020042 047506
2484 012506 020122 042504 040524
2485 012514 046111 000123
2486 012520 047045 052045 000 HLPF: .ASCIZ /%N%A?/
2487 012525 104 046103 020124 HLP1: .ASCIZ /DCLT CMDS:/
2488 012532 046503 051504 000072
2489 012540 041440 042514 051101 HLP2: .ASCII / CLEAR OR SHOW EXPECTLIST OR TRANSMITLIST/<15><12>
2490 012546 047440 020122 044123
```

2491	012554	053517	020040	054105
2492	012562	042520	052103	044514
2493	012570	052123	047440	020122
2494	012576	051124	047101	046523
2495	012604	052111	044514	052123
2496	012612	005015		
2497	012614	050040	044522	052116
2498	012622	005015		
2499	012624	042040	046525	020120
2500	012632	052123	051101	026524
2501	012640	047105	027504	000102
2502	012646	051440	052105	042440
2503	012654	050130	041505	046524
2504	012662	043523	047440	020122
2505	012670	051124	047101	046523
2506	012676	052111	051515	036507
2507	012704	054524	042520	051457
2508	012712	055111	036505	020116
2509	012720	051117	027440	047503
2510	012726	054520	047075	000
2511	012733	040	020040	054524
2512	012740	042520	047475	042516
2513	012746	026123	042532	047522
2514	012754	051505	030454	046101
2515	012762	026124	040460	052114
2516	012770	044454	042524	026120
2517	012776	041503	052111	026124
2518	013004	046101	044120	000101
2519	013012	020040	020040	020040
2520	013020	047440	020122	047442
2521	013026	051120	051440	041520
2522	013034	036504	026501	026132
2523	013042	050123	052054	041101
2524	013050	030054	034455	044440
2525	013056	020116	052521	052117
2526	013064	051505	000042	
2527	013070	051040	047125	046440
2528	013076	042117	036505	052115
2529	013104	050131	046057	047517
2530	013112	036520	052114	050131
2531	013120	041457	042510	045503
2532	013126	051454	040524	052524
2533	013134	026123	041505	047510
2534	013142	050054	051501	036523
2535	013150	000116		
2536	013152	020040	046440	054524
2537	013160	036520	051124	047101
2538	013166	051054	041505	040454
2539	013174	052103	050054	051501
2540	013202	052054	046101	046054
2541	013210	051511	042054	053517
2542	013216	006516	012	
2543	013221	040	020040	052114
2544	013226	050131	044475	052116
2545	013234	041454	041101	046054
2546	013242	041517	051054	046505

.ASCII / PRINT/<15><12>  
.ASCIZ ? DUMP START-END/B?  
HLP3: .ASCIZ ? SET EXPECTMSG OR TRANSMITMSG=TYPE/SIZE=N OR /COPY=N?  
HLP4: .ASCIZ ? TYPE=ONES,ZEROES,1ALT,0ALT,ITEP,CCITT,ALPHA?  
HLP4A: .ASCIZ / OR 'OPR SPCD=A-Z,SP,TAB,0-9 IN QUOTES'  
HLP5: .ASCIZ ? RUN MODE=MTYP/LOOP=LTYP/CHECK,STATUS,ECHO,PASS=N?  
HLP6: .ASCII / MTYP=TRAN,REC,ACT,PAS,TAL,LIS,DOWN/<15><12>  
.ASCIZ / LTYP=INT,CAB,LOC,REM/



2547	013250	000					
2548							
2549	013251	045	022516	046501	SHMSG:	.ASCIZ	?%N%AMSG: TYPE=%T%A/SIZE=%D3?
2550	013256	043523	020072	054524			
2551	013264	042520	022475	022524			
2552	013272	027501	044523	042532			
2553	013300	022475	031504	000			
2554	013305	132	051105	042517	SHTYPO:	.ASCIZ	/ZEROES/
2555	013312	000123					
2556	013314	047117	051505	000	SHTYP1:	.ASCIZ	/ONES/
2557	013321	061	046101	000124	SHTYP2:	.ASCIZ	/1ALT/
2558	013326	040460	052114	000	SHTYP3:	.ASCIZ	/OALT/
2559	013333	103	044503	052124	SHTYP4:	.ASCIZ	/CCITT/
2560	013340	000					
2561	013341	111	042524	000120	SHTYP5:	.ASCIZ	/ITEP/
2562	013346	046101	044120	000101	SHTYP6:	.ASCIZ	/ALPHA/
2563	013354	050117	020122	050123	SHTYP7:	.ASCIZ	/OPR SPEC/
2564	013362	041505	000				
2565	013365	122	041505	044505	MO0:	.ASCIZ	/RECEIVE/
2566	013372	042526	000				
2567	013375	124	040522	051516	MO1:	.ASCIZ	/TRANSMIT/
2568	013402	044515	000124				
2569	013406	040520	051523	053111	MO2:	.ASCIZ	/PASSIVE/
2570	013414	000105					
2571	013416	041501	044524	042526	MO3:	.ASCIZ	/ACTIVE/
2572	013424	000					
2573	013425	104	053517	046116	MO4:	.ASCIZ	/DOWNLINELOAD/
2574	013432	047111	046105	040517			
2575	013440	000104					
2576	013442	040524	045514	000	MO5:	.ASCIZ	/TALK/
2577	013447	114	051511	042524	MO6:	.ASCIZ	/LISTEN/
2578	013454	000116					
2579	013456	000			LP0:	.ASCIZ	//
2580	013457	057	047514	050117	LP00:	.ASCIZ	?/LOOP=?
2581	013464	000075					
2582	013466	047111	042524	047122	LP1:	.ASCIZ	?INTERNAL?
2583	013474	046101	000				
2584	013477	103	041101	042514	LP2:	.ASCIZ	?CABLE?
2585	013504	000					
2586	013505	114	041517	046101	LP3:	.ASCIZ	?LOCALMODEM?
2587	013512	047515	042504	000115			
2588	013520	042522	047515	042524	LP4:	.ASCIZ	?REMOTEMODEM?
2589	013526	047515	042504	000115			
2590	013534	047516			PNST:	.ASCII	/NO/
2591	013536	052123	052101	051525	PST:	.ASCIZ	/STATUS/
2592	013544	000					
2593	013545	116	117		PNCK:	.ASCII	/NO/
2594	013547	103	042510	045503	PCK:	.ASCIZ	/CHECK/
2595	013554	000					
2596	013555	116	117		PNEC:	.ASCII	/NO/
2597	013557	105	044103	000117	PEC:	.ASCIZ	/ECHO/
2598							
2599							
2600	013564	047045	040445	044514	LISP:	.ASCIZ	/%N%ALIS>/
2601	013572	037123	000				
2602	013575	124	045514	000076	OPRMM:	.ASCIZ	/TLK>/

2603	013602	044124	051511	040440	L5060: .ASCIZ /THIS A 50. OR 60. HZ. LSI-11:/
2604	013610	032440	027060	047440	
2605	013616	020122	030066	020056	
2606	013624	055110	020056	051514	
2607	013632	026511	030461	000072	
2608					.EVEN
2609					
2610					
2611					
2612					:
2613					: FORMAT STATEMENTS USED IN PRINT CALLS
2614					:
2615					
2616	013640	047045	040445	047504	DLLCM: .ASCIZ /%N%ADOWN LINE LOAD COMPLETED SUCCESSFULLY/
2617	013646	047127	046040	047111	
2618	013654	020105	047514	042101	
2619	013662	041440	046517	046120	
2620	013670	052105	042105	051440	
2621	013676	041525	042503	051523	
2622	013704	052506	046114	000131	
2623	013712	047045	040445	040502	NOCLK: .ASCIZ /%N%ABAD CLOCK - PROGRAM WILL HANG ON 'TIMEOUT'!!!/
2624	013720	020104	046103	041517	
2625	013726	020113	020055	051120	
2626	013734	043517	040522	020115	
2627	013742	044527	046114	044040	
2628	013750	047101	020107	047117	
2629	013756	021040	044524	042515	
2630	013764	052517	021124	020441	
2631	013772	000			
2632	013773	115	054101	020056	TABEX: .ASCIZ /MAX. CHAR. MSG COUNT EXCEEDED -/
2633	014000	044103	051101	020056	
2634	014006	051515	020107	047503	
2635	014014	047125	020124	054105	
2636	014022	042503	042105	042105	
2637	014030	026440	000		
2638	014033	102	043125	042506	BUFEX: .ASCIZ /BUFFER FULL -/
2639	014040	020122	052506	046114	
2640	014046	026440	000		
2641	014051	045	022516	022524	MSGTRN: .ASCIZ /%N%T%A MSG. NOT BUILT !!!/
2642	014056	020101	051515	027107	
2643	014064	047040	052117	041040	
2644	014072	044525	052114	020440	
2645	014100	000041			
2646	014102	047045	040445	044103	MSGTRU: .ASCIZ /%N%ACHAR. COUNT EXCEEDS BUFF LIMIT - MSG TRUNCATED/
2647	014110	051101	020056	047503	
2648	014116	047125	020124	054105	
2649	014124	042503	042105	020123	
2650	014132	052502	043106	046040	
2651	014140	046511	052111	026440	
2652	014146	046440	043523	052040	
2653	014154	052522	041516	052101	
2654	014162	042105	000		
2655	014165	045	022516	032523	SHFO: .ASCIZ ?%N%SS%AMODE=%T%T%T%A/PASS=%Z5?
2656	014172	040445	047515	042504	
2657	014200	022475	022524	022524	
2658	014206	022524	027501	040520	

2659	014214	051523	022475	032532		
2660	014222	000				
2661						
2662						
2663	014223	045	022516	032523	SHF1:	.ASCIZ ?%N%S5%S5%S5%A/%T%A/%T%A/%T?
2664	014230	051445	022465	032523		
2665	014236	040445	022457	022524		
2666	014244	027501	052045	040445		
2667	014252	022457	000124			
2668						
2669	014256	051445	022465	052101	EFM2:	.ASCIZ /%S5%ATOTAL MISMATCHES IN MSG = %D5/
2670	014264	052117	046101	046440		
2671	014272	051511	040515	041524		
2672	014300	042510	020123	047111		
2673	014306	046440	043523	036440		
2674	014314	022440	032504	000		
2675	014321	045	022516	031523	PCPM:	.ASCIZ /%N%S3%ACALLED FROM PC=%06/
2676	014326	040445	040503	046114		
2677	014334	042105	043040	047522		
2678	014342	020115	041520	022475		
2679	014350	033117	000			
2680	014353	045	032523	040445	EFM11:	.ASCIZ /%S5%ACOMPARE COUNT=%D5%S3%ARECEIVE COUNT=%D5/
2681	014360	047503	050115	051101		
2682	014366	020105	047503	047125		
2683	014374	036524	042045	022465		
2684	014402	031523	040445	042522		
2685	014410	042503	053111	020105		
2686	014416	047503	047125	036524		
2687	014424	042045	000065			
2688						
2689						
2690						:EVENT DESCRIPTION MESSAGES
2691						
2692	014430	051124	047101	046523	EDTXQ:	.ASCIZ /TRANSMIT MSG QUEUED/
2693	014436	052111	046440	043523		
2694	014444	050440	042525	042525		
2695	014452	000104				
2696	014454	051124	047101	046523	EDTXC:	.ASCIZ /TRANSMIT MSG COMPLETED/
2697	014462	052111	046440	043523		
2698	014470	041440	046517	046120		
2699	014476	052105	042105	000		
2700	014503	122	041505	044505	EDRXQ:	.ASCIZ /RECEIVE SPACE QUEUED/
2701	014510	042526	051440	040520		
2702	014516	042503	050440	042525		
2703	014524	042525	000104			
2704	014530	042522	042503	053111	EDRXC:	.ASCIZ /RECEIVE MSG COMPLETED/
2705	014536	020105	051515	020107		
2706	014544	047503	050115	042514		
2707	014552	042524	000104			
2708	014556	042504	044526	042503	EDDER:	.ASCIZ /DEVICE ERROR/
2709	014564	042440	051122	051117		
2710	014572	000				
2711	014573	104	052101	020101	EDDCK:	.ASCIZ /DATA COMPARISON STARTED/
2712	014600	047503	050115	051101		
2713	014606	051511	047117	051440		
2714	014614	040524	052122	042105		



2771	015176	047045	051445	022463	EVTF2: .ASCIIZ /%N%S3%AADDR OF MSG=%06%S3%ABYTE COUNT=%D5/
2772	015204	040501	042104	020122	
2773	015212	043117	046440	043523	
2774	015220	022475	033117	051445	
2775	015226	022463	041101	052131	
2776	015234	020105	047503	047125	
2777	015242	036524	042045	000065	
2778	015250	047045	051445	022463	EVTF3: .ASCIIZ /%N%S3%T%N/
2779	015256	022524	000116		
2780	015262	051445	022463	033117	EVTF3C: .ASCIIZ /%S3%06%S3%06/
2781	015270	051445	022463	033117	
2782	015276	000			
2783	015277	045	031523	047445	EVTF3D: .ASCIIZ /%S3%06%S3%06%S3%T/
2784	015304	022466	031523	047445	
2785	015312	022466	031523	052045	
2786	015320	000			
2787	015321	045	022516	031523	EVTF4: .ASCIIZ /%N%S3%AADDR OF MSG=%06%S3%ABYTE COUNT=%D5%S3%ANO. OF CMP ERRS=%D5/
2788	015326	040445	042101	051104	
2789	015334	047440	020106	051515	
2790	015342	036507	047445	022466	
2791	015350	031523	040445	054502	
2792	015356	042524	041440	052517	
2793	015364	052116	022475	032504	
2794	015372	051445	022463	047101	
2795	015400	027117	047440	020106	
2796	015406	046503	020120	051105	
2797	015414	051522	022475	032504	
2798	015422	000			
2799	015423	045	022516	031523	EVTF4A: .ASCIIZ /%N%S3%AADDR OF MSG=%06%S3%ARX BYTES=%D5%S3%ACOMPARE BYTES=%D5/
2800	015430	040445	042101	051104	
2801	015436	047440	020106	051515	
2802	015444	036507	047445	022466	
2803	015452	031523	040445	054122	
2804	015460	041040	052131	051505	
2805	015466	022475	032504	051445	
2806	015474	022463	041501	046517	
2807	015502	040520	042522	041040	
2808	015510	052131	051505	022475	
2809	015516	032504	000		
2810	015521	045	022516	031523	EVTF4B: .ASCIIZ /%N%S3%APASS=%D5%S3%AERRORS=%D5%S3%ANOBUFFS=%D5/
2811	015526	040445	040520	051523	
2812	015534	022475	032504	051445	
2813	015542	022463	042501	051122	
2814	015550	051117	036523	042045	
2815	015556	022465	031523	040445	
2816	015564	047516	052502	043106	
2817	015572	036523	042045	000065	
2818	015600	051445	022465	041101	EVTF5A: .ASCIIZ /%S5%ABYTE # IN MSG.=%D5%S3%AEXPTD=%03%S3%ARECVD=%03/
2819	015606	052131	020105	020043	
2820	015614	047111	046440	043523	
2821	015622	036456	042045	022465	
2822	015630	031523	040445	054105	
2823	015636	052120	036504	047445	
2824	015644	022463	031523	040445	
2825	015652	042522	053103	036504	
2826	015660	047445	000063		

```
2827
2828 015664 047045 051445 022471 EVMOCG: .ASCIZ /%N%S9%ACHANGED TO:/
2829 015672 041501 040510 043516
2830 015700 042105 052040 035117
2831 015706 000
2832
2833 ; *****
2834 ;DO NOT SEPERATE THE NEXT LIST OF MESSAGES - MODEM SIGNAL HEADER AND REPORT
2835
2836 015707 045 022516 034123 EVMOH: .ASCIZ /%N%S8%MODEM STATUS: CTS DSR DCD RTS RI SQD TM/
2837 015714 040445 047515 042504
2838 015722 020115 052123 052101
2839 015730 051525 020072 052103
2840 015736 020123 051504 020122
2841 015744 041504 020104 052122
2842 015752 020123 044522 020040
2843 015760 050523 020104 046524
2844 015766 000
2845 015767 045 022516 034523 EVMOST: .ASCII /%N%S9%S9%S5%A/
2846 015774 051445 022471 032523
2847 016002 040445
2848 016004 130 040 040 EVMCTS: .BYTE 'X,40,40,40
2849 016007 040
2850 016010 130 040 040 EVMDSR: .BYTE 'X,40,40,40
2851 016013 040
2852 016014 130 040 040 EVMDCD: .BYTE 'X,40,40,40
2853 016017 040
2854 016020 130 040 040 EVMRTS: .BYTE 'X,40,40,40
2855 016023 040
2856 016024 130 040 040 EVMRI: .BYTE 'X,40,40,40
2857 016027 040
2858 016030 130 040 040 EVMSQD: .BYTE 'X,40,40,40
2859 016033 040
2860 016034 130 040 040 EVMTM: .BYTE 'X,40,40,40
2861 016037 040
2862 016040 .BYTE 0
2863 016042 .EVEN
2864
2865 ;EXECUTION STATUS MESSAGES TO BE PRINTED TO KEEP OPERATOR AWAKE
2866 016042 047045 000 CR: .ASCIZ /%N/ ;CR FOR LINES IN A ROW
2867 016045 045 031523 040445 STXQ: .ASCIZ /%S3%ATXQ/ ;ABOUT TO TRANSMIT
2868 016052 054124 000121
2869 016056 051445 022463 052101 STXC: .ASCIZ /%S3%ATXC/ ;TX COMPLETED
2870 016064 041530 000
2871 016067 045 031523 040445 SRXQ: .ASCIZ /%S3%ARXQ/ ;ABOUT TO RECEIVE
2872 016074 054122 000121
2873 016100 051445 022463 042501 SDVE: .ASCIZ /%S3%AERR/ ;DEVICE ERROR
2874 016106 051122 000
2875 016111 045 031523 040445 SCM: .ASCIZ /%S3%ACMP/ ;ABOUT TO DO DATA CHECKING OF RECVD VS. EXPTD
2876 016116 046503 000120
2877 016122 051445 022463 044501 SDVI: .ASCIZ /%S3%AINI/ ;DEVICE ABOUT TO BE INITIALIZED
2878 016130 044516 000
2879 016133 045 031523 040445 SCML: .ASCIZ /%S3%ACML/ ;COMPARE LENGTH ERROR
2880 016140 046503 000114
2881 016144 051445 022463 041501 SCMD: .ASCIZ /%S3%ACMD/ ;COMPARE DATA ERROR
2882 016152 042115 000
```

CZCLKAO DMR,DMC-11 DATA COMM. LINK TEST  
CZCLKA.P11 18-APR-80 09:24

MACY11 30A(1052) 18-APR-80 09:24 PAGE 63  
GLOBAL FORMAT STATEMENTS, MESSAGES, AND ASCII INFO

SEQ 0062

2883 016155 045 031523 040445 SEOP:  
2884 016162 047505 000120  
2885  
2886

.ASCIZ /%S3%AEOP/ ;END OF PASS  
.EVEN

```
2887
2888
2889          :DEVICE ERROR MESSAGES
2890 016166 044524 042515 047440 DVEM0: .ASCII /TIME OUT WAITING FOR RDI TO CLEAR/
2891 016174 052125 053440 044501
2892 016202 044524 043516 043040
2893 016210 051117 051040 044504
2894 016216 052040 020117 046103
2895 016224 040505      122
2896 016227      015 020012 020040          .ASCIZ <15><12>/ SEL0 SEL2 /
2897 016234 042523 030114 020040
2898 016242 020040 051440 046105
2899 016250 020062 000040
2900 016254 044524 042515 047440 DVEM1: .ASCII /TIME OUT WAITING FOR RDI TO SET/
2901 016262 052125 053440 044501
2902 016270 044524 043516 043040
2903 016276 051117 051040 044504
2904 016304 052040 020117 042523
2905 016312      124
2906 016313      015 020012 020040          .ASCIZ <15><12>/ SEL0 SEL2 /
2907 016320 042523 030114 020040
2908 016326 020040 051440 046105
2909 016334 020062 000040
2910 016340 044524 042515 047440 DVEM3: .ASCII /TIME OUT WAITING FOR RUN TO SET/
2911 016346 052125 053440 044501
2912 016354 044524 043516 043040
2913 016362 051117 051040 047125
2914 016370 052040 020117 042523
2915 016376      124
2916 016377      015 020012 020040          .ASCIZ <15><12>/ SEL0 SEL2 /
2917 016404 042523 030114 020040
2918 016412 020040 051440 046105
2919 016420 020062 000040
2920 016424 044524 042515 047440 DVEM4: .ASCII /TIME OUT WAITING FOR OUTPUT INTERRUPT/
2921 016432 052125 053440 044501
2922 016440 044524 043516 043040
2923 016446 051117 047440 052125
2924 016454 052520 020124 047111
2925 016462 042524 051122 050125
2926 016470      124
2927 016471      015 020012 020040          .ASCIZ <15><12>/ SEL0 SEL2 /
2928 016476 042523 030114 020040
2929 016504 020040 051440 046105
2930 016512 020062 000040
2931 016516 047111 052520 020124 DVEM5: .ASCII /INPUT INTERRUPT WHEN EXPECTING OUTPUT/
2932 016524 047111 042524 051122
2933 016532 050125 020124 044127
2934 016540 047105 042440 050130
2935 016546 041505 044524 043516
2936 016554 047440 052125 052520
2937 016562      124
2938 016563      015 020012 020040          .ASCIZ <15><12>/ SEL0 SEL2 /
2939 016570 042523 030114 020040
2940 016576 020040 051440 046105
2941 016604 020062 000040
2942 016610 046111 042514 040507 DVEM6: .ASCII /ILLEGAL OUTPUT INTERRUPT/
```



2943	016616	020114	052517	050124	
2944	016624	052125	044440	052116	
2945	016632	051105	052522	052120	
2946	016640	005015	020040	051440	.ASCIZ <15><12>/ SEL2 SEL6 /
2947	016646	046105	020062	020040	
2948	016654	020040	042523	033114	
2949	016662	020040	000		
2950	016665	103	047117	051124	DVEM7: .ASCII /CONTROL OUT INSTEAD OF BA-CC OUT/
2951	016672	046117	047440	052125	
2952	016700	044440	051516	042524	
2953	016706	042101	047440	020106	
2954	016714	040502	041455	020103	
2955	016722	052517	124		
2956	016725	015	020012	020040	.ASCIZ <15><12>/ SEL2 SEL6 /
2957	016732	042523	031114	020040	
2958	016740	020040	051440	046105	
2959	016746	020066	000040		
2960					
2961	016752	054124	041040	043125	DVEM8: .ASCII /TX BUFF COMPLETED AND SHOULD BE RX/
2962	016760	020106	047503	050115	
2963	016766	042514	042524	020104	
2964	016774	047101	020104	044123	
2965	017002	052517	042114	041040	
2966	017010	020105	054122		
2967	017014	005015	020040	051440	.ASCIZ <15><12>/ SEL4 SEL6 /
2968	017022	046105	020064	020040	
2969	017030	020040	042523	033114	
2970	017036	020040	000		
2971	017041	122	020130	052502	DVEM9: .ASCII /RX BUFF COMPLETED AND SHOULD BE TX/
2972	017046	043106	041440	046517	
2973	017054	046120	052105	042105	
2974	017062	040440	042116	051440	
2975	017070	047510	046125	020104	
2976	017076	042502	052040	130	
2977	017103	015	020012	020040	.ASCIZ <15><12>/ SEL4 SEL6 /
2978	017110	042523	032114	020040	
2979	017116	020040	051440	046105	
2980	017124	020066	000040		
2981	017130	042040	053517	020116	DLLAB: .ASCII / DOWN LINE LOAD ABORTED/
2982	017136	044514	042516	046040	
2983	017144	040517	020104	041101	
2984	017152	051117	042524	104	
2985	017157	015	020012	020040	.ASCIZ <15><12>/ RXBUF TXBUF /
2986	017164	054122	052502	020106	
2987	017172	020040	052040	041130	
2988	017200	043125	000040		
2989					
2990	017204	051120	041517	042105	PROEM: .ASCIZ /PROCEDURE ERROR/
2991	017212	051125	020105	051105	
2992	017220	047522	000122		
2993	017224	047516	020116	054105	NXMM: .ASCIZ /NON EXIST MEM/
2994	017232	051511	020124	042515	
2995	017240	000115			
2996	017242	042104	046503	020120	DDCSR: .ASCIZ /DDCMP START REC/
2997	017250	052123	051101	020124	
2998	017256	042522	000103		

2999	017262	044504	041523	047117	DISCOM: .ASCIZ /DISCONNECT/
3000	017270	042516	057103	000	
3001	017275	114	051517	020124	LOSDAM: .ASCIZ /LOST DATA/
3002	017302	040504	040524	000	
3003	017307	104	041504	050115	DDCMPM: .ASCIZ /DDCMP MAINT REC/
3004	017314	046440	044501	052116	
3005	017322	051040	041505	000	
3006	017327	124	046511	020105	TIMOM: .ASCIZ /TIME OUT/
3007	017334	052517	000124		
3008	017340	040504	040524	041440	DATCKM: .ASCIZ /DATA CHECK/
3009	017346	042510	045503	000	
3010					
3011	017354				.EVEN
3012					

3013  
3014  
3015  
3016  
3017 017370 017370  
3018 017370 000400  
3019  
3020 020000

:THIS SECTION IS USED BY A M9301-YJ BOOT ROM FOR DOING DOWN-LINE-LOAD ?????  
:MUST BE IN THE AREA OF '017370 + 256. BYTES' + A FEW

BASE: .=17370  
.BLKB 256. ;BASE TABLE ADDRESS  
.=20000

3021	020000	052522	020116	042523	RUNSBM:	.ASCIZ	/RUN SET ILLEAGLLY/
3022	020006	020124	046111	042514			
3023	020014	043501	046114	000131			
3024	020022	054122	044440	046104	RXIDM:	.ASCIZ	/RX IDLE/
3025	020030	000105					
3026	020032	042103	043440	044514	CDGLM:	.ASCIZ	/CD GLITCHED/
3027	020040	041524	042510	000104			
3028	020046	052103	020123	040506	CTSFM:	.ASCIZ	/CTS FALIED/
3029	020054	044514	042514	000104			
3030	020062	054124	047040	052117	TXNC:	.ASCIZ	/TX NOT COMPLETE/
3031	020070	041440	046517	046120			
3032	020076	052105	000105				
3033	020102	054122	047040	052117	RXNC:	.ASCIZ	/RX NOT COMPLETE/
3034	020110	041440	046517	046120			
3035	020116	052105	000105				
3036	020122	042523	020103	042522	RXM1:	.ASCIZ	/SEC REQ ERR WORD 1/
3037	020130	020121	051105	020122			
3038	020136	047527	042122	030440			
3039	020144	000					
3040	020145	123	041505	051040	RXM2:	.ASCIZ	/SEC REQ ERR WORD 2/
3041	020152	050505	042440	051122			
3042	020160	053440	051117	020104			
3043	020166	000062					
3044						.EVEN	
3045							
3046							
3047							
3048							
3049							
3050							
3051							

.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.  
:--

3052  
3053  
3054  
3055  
3056  
3057  
3058  
3059  
3060  
3061  
3062  
3063  
3064  
3065  
3066  
3067  
3068  
3069  
3070  
3071  
3072  
3073  
3074  
3075  
3076  
3077  
3078  
3079  
3080  
3081  
3082  
3083  
3084  
3085  
3086  
3087  
3088  
3089  
3090  
3091  
3092  
3093  
3094  
3095  
3096  
3097  
3098  
3099  
3100  
3101  
3102  
3103  
3104  
3105  
3106  
3107

020170  
020170  
020170 005046  
020172 153716 007211  
020176 005046  
020200 153716 007210  
020204 013746 007170  
020210 012746 015600  
020214 012746 000004  
020220 010600  
020222 104414  
020224 062706 000012  
020230  
020230 104423  
020232  
020232  
020232 013746 007202  
020236 012746 014256  
020242 012746 000002  
020246 010600  
020250 104414  
020252 062706 000006  
020256  
020256 104423  
020260  
020260  
020260 013746 007200  
020264 010446  
020266 012746 014353  
020272 012746 000003  
020276 010600  
020300 104414  
020302 062706 000010  
020306  
020306 104423

BGNMSG ERR1  
PRINTB #EVTF5A,OFSET,<B,GOOD>,<B,BAD>  
ENDMSG  
BGNMSG ERR2  
PRINTB #EFM2,TEMP4  
ENDMSG  
BGNMSG ERR10  
PRINTB #EFM11,R4,TEMP3  
ENDMSG

ERR1::  
;INDIVIDUAL DATA COMPARE ERROR  
CLR -(SP)  
BISB BAD,(SP)  
CLR -(SP)  
BISB GOOD,(SP)  
MOV OFSET, -(SP)  
MOV #EVTF5A, -(SP)  
MOV #4, -(SP)  
MOV SP,RC  
TRAP C\$PNTB  
ADD #12,SP  
L10001:  
TRAP C\$MSG  
ERR2::  
;TOTAL DATA COMPARE FAILS ERROR  
MOV TEMP4, -(SP)  
MOV #EFM2, -(SP)  
MOV #2, -(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
L10002:  
TRAP C\$MSG  
ERR10::  
MOV TEMP3, -(SP)  
MOV R4, -(SP)  
MOV #EFM11, -(SP)  
MOV #3, -(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #10,SP  
L10003:  
TRAP C\$MSG

3108 020310  
3109 020310  
3110 020310  
3111 020310 013746 007204  
3112 020314 013746 007202  
3113 020320 013746 007200  
3114 020324 012746 015277  
3115 020330 012746 000004  
3116 020334 010600  
3117 020336 104414  
3118 020340 062706 000012  
3119 020344  
3120 020344 013746 007214  
3121 020350 012746 014321  
3122 020354 012746 000002  
3123 020360 010600  
3124 020362 104414  
3125 020364 062706 000006  
3126 020370  
3127 020370  
3128 020370 104423  
3129  
3130 020372  
3131 020372  
3132 020372  
3133 020372 013746 007202  
3134 020376 013746 007200  
3135 020402 012746 015262  
3136 020406 012746 000003  
3137 020412 010600  
3138 020414 104414  
3139 020416 062706 000010  
3140 020422  
3141 020422 013746 007214  
3142 020426 012746 014321  
3143 020432 012746 000002  
3144 020436 010600  
3145 020440 104414  
3146 020442 062706 000006  
3147 020446  
3148 020446  
3149 020446 104423  
3150  
3151 020450  
3152 020450  
3153 020450  
3154 020450 013746 007202  
3155 020454 013746 007200  
3156 020460 012746 015262  
3157 020464 012746 000003  
3158 020470 010600  
3159 020472 104414  
3160 020474 062706 000010  
3161 020500  
3162 020500  
3163 020500 104423

BGNMSG ERR8  
PRINTB #EVTF3D,TEMP3,TEMP4,CONOTM  
  
PRINTB #PCPM,PCADD  
  
ENDMSG  
  
BGNMSG ERR9  
PRINTB #EVTF3C,TEMP3,TEMP4  
  
PRINTB #PCPM,PCADD  
  
ENDMSG  
  
BGNMSG ERR13  
PRINTB #EVTF3C,TEMP3,TEMP4  
  
ENDMSG

ERR8::  
MOV CONOTM,-(SP)  
MOV TEMP4,-(SP)  
MOV TEMP3,-(SP)  
MOV #EVTF3D,-(SP)  
MOV #4,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #12,SP  
  
MOV PCADD,-(SP)  
MOV #PCPM,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
  
L10004:  
TRAP C\$MSG  
  
ERR9::  
MOV TEMP4,-(SP)  
MOV TEMP3,-(SP)  
MOV #EVTF3C,-(SP)  
MOV #3,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #10,SP  
  
MOV PCADD,-(SP)  
MOV #PCPM,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
  
L10005:  
TRAP C\$MSG  
  
ERR13::  
MOV TEMP4,-(SP)  
MOV TEMP3,-(SP)  
MOV #EVTF3C,-(SP)  
MOV #3,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #10,SP  
  
L10006:  
TRAP C\$MSG

3164  
3165 020502  
3166 020502  
3167 020502  
3168 020502 013746 007204  
3169 020506 013746 007202  
3170 020512 013746 007200  
3171 020516 012746 015277  
3172 020522 012746 000004  
3173 020526 010600  
3174 020530 104414  
3175 020532 062706 000012  
3176 020536  
3177 020536  
3178 020536 104423  
3179  
3180 020540  
3181 020540 000167  
3182 020542 177772  
3183  
3184

BGNMSG ERR14  
PRINTB #EVTF3D,TEMP3,TEMP4,CONOTM  
  
ENDMSG  
  
EXIT MSG

ERR14::

MOV CONOTM,-(SP)  
MOV TEMP4,-(SP)  
MOV TEMP3,-(SP)  
MOV #EVTF3D,-(SP)  
MOV #4,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #12,SP

L10007:

TRAP C\$MSG

.WORD JSJMP  
.WORD L10007-2-

3185  
3186  
3187  
3188  
3189  
3190  
3191  
3192  
3193  
3194  
3195  
3196  
3197  
3198  
3199  
3200  
3201  
3202  
3203  
3204  
3205  
3206  
3207  
3208  
3209  
3210  
3211  
3212  
3213  
3214  
3215  
3216  
3217  
3218  
3219  
3220  
3221  
3222  
3223  
3224  
3225  
3226  
3227  
3228  
3229  
3230  
3231

.SBTTL GLOBAL SUBROUTINES SECTION

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

.SBTTL CLOCK SETUP SUBROUTINE

;++  
: FUNCTIONAL DESCRIPTION:  
: THIS SUBROUTINE SETS UP THE CLOCK INFORMATION TABLE FOLLOWING A 'CLOCK'  
: CALL EXECUTED IN THE INITIALIZATION CODE. BUT SINCE THE 'CLOCK' CALL  
: SAYS NOTHING ABOUT AN LSI-11'S CLOCK, THIS ROUTINE IS ONLY USED IF A  
: LINE OR P-CLOCK IS FOUND.

INPUTS:  
R1= POINTS TO SUPERVISOR SPACE WHERE CLOCK INFO WAS RETURNED  
R2= POINTS TO 'CLK' TABLE WHERE CLOCK INFO WILL BE KEPT

IMPLICIT INPUTS:  
THE SUPERVISOR SPACE WHERE CLOCK INFO WAS RETURNED BY THE 'CLOCK' CALL

OUTPUTS:  
'CLKCSR' GETS LOADED WITH THE CLOCK'S CSR ADDRESS  
'CLKBR' GETS LOADED WITH THE CLOCK'S INTERRUPT LEVEL  
'CLKVEC' GETS LOADED WITH THE CLOCK'S INTERRUPT VECTOR  
'CLKHZ' GETS LOADED WITH THE LINE FREQ. (HERTZ RATE) WHICH DETERMINES  
THE NUMBER OF TICKS IN A SECOND

CALLING SEQUENCE:  
JSR PC,CLKSET ;CALL CLOCK SETUP WITH R1 & R2 SETUP  
:--

CLKSET:  
MOV (R1)+,(R2)+ ;LOAD CLOCK'S CSR ADDR. INTO 'CLKCSR'  
MOV (R1)+,(R2) ;LOAD CLOCK'S INT. LEVEL INTO 'CLKBR'  
ASL (R2) ;ADJUST THE INT. LEVEL FOR LOADING INTO  
ASL (R2) ; THE PSW WITH A 'SETVEC' CALL  
ASL (R2)  
ASL (R2)+  
MOV (R1)+,(R2)+ ;LOAD CLOCK'S INT. VECTOR INTO 'CLKVEC'  
MOV (R1)+,(R2)+ ;LOAD CLOCK'S HERTZ RATE INTO 'CLKHZ'  
RTS PC



3232  
 3233  
 3234  
 3235  
 3236  
 3237  
 3238  
 3239  
 3240  
 3241  
 3242  
 3243  
 3244  
 3245  
 3246  
 3247  
 3248  
 3249  
 3250  
 3251  
 3252  
 3253  
 3254  
 3255  
 3256  
 3257  
 3258  
 3259  
 3260  
 3261  
 3262  
 3263  
 3264  
 3265  
 3266  
 3267  
 3268  
 3269  
 3270  
 3271  
 3272  
 3273  
 3274  
 3275  
 3276  
 3277  
 3278  
 3279  
 3280  
 3281  
 3282  
 3283  
 3284  
 3285  
 3286  
 3287

```

.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.
--
  
```

BGNSRV CLKINT

CLKINT::

```

020570 005077 166456 CLR @CLKCSR ;DISABLE THE CLOCK FORM INTERRUPTING
020574 005337 007270 DEC TIMTCK ;DECREMENT THE # OF TICKS/SEC.
020600 001015 BNE 1$ ;GO CHECK TIMERS (1&2-TICKS, 3-SECONDS)
020602 013737 007260 007270 MOV CLKHZ,TIMTCK ;RESET THE # OF TICKS/SEC.
020610 005237 007266 INC TIMSEC ;INC # OF SECS-SINCE-START
020614 022737 000074 007266 CMP #60.,TIMSEC ;SEE IF WE'VE COUNTED 60 SECS. YET
020622 001004 BNE 1$ ;IF NOT, GO CHECK TIMERS
020624 005237 007264 INC TIMMIN ; ELSE INC MINUTES-SINCE-START
020630 005037 007266 CLR TIMSEC ; AND RESTART SECOND COUNTER
:
020634 005737 007272 1$: TST TIMER1 ;SEE IF TIMER #1, TIMING ANYTHING
020640 001402 BEQ 2$ ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
020642 005337 007272 DEC TIMER1 ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
020646 005737 007274 2$: TST TIMER2 ;SEE IF TIMER #2, TIMING ANYTHING
020652 001402 BEQ 3$ ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
020654 005337 007274 DEC TIMER2 ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
020660 005737 007276 3$: TST TIMERS ;SEE IF TIMER #3, TIMING ANYTHING
020664 001406 BEQ 4$ ; IF=0, NOTHING BEING TIMED, LEAVE
020666 023737 007260 007270 CMP CLKHZ,TIMTCK ;SEE IF A SECOND HAS BEEN COUNTED OFF
020674 001002 BNE 4$ ; BR IF NO
020676 005337 007276 DEC TIMERS ; ELSE DECREMENT THE TIMER VALUE (BY 1 SEC.)
  
```

CZCLKAO DMR,DMC-11 DATA COMM. LINK TEST  
CZCLKA.P11 18-APR-80 09:24

MACY11 30A(1052) 18-APR-80 09:24 PAGE 74  
CLOCK INTERRUPT SERVICE ROUTINE

SEQ 0073

3288 020702 013777 007262 166342 48:  
3289 020710  
3290 020710  
3291 020710 000002

MOV CLKEN,@CLKCSR ;REENABLE THE CLOCK TO INTERRUPT  
ENDSRV

L10010:  
RTI

3292  
 3293  
 3294  
 3295  
 3296  
 3297  
 3298  
 3299  
 3300  
 3301  
 3302  
 3303  
 3304  
 3305  
 3306  
 3307  
 3308  
 3309  
 3310  
 3311  
 3312  
 3313  
 3314  
 3315  
 3316  
 3317  
 3318  
 3319  
 3320  
 3321  
 3322  
 3323  
 3324  
 3325  
 3326  
 3327  
 3328  
 3329  
 3330  
 3331  
 3332  
 3333  
 3334  
 3335  
 3336  
 3337  
 3338  
 3339  
 3340  
 3341  
 3342  
 3343  
 3344  
 3345  
 3346  
 3347

```
.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 'EVTPTN' (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

:--

LOGTXQ:  MOV    #STXQ,TEMP1      ;SET UP MSG. TO PRINT
:        MOV    #TXQ,TEMP      ;SET UP EVENT TYPE
:        BR     LOGS1          ;GO LOG EVENT AND TIME

LOGTXC:  MOV    #STXC,TEMP1      ;SET UP MSG. TO PRINT
:        MOV    #TXC,TEMP      ;SET UP EVENT TYPE
:        BR     LOGS1          ;GO LOG EVENT AND TIME

LOGRXQ:  MOV    #SRXQ,TEMP1      ;SET UP MSG. TO PRINT
:        MOV    #RXQ,TEMP      ;SET UP EVENT TYPE
:        BR     LOGS1          ;GO LOG EVENT AND TIME

LOGRXC:  MOV    #RXC,TEMP        ;SET UP EVENT TYPE
:        BR     LOGS1          ;GO LOG EVENT AND TIME

LGDVE:
```

```
020712
020712 012737 016045 007174
020720 012737 000000 007172
020726 000510

020730
020730 012737 016056 007174
020736 012737 000002 007172
020744 000501

020746
020746 012737 016067 007174
020754 012737 000004 007172
020762 000472

020764
020764 012737 000006 007172
020772 000466
020774
```

```

3348 020774 012737 016100 007174  MOV    #SDVE,TEMP1    ;SET UP MSG. TO PRINT
3349 021002 012737 000010 007172  MOV    #DER,TEMP      ;SET UP EVENT TYPE
3350 021010 000474                BR     LOGS3          ;GO LOG EVENT AND TIME
3351
3352 021012                LOGDVI:
3353 021012 012737 016122 007174  MOV    #SDVI,TEMP1    ;SET UP MSG. TO PRINT
3354 021020 012737 000012 007172  MOV    #DVI,TEMP      ;SET UP EVENT TYPE
3355 021026 113737 007220 007176  MOVB  MODTYP,TEMP2
3356 021034 113737 007222 007177  MOVB  MLTYP,TEMP2+1
3357 021042 013737 007230 007200  MOV    RPASS,TEMP3
3358 021050 013737 007226 007202  MOV    PARAM,TEMP4    ;SET UP EVNT ENTRIES
3359 021056 000451                BR     LOGS3          ;GO LOG EVENT AND TIME
3360
3361 021060                LOGCMP:
3362 021060 012737 016111 007174  MOV    #SCM,TEMP1     ;SET UP MSG. TO PRINT
3363 021066 012737 000014 007172  MOV    #DCK,TEMP      ;SET UP EVENT TYPE
3364 021074 000442                BR     LOGS3
3365 021076                LOGCML:
3366 021076 012737 016133 007174  MOV    #SCML,TEMP1
3367 021104 012737 000020 007172  MOV    #DLE,TEMP      ;SET UP MSG. AND TYPE
3368 021112 000433                BR     LOGS3          ;GO LOG EVENT AND TIME
3369 021114                LOGCMD:
3370 021114 012737 016144 007174  MOV    #SCMD,TEMP1
3371 021122 012737 000022 007172  MOV    #DDE,TEMP
3372 021130 000424                BR     LOGS3          ;GO LOG MSG TYPE AND TIME
3373 021132                LOGEOP:
3374 021132 012737 016155 007174  MOV    #SEOP,TEMP1
3375 021140 012737 000024 007172  MOV    #EOP,TEMP
3376 021146 000415                BR     LOGS3          ;GO LOG MSG TYPE AND TIME
3377
3378 021150 013746 007144                LOGS1: MOV    ERRCNT,-(SP)    ;SAVE CURRENT ERROR COUNT
3379 021154 004737 034064                JSR    PC,DVMODS      ;GO GET MODEM STATUS
3380 021160 012604                MOV    (SP)+,R4       ;GET SAVED ERRCNT VALUE
3381 021162 020437 007144                CMP    R4,ERRCNT     ;WHERE ANY ERRORS FOUND
3382 021166 001402                BEQ    1$             ;BR IF NONE
3383 021170 000137 021404                JMP    LOGEX          ;ELSE, LEAVE WITHOUT LOGGING ANYTHING
3384                                ; BUT THE DEVICE ERROR FROM 'D/MODS'
3385 021174 013737 010206 007202 1$:  MOV    MODS,TEMP4     ;AND PUT IT IN TEMP4
3386
3387 021202                LOGS3:
3388 021202 022737 000006 007172  CMP    #RXC,TEMP
3389 021210 001434                BEQ    LOGS5          ;IF RXC DONT PRINT
3390 021212 032737 000001 007226  BIT    #STATB,PARAM
3391 021220 001430                BEQ    LOGS5          ;IF NO STATUS SELECTED
3392                                ;GO TO 5
3393
3394 021222 022737 000010 007136  CMP    #10,LNCNT     ;HAVE WE DONE 10?
3395 021230 001012                BNE    LOGS4          ;IF NOT GO TO 4
3396 021232 005037 007136                CLR    LNCNT         ;ESLE CLEAR IT
3397
3398 021236                PRINTF #CR           ;ELSE PRINT CR
3399 021236 012746 016042                MOV    #CR,-(SP)
3400 021242 012746 000001                MOV    #1,-(SP)
3401 021246 010600                MOV    SP,R0
3402 021250 104417                TRAP  C$PNTF
3403 021252 062706 000004                ADD    #4,SP

```



```

3433          .SBTTL          DUMP EVENT LOG AND BASE TABLE
3434
3435
3436 021406 010246          REPORT: MOV      R2,-(SP)          ;SAVE R2,R3,R4 ON THE STACK
3437 021410 010346          MOV      R3,-(SP)
3438 021412 010446          MOV      R4,-(SP)
3439 021414 005037 007172  CLR      TEMP
3440 021420          GMANIL #BASM1A,TEMP,1,YES
3441 021420 104443
3442 021422 000404          TRAP    CS$MAN
3443 021424 007172          BR      10000$
3444 021426 000130          .WORD  TEMP
3445 021430 014755          .WORD  T$CODE
3446 021432 000001          .WORD  #BASM1A
3447 021434          .WORD  1
3448 021434 005737 007172          TST     TEMP
3449 021440 001413          BEQ     BASN1          ;IF NO BASE PRINT GO TO 1
3450
3451 021442 012737 017370 007146  MOV     #BASE,STADD
3452 021450 012737 017767 007150  MOV     #BASE+255.,ENADD
3453 021456 012737 000001 007152  MOV     #1,BYTBIT
3454 021464 004737 022442          JSR     PC,DUMPSR          ;GO DUMP BASE TABLE
3455 021470 013702 007300          BASN1: MOV     EVTPTR,R2          ;MAKE R2 A POINTER TO EVENT TABLE
3456 021474 023727 007302 177777  CMP     EVTLOG,#-1          ;SEE IF EVENT TABLE IS EMPTY
3457 021502 001034          BNE     RPT0          ;BR IF NO
3458 021504          PRINTS #NULEVT          ;IF EMPTY TELL OPERATOR.
3459 021504 012746 015016          MOV     #NULEVT,-(SP)
3460 021510 012746 000001          MOV     #1,-(SP)
3461 021514 010600          MOV     SP,R0
3462 021516 104416          TRAP    CS$PNTS
3463 021520 062706 000004          ADD     #4,SP
3464 021524 000137 022320          JMP     ENDEVT          ;AND END
3465
3466 021530 162702 000012          RPT:   SUB     #12,R2          ;NOW POINT BACK TO TOP OF ENTRY U
3467          ;JUST PRINTED
3468
3469 021534 020227 007302          CMP     R2,#EVTLOG          ;POINTING TO TOP OF EVNT LOG QUEUE?
3470 021540 001010          BNE     RPT1          ;BR IF NO
3471 021542 012702 010204          MOV     #EVTEND,R2          ;SET R2 TO POINT TO BOTTOM OF LOG
3472 021546 026227 177776 177777  CMP     -2(R2),#-1
3473 021554 001007          BNE     RPT0          ;IF END OF LOG IS NOT EMPTY
3474 021556 000137 022320          JMP     ENDEVT          ;CONTINUE...ELSE EXIT
3475
3476 021562 020237 007300          RPT1:  CMP     R2,EVTPTR          ;ARE WE BACK TO POINTER?
3477 021566 001002          BNE     RPT0          ;IF NOT CONTINUE
3478 021570 000137 022320          JMP     ENDEVT          ;IF SO EXIT....
3479
3480 021574 162702 000012          RPT0:  SUB     #12,R2          ;POINT R2 TO START OF ENTRY
3481 021600          RPTAA: PRINTS #EVTFO          ;PRINT EVENT ENTRY HEADER
3482 021600 012746 015051          MOV     #EVTFO,-(SP)
3483 021604 012746 000001          MOV     #1,-(SP)
3484 021610 010600          MOV     SP,R0
3485 021612 104416          TRAP    CS$PNTS
3486 021614 062706 000004          ADD     #4,SP
3487 021620 112203          MOV     (R2)+,R3          ;PUT EVENT TYPE INTO R3
3488 021622 112237 010276          MOV     (R2)+,EVTCK
  
```

3489	021626	112237	010272	MOVB	(R2)+,EVTSEC	;PUT EVENT TIME (TICKS,SECS,MINS IN TEMP LOC.S)		
3490	021632	112237	010274	MOVB	(R2)+,EVTMIN			
3491	021636			PRINTS	#EVTF1,EVTMIN,EVTSEC,EVTCK,EVTLST(R3)	;PRINT EVENT TIME AND DESCRIPT.		
3492	021636	016346	010244				MOV	EVTLST(R3),-(SP)
3493	021642	013746	010276				MOV	EVTCK,-(SP)
3494	021646	013746	010272				MOV	EVTSEC,-(SP)
3495	021652	013746	010274				MOV	EVTMIN,-(SP)
3496	021656	012746	015147				MOV	#EVTF1,-(SP)
3497	021662	012746	000005				MOV	#5,-(SP)
3498	021666	010600					MOV	SP,R0
3499	021670	104416					TRAP	CSPNTS
3500	021672	062706	000014				ADD	#14,SP
3501	021676	000173	010306	JMP	@RPTDSP(R3)	;DISPATCH TO DECODING SECTION FOR SPECIFIC TYPE		
3502								
3503	021702	012237	010300	RPTTXQ:	MOV	(R2)+,EVTADD	;STORE MESSAGE ADDRESS FOR PRINTING	
3504	021706	012237	010302		MOV	(R2)+,EVTBCT	;STORE BYTE COUNT FOR PRINTING	
3505	021712	012203			MOV	(R2)+,R3	;STORE MODEM STATUS FOR PRINTING	
3506	021714				PRINTS	#EVTF2,EVTADD,EVTBCT	;PRINT ADDR,BYTE CNT	
3507	021714	013746	010302				MOV	EVTBCT,-(SP)
3508	021720	013746	010300				MOV	EVTADD,-(SP)
3509	021724	012746	015176				MOV	#EVTF2,-(SP)
3510	021730	012746	000003				MOV	#3,-(SP)
3511	021734	010600					MOV	SP,R0
3512	021736	104416					TRAP	CSPNTS
3513	021740	062706	000010				ADD	#10,SP
3514	021744	004737	022330	JSR	PC,RPTMSB	;GO PRINT MODEM STATUS		
3515	021750	000137	021530	JMP	RPT	;GO BACK FOR NEXT EVENT ENTRY		
3516								
3517	021754	012237	010304	RPTDER:	MOV	(R2)+,EVTTMP	;GET ADDRESS OF DEVICE INFO MESSAGE	
3518	021760	012237	010334		MOV	(R2)+,DEV1	;STORE DEVICE REG CONTENTS FOR PRINTING	
3519	021764	012237	010336		MOV	(R2)+,DEV2		
3520	021770				PRINTS	#EVTF3,EVTTMP	;PRINT DEVICE REG CONTENTS.	
3521	021770	013746	010304				MOV	EVTTMP,-(SP)
3522	021774	012746	015250				MOV	#EVTF3,-(SP)
3523	022000	012746	000002				MOV	#2,-(SP)
3524	022004	010600					MOV	SP,R0
3525	022006	104416					TRAP	CSPNTS
3526	022010	062706	000006				ADD	#6,SP
3527	022014			PRINTS	#EVTF3C,DEV1,DEV2			
3528	022014	013746	010336				MOV	DEV2,-(SP)
3529	022020	013746	010334				MOV	DEV1,-(SP)
3530	022024	012746	015262				MOV	#EVTF3C,-(SP)
3531	022030	012746	000003				MOV	#3,-(SP)
3532	022034	010600					MOV	SP,R0
3533	022036	104416					TRAP	CSPNTS
3534	022040	062706	000010				ADD	#10,SP
3535	022044	000137	021530	JMP	RPT	;GO BACK FOR NEXT EVENT ENTRY		
3536								
3537	022050	005037	010334	RPTDVI:	CLR	DEV1		
3538	022054	005037	010336		CLR	DEV2	;CLEAR UPPER BYTES OF DEV1 & DEV2 BEFORE USE	
3539	022060	112237	010334		MOVB	(R2)+,DEV1	;STORE SETUP OPERATION PARAMETERS FOR PRINTING	
3540	022064	112237	010336		MOVB	(R2)+,DEV2		
3541	022070	012237	010340		MOV	(R2)+,DEV3		
3542	022074	012237	010342		MOV	(R2)+,DEV4		
3543	022100	010246			MOV	R2,-(SP)	;SAVE R2 ON THE STACK	
3544	022102	004737	023022		JSR	PC,SHWOP	;GO PRINT MODE, MAINT-LOOP TYPE, PARAMTERS.	

```

3545 022106 012602
3546 022110 000137 021530
3547 022114 012237 010300
3548 022120 012237 010302
3549 022124 012237 010304
3550 022130
3551 022130 013746 010304
3552 022134 013746 010302
3553 022140 013746 010300
3554 022144 012746 015521
3555 022150 012746 000004
3556 022154 010600
3557 022156 104416
3558 022160 062706 000012
3559
3560 022164 000137 021530
3561
3562
3563 022170 012237 010300
3564 022174 012237 010302
3565 022200 012237 010304
3566 022204
3567 022204 013746 010304
3568 022210 013746 010302
3569 022214 013746 010300
3570 022220 012746 015321
3571 022224 012746 000004
3572 022230 010600
3573 022232 104416
3574 022234 062706 000012
3575 022240 000137 021530
3576
3577 022244
3578 022244 012237 010300
3579 022250 012237 010302
3580 022254 012237 010304
3581 022260
3582 022260 013746 010304
3583 022264 013746 010302
3584 022270 013746 010300
3585 022274 012746 015423
3586 022300 012746 000004
3587 022304 010600
3588 022306 104416
3589 022310 062706 000012
3590
3591 022314 000137 021530
3592
3593 022320 012604
3594 022322 012603
3595 022324 012602
3596 022326 000207
3597
3598
3599
3600

      MOV      (SP)+,R2      ;RESTORE R2
      JMP      RPT          ;GO BACK FOR NEXT EVENT ENTRY
RPTTEOP: MOV      (R2)+,EVTADD
      MOV      (R2)+,EVTBCT
      MOV      (R2)+,EVTTMP
      PRINTS   #EVT4B,EVTADD,EVTBCT,EVTTMP ;PRINT ADDR,RXBYTES,CMPBYTES.
      MOV      EVTTMP,-(SP)
      MOV      EVTBCT,-(SP)
      MOV      EVTADD,-(SP)
      MOV      #EVT4B,-(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   #EVT4,EVTADD,EVTBCT,EVTTMP ;PRINT ADDR, BYTE CNT, # CMP ERRS
      MOV      EVTTMP,-(SP)
      MOV      EVTBCT,-(SP)
      MOV      EVTADD,-(SP)
      MOV      #EVT4,-(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   #EVT4A,EVTADD,EVTBCT,EVTTMP ;PRINT ADDR,RXBYTES,CMPBYTES.
      MOV      EVTTMP,-(SP)
      MOV      EVTBCT,-(SP)
      MOV      EVTADD,-(SP)
      MOV      #EVT4A,-(SP)
      MOV      #4,-(SP)
      MOV      SP,R0
      TRAP     C$PNTS
      ADD      #12,SP

      JMP      RPT          ;THEN GO GET NEXT EVENT ENTRY

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

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

```





3631  
3632  
3633  
3634  
3635  
3636  
3637  
3638  
3639  
3640  
3641  
3642  
3643  
3644  
3645  
3646  
3647  
3648  
3649  
3650  
3651  
3652  
3653  
3654  
3655  
3656  
3657  
3658  
3659  
3660  
3661  
3662  
3663  
3664  
3665  
3666  
3667  
3668  
3669  
3670  
3671  
3672  
3673  
3674  
3675  
3676  
3677  
3678  
3679  
3680  
3681  
3682  
3683  
3684  
3685  
3686

.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  
:--

```
DUMPSR: MOV STADD,R2 ;SET R2 UP TO STARTING ADDR.
DUM4: CLR R3 ;CLEAR R3
PRINTF #BASM1,R2 ;PRINT ADDRESS

MOV R2,-(SP)
MOV #BASM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP

DUM3: TST BYTBIT ;IS THIS BYTE OR WORD
BEQ DUM1 ;BR IF WORD
MOVB (R2)+,TEMP ;MOV BYTE TO TEMP
PRINTF #BASM3,<B,TEMP> ;PRINT BYTE

CLR -(SP)
BISB TEMP,(SP)
MOV #BASM3,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP

DUM1: BR DUM2
PRINTF #BASM2,(R2)+ ;PRINT WORD

MOV (R2)+,-(SP)
MOV #BASM2,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP

DUM2: CMP R2,ENADD ;COMPARE FOR LAST ADD
```

3687 022562 003005  
3688 022564 005203  
3689 022566 022703 000010  
3690 022572 001725  
3691 022574 000736  
3692  
3693 022576 000207  
3694

DUMEX: RTS PC

BGT DUMEX  
INC R3  
CMP #8,R3  
BEQ DUM4  
BR DUM3

;IF DONE EXIT  
;ELSE BUMP R3  
;HAVE WE PRINTED 8 ACCROSS  
;IF SO GO BACK TO 4  
;ELSE GO BACK AND PRINT ANOTHER  
;BYTE OR WORD  
;RETURN TO CALLER

3695  
 3696  
 3697  
 3698  
 3699  
 3700  
 3701  
 3702  
 3703  
 3704  
 3705  
 3706  
 3707  
 3708  
 3709  
 3710  
 3711  
 3712  
 3713  
 3714  
 3715  
 3716  
 3717  
 3718  
 3719  
 3720  
 3721  
 3722  
 3723  
 3724  
 3725  
 3726  
 3727  
 3728  
 3729  
 3730  
 3731  
 3732  
 3733  
 3734  
 3735  
 3736

.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
  :--
  
```

```

  ADDCC: ADD    CURCC,TOTCC      ;ADD CURRENT TO TOTAL
         CMP    #BUFLIM,TOTCC   ; COMPARE TO 'BUFLIM'
         BHIS  ADDC1            ;IF NOT MORE THEN 'BUFLIM' EXIT
         ; PRINT MESSAGE AND TRUNCATE COUNT
         PRINTF #MSGTRU
         MOV    #MSGTRU,-(SP)
         MOV    #1,-(SP)
         MOV    SP,R0
         TRAP  C$PNTF
         ADD    #4,SP
  SUB    CURCC,TOTCC      ;SUB CURRENT FROM TOTAL
  MOV    #BUFLIM,TEMP    ;MOV 'BUFLIM' TO TEMP
  SUB    TOTCC,TEMP      ;SUB TOTAL FROM 'BUFLIM'
  MOV    TEMP,CURCC      ;AND ESTABLISH NEW CURRENT
  ADD    CURCC,TOTCC     ;ADD 'ADJUSTED CURRENT' TO TOTAL CHAR. CNT.
  ADDC1: RTS    PC       ;RETURN TO CALLER
  
```

```

3737 .SBTTL          BUILD MESSAGE BUFFERS SUBROUTINE
3738
3739
3740 :++
3741 : FUNCTIONAL DESCRIPTION:
3742 :   BLDBUF-- BUILD POINTER TABLE AND BUFFERS
3743 :
3744 :   THIS SUBROUTINE ADDS A MESSAGE TO THE TRANSMIT OR EXPECT LIST
3745 :   USING THE POINTER, BYTE COUNT, AND ADDRESS PASSED TO IT.
3746 :
3747 : INPUTS:
3748 :   CURCC= CHAR. COUNT OF MESSAGE TO BE ADDED
3749 :   CURADD= ADDRESS OF MESSAGE TO BE ADDED
3750 :   CPTR=  ADDRESS OF POINTER TABLE WORD WHERE MESSAGE POINTERS ARE
3751 :           TO BE BUILT
3752 :   MSGTYP= VALUE TO USE AS AN INDEX TO FIND SOURCE OF MESSAGE DATA
3753 :           INDEX INTO DMSGCT() AND DMSGAD().
3754 :
3755 : OUTPUTS:
3756 :   A MESSAGE ADDED TO EITHER TXBUF OR CMPBUF
3757 :   APPROPRIATE POINTERS IN PTRTAB POINTER TABLE
3758 :
3759 : CALLING SEQUENCE:
3760 :   JSR PC,BLDBUF          ;BUILD MESSAGE IN BUFFER  " ADD PTRS.
3761 :--
3761 022676          BLDBUF:
3762 022676 010246          MOV     R2,-(SP)          ;SAVE R2 AND R3 ON THE STACK
3763 022700 010346          MOV     R3,-(SP)
3764 022702 013702 007162  MOV     CPTR,R2
3765
3766 022706 013722 007164  BLDB1:  MOV     CURADD,(R2)+      ;PUT CURRENT ADD ON POINTER TAB
3767 022712 013722 007156  MOV     CURCC,(R2)+      ;PUT CURRENT CC ON POINTER TAB
3768 022716 010237 007162  MOV     R2,CPTR        ;PUT UPDATED R2 BACK TO CURRENT POINT
3769 022722 013702 007154  MOV     MSGTYP,R2     ;GET MESSAGE TYPE TO USE AS INDEX
3770 022726 006302          ASL     R2                ;DOUBLE FOR WORD INDEX
3771 022730 013737 007164 007172  MOV     CURADD,TEMP    ;MOVE CURRENT ADD TO TEMP
3772 022736 063737 007156 007172  ADD     CURCC,TEMP     ;ADD CHAR COUNT TO IT TO GET END
3773 022744 013703 007164  MOV     CURADD,R3     ;SET R3 TO CURRENT START ADD
3774 022750 016237 002150 007176  BLDB2:  MOV     DMSGCT(R2),TEMP2  ;GET BYTE COUNT
3775 022756 016204 002176  MOV     DMSGAD(R2),R4  ;PUT STARTING FROM ADD IN R4
3776 022762 060437 007176  ADD     R4,TEMP2     ;ADD IT TO TEMP2 TO GET END OF FROM
3777 022766 112423          BLDB3:  MOVB    (R4)+,(R3)+    ;MOV BYTE FROM PATTERN TO BUFFER
3778 022770 020337 007172  CMP     R3,TEMP      ;ALL DONE?
3779 022774 001404          BEQ    BLDBEX        ;IF SO EXIT
3780 022776 020437 007176  CMP     R4,TEMP2     ;IS PATTERN COUNT EXPIRED
3781 023002 001762          BEQ    BLDB2        ;IF SO GO START AGAIN
3782 023004 000770          BR     BLDB3        ;IF NOT GET ANOTHER BYTE
3783 023006 063737 007156 007164  BLDBEX:  ADD     CURCC,CURADD  ;BUMP CURADD
3784 023014 012603          MOV     (SP)+,R3     ;RESTORE R3 AND R2
3785 023016 012602          MOV     (SP)+,R2
3786 023020 000207          RTS     PC          ;RETURN TO CALLER
3787
3788

```

3789  
 3790  
 3791  
 3792  
 3793  
 3794  
 3795  
 3796  
 3797  
 3798  
 3799  
 3800  
 3801  
 3802  
 3803  
 3804  
 3805  
 3806  
 3807  
 3808  
 3809  
 3810  
 3811  
 3812  
 3813  
 3814  
 3815  
 3816  
 3817  
 3818  
 3819  
 3820  
 3821  
 3822  
 3823  
 3824  
 3825  
 3826  
 3827  
 3828  
 3829  
 3830  
 3831  
 3832  
 3833  
 3834  
 3835  
 3836  
 3837  
 3838  
 3839  
 3840  
 3841  
 3842  
 3843  
 3844

.SBTTL SHOW MODE OF OPERATION, LOOP TYPE AND QUALIFIERS

++  
 : FUNCTIONAL DESCRIPTION:  
 : SHWOP - SHOW MODE OF OPERATION, LOOP, QUALIFIERS  
 : PRINTED ON THE OPERATOR'S CONSOLE.

: INPUTS:  
 : DEV1= MODE TYPE (MODTYP)  
 : DEV2= MAINT LOOP TYPE (MLTYP)  
 : DEV3= 'RUN PASS' COUNT (RPASS) - COUNT DOWN  
 : DEV4= PARAMETERS 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  
 :--

```

SHWOP: MOV     DEV1,R2           ;GET THE MODE TYPE IN R2
        ASL     R2             ;MAKE IT A WORD TABLE OFFSET
        MOV     MODES(R2),TEMP ;GET ADDRESS OF MODE-IN-ASCII
        MOV     DEV2,R2       ;GET MAINTENANCE LOOP TYPE
        ASL     R2
        MOV     #LP00,TEMP3    ;LOAD TEMP3 TO POINT TO "/LOOP="
        TST     R2            ;SEE IF /LOOP=XXXXX OR NONE
        BNE     10$           ;BR IF /LOOP= OF SOME KIND
        MOV     #LP0,TEMP3     ;IF NO LOOP THEN DON'T PRINT "/LOOP="
10$:    MOV     LOOPS(R2),TEMP1 ;GET ADDRESS OF LOOP-IN-ASCII
        MOV     DEV3,TEMP2    ;GET NUMBER OF PASSES
        PRINTS #SHF0,TEMP,TEMP3,TEMP1,TEMP2
                                MOV     TEMP2,-(SP)
                                MOV     TEMP1,-(SP)
                                MOV     TEMP3,-(SP)
                                MOV     TEMP,-(SP)
                                MOV     #SHF0,-(SP)
                                MOV     #5,-(SP)
                                MOV     SP,R0
                                TRAP   C$PNTS
                                ADD     #14,SP

        CLR     R2             ;NOW SET UP FOR QUALIFIERS IN ASCII
        MOV     #PST,TEMP
        BIT     #STATB,DEV4    ;SEE IF /STATUS OR /NOSTATUS
        BNE     1$            ;BR IF /STATUS
        MOV     #PNST,TEMP
        MOV     #PCK,TEMP1     ;SEE IF /CHECK OR /NOCHECK
        BIT     #DATCKB,DEV4   ;BR IF /CHECK
        BNE     2$
        MOV     #PNCK,TEMP1
        MOV     #PEC,TEMP2
        BIT     #ECHOB,DEV4    ;SEE IF /ECHO OR /NOECHO
        BNE     5$            ;BR IF /ECHO
        MOV     #PNEC,TEMP2
  
```

```
3845
3846
3847 023236          5$: PRINTS #SHF1,TEMP,TEMP1,TEMP2          ;,TEMP3,TEMP4 **;SEE NOTE ABOVE
3848 023236 013746 007176          MOV          TEMP2,-(SP)
3849 023242 013746 007174          MOV          TEMP1,-(SP)
3850 023246 013746 007172          MOV          TEMP,-(SP)
3851 023252 012746 014223          MOV          #SHF1,-(SP)
3852 023256 012746 000004          MOV          #4,-(SP)
3853 023262 010600          MOV          SP,R0
3854 023264 104416          TRAP        C$PNTS
3855 023266 062706 000012          ADD         #12,SP
3856 023272 000207          RTS         PC          ;RETURN
3857
3858
```

```

3859 .SBTTL          TRAVERSE COMMAND LINE SUBROUTINES
3860
3861 :++
3862 :      P$TRV SUBROUTINE
3863
3864 :PARSE THE COMMAND LINE SUBROUTINE
3865 :TAKE ACTIONS (VIA ACTION TREE) AS PARSING LINE
3866 :PARSING DIRECTIONS FROM 'CLI PARSING NODES'
3867 :      REGS USED:
3868
3869 :      R1,R5=SCRATCH          PSNUM=NUMERIC CODE FROM DATA
3870 :      R2=ACTION CODE PARAMETER FROM TREE
3871 :      R3=PARSE TREE POINTER
3872 :      R4=INPUT STRING POINTER
3873 :      CALLING SEQUENCE:
3874 :      JSR      PC,P$TRV
3875 :--
3876
3877 023274          P$TRV:
3878 023274 013704 003310      MOV      PSBUFA,R4
3879 023300 013703 003312      MOV      P$TREE,R3
3880 023304 105714          P$TR5:  TSTB      (R4)          ;SEE IF ANY CHARS LEFT IN INPUT STRING
3881 023306 001441          BEQ      P$EXIT      ;BR IF NO
3882 023310 121327 000013      CMPB      (R3),#11.  ;SEE IF SPECIAL CLI CHAR CODE OR ASCII
3883 023314 003023          BGT      20$        ;BR IF REGULAR ASCII CHAR.
3884 023316 111305          MOVB      (R3),R5    ;GET SPECIAL CHAR CODE INTO R5
3885 023320 006305          ASL      R5
3886 023322 016505 023336      MOV      10$(R5),R5 ;BUILD TRAVERSE ROUTINE ADDRESS
3887 023326 062705 023336      ADD      #10$,R5
3888 023332 004715          JSR      PC,(R5)    ;JSR TO SPECIAL CLI TRAVERSE ROUTINE
3889 023334 000763          BR       P$TR5     ;GO SEE IF MORE OF STRING LEFT
3890
3891
3892 023336 000114          10$:   .WORD    TRVERR-10$ ;TRAVERSE TABLE FOR 'CLI FUNCTIONS'
3893 023340 000134          .WORD    TRVEXI-10$ ;1
3894 023342 000152          .WORD    TRVBR-10$   ;2
3895 023344 000162          .WORD    TRVBIF-10$  ;3
3896 023346 000204          .WORD    TRVSPA-10$  ;4
3897 023350 000270          .WORD    TRVNUM-10$  ;5
3898 023352 000604          .WORD    TRVALP-10$  ;6
3899 023354 000650          .WORD    TRVALN-10$  ;7
3900 023356 000270          .WORD    TRVOCT-10$  ;8
3901 023360 000256          .WORD    TRVDEC-10$  ;9
3902 023362 000736          .WORD    TRVSTR-10$ ;10
3903
3904 :NOT A SPECIAL CODE
3905
3906 023364 121314          20$:   CMPB      (R3),(R4)  ;SEE IF FIRST CHAR OF STRING IS A MATCH
3907 023366 001403          BEQ      22$        ;BR IF A MATCH
3908 023370 004737 023434      JSR      PC,TRVBR   ;IF NOT A MATCH, GO TAKE MISS BRANCH
3909 023374 000743          BR       P$TR5     ;THEN GO BACK PT'G TO MISS NODE
3910 023376 004737 023414      22$:   JSR      PC,TRVACT  ;IF A MATCH, GO DO ACTION DEFINED BY
3911 023402 062703 000004      ADD      #4,R3     ;ACTION CODE IN CLI NODE, THEN
3912
3913 023406 005204          INC      R4        ;ADJUST PTR TO NEXT CLI NODE
3914 023410 000735          BR       P$TR5     ;ADJUST BUF PTR TO NEXT CHAR IF MATCH

```



```

3915
3916 023412 000207 P$EXIT: RTS PC ;RETURN FROM PARSER
3917
3918 ;-----
3919
3920 ;GOTO USER ACTION ROUTINE
3921 023414 116302 000001 TRVACT: MOVB 1(R3),R2 ;GET ACTION CODE FROM CLI NODE
3922 023420 042702 177400 BIC #177400,R2 ;CLEAR ANY SIGN EXTENSION
3923 023424 013705 003314 MOV P$ACT,R5 ;GET ADDRESS OF CLI ACTION ROUTINE
3924 023430 004715 JSR PC,(R5) ;GO DO ACTION DEFINED BY CODE
3925 023432 000207 RTS PC ;RETURN TO CALLING CODE
3926
3927 ;TAKE BRANCH IN TREE
3928 023434 016305 000002 TRVBRC: MOV 2(R3),R5 ;GET BRANCH DISPLACEMENT FROM TREE
3929 023440 060503 ADD R5,R3 ; AND POINT R3 TO THE 'MISS' NODE
3930 023442 000207 RTS PC ; RETURN TO P$TRV
3931
3932 ;NO BRANCH TAKEN
3933 023444 062703 000004 TRVNOB: ADD #4,R3 ;THINGS OK, UPDATE R3 TO POINT TO NEXT
3934 023450 000207 RTS PC ; NODE AND RETURN TO P$TRV
3935
3936 ;-----
3937 023452 004737 023414 TRVERR: JSR PC,TRVACT ;TAKE ERROR ACTION
3938 023456 112737 177777 003325 MOVB #-1,P$GDBD ;SET ERROR RETURN FLAG
3939 023464 005726 TST (SP)+ ;GET RID OF 'JSR PUSH TO TRVERR'
3940 023466 000137 023412 JMP P$EXIT ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
3941
3942 023472 004737 023414 TRVEXI: JSR PC,TRVACT ;TAKE EXIT ACTION
3943 023476 105037 003325 CLRB P$GDBD ;SET GOOD/BAD FLAG TO 'SUCCESS (0)''
3944 023502 005726 TST (SP)+ ;GET RID OF 'JSR PUSH TO TRVEXI''
3945 023504 000137 023412 JMP P$EXIT ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
3946
3947 023510 004737 023414 TRVBR: JSR PC,TRVACT ;GO TAKE BRANCH ACTION
3948 023514 000137 023434 JMP TRVBRC
3949
3950 023520 004737 023414 TRVBIF: JSR PC,TRVACT
3951 023524 105737 003325 TSTB P$GDBD ;SEE IF P$GDBD SET OR CLEARED BY ACTION
3952 023530 001402 BEQ 1$ ;IF CLEAR FALL THRU TO NEXT NODE
3953 023532 000137 023434 JMP TRVBRC ;ELSE TAKE THE 'MISS' BRANCH
3954 023536 000137 023444 1$: JMP TRVNOB ;JUST UPDATE TO NEXT NODE IF THINGS OK
3955
3956 023542 005005 TRVSPA: CLR R5 ;CLEAR 'SPACE OR TAB FOUND' FLAG
3957 023544 121427 000011 1$: CMPB (R4),#11 ;SEE IF CHAR. IN CMD LINE= TAB
3958 023550 001003 BNE 2$ ;BR IF NO, NOT A TAB
3959 023552 005204 INC R4 ;INC INPUT STRING POINTER
3960 023554 005205 INC R5 ;INDICATE A TAB FOUND
3961 023556 000772 BR 1$ ;GO CHECK NEXT CHAR
3962
3963 023560 121427 000040 2$: CMPB (R4),#40 ;SEE IF CHAR. IN CMD LINE= SPACE
3964 023564 001003 BNE 10$ ;BR IF NO, NON-SPACE OR NON-TAB CHAR.
3965 023566 005204 INC R4 ;INC INPUT STRING POINTER
3966 023570 005205 INC R5 ;INDICATE A SPACE FOUND
3967 023572 000764 BR 1$ ;GO CHECK NEXT CHAR
3968 023574 005705 10$: TST R5 ;SEE IF ANY SPACES OR TABS FOUND
3969 023576 001404 BEQ 15$ ;BR IF NO, TAKE NO ACTION
3970 023600 004737 023414 JSR PC,TRVACT ;GO TAKE ACTION IF ANY FOUND

```

```

3971 023604 000137 023444          JMP      TRVNOB          ;JUST GO UPDATE R3 TO NEXT NODE IF OK
3972 023610 000137 023434          JMP      TRVBRC          ;TAKE BRANCH (MISS) IF NONE FOUND
3973
3974
3975 023614 012737 000012 003322 TRVDEC: MOV      #10.,PSRADX      ;USE DECIMAL AS RADIX AND ASSUME +
3976 023622 000137 023634          JMP      TRVNMA
3977 023626          TRVOCT: ;(SAME AS TRVNUM SINCE DEFAULT RADIX IS OCTAL)
3978 023626 012737 000010 003322 TRVNUM: MOV      #8.,PSRADX      ;USE OCTAL AS RADIX AND ASSUME +
3979 023634 005005          TRVNMA: CLR      R5          ;CLEAR DIGIT COUNTER
3980 023636 121427 000053          CMPB     (R4),#'+'      ;SEE IF THERE'S A + SIGN THERE
3981 023642 001001          BNE      10$           ; BR IF NO
3982 023644 000406          BR       11$           ; ELSE PSRADX ALREADY SAYS +, JUST BR
3983 023646 121427 000055          10$:    CMPB     (R4),#'-'      ;SEE IF THERE'S A - SIGN THERE
3984 023652 001004          BNE      1$            ; BR IF NO
3985 023654 112737 177777 003323 MOVB     #-1,PSRADX+1    ;SET 'MINUS FLAG' (HI BYTE OF PSRADX)
3986 023662 005204          11$:    INC      R4          ;BUMP R4 TO POINT TO FIRST CHAR
3987
3988 023664 121427 000060          1$:    CMPB     (R4),#60      ;SEE IF CHAR. LESS THAN A '0'
3989 023670 002434          BLT     2$            ;BR IF YES (NOT NUMERIC)
3990 023672 121427 000067          CMPB     (R4),#67      ;SEE IF CHAR. GREATER THAN A '7'
3991 023676 003426          BLE     13$          ; BR IF YES
3992 023700 123727 003322 000012 CMPB     PSRADX,#10.    ;SEE IF IN DECIMAL MODE
3993 023706 001417          BEQ     12$          ; BR IF YES (CAN USE HIGHER LIMIT)
3994 023710 121427 000071          CMPB     (R4),#71      ;SEE IF DIGIT WAS A 8 OR 9
3995 023714 003022          BGT     2$            ;BR IF NON-NUMERIC
3996 023716          PRINTF #CLIBRX        ;ELSE WAS A 8 OR 9 WHEN IN OCTAL RADIX
3997 023716 012746 012217          MOV      #CLIBRX,-(SP)
3998 023722 012746 000001          MOV      #1,-(SP)
3999 023726 010600          MOV      SP,R0
4000 023730 104417          TRAP    C$PNTF
4001 023732 062706 000004          ADD     #4,SP
4002 023736 112737 177777 003325 MOVB     #-1,PSGDBD    ;SET ERROR RETURN FLAG
4003 023744 000474          BR      5$            ; PRINT ERROR AND TAKE MISS
4004
4005 023746 121427 000071          12$:    CMPB     (R4),#71      ;SEE IF CHAR. GREATER THAN A '9'
4006 023752 003003          BGT     2$            ;BR IF YES (NOT NUMERIC)
4007 023754 005204          13$:    INC      R4          ;UPDATE CMD LINE PTR TO NEXT CHAR.
4008 023756 005205          INC      R5          ;INDICATE A NUMERIC FOUND
4009 023760 000741          BR      1$            ;GO LOOK AT NEXT CHAR.
4010
4011 023762 005705          2$:    TST      R5          ;SEE IF FOUND ANY NUMERICS
4012 023764 001464          BEQ     5$            ;BR IF NO, TAKE 'MISS' BRANCH
4013 023766 010401          MOV     R4,R1          ;GET POINTER TO START OF NUMERIC STRING
4014 023770 160501          SUB     R5,R1
4015 023772 005037 003320          CLR     PSNUM          ;CLEAR LOC. WHERE VALUE WILL BE STORED
4016 023776 112102          3$:    MOVB     (R1)+,R2      ;GET ASCII CHAR AND CONVERT IT TO A #
4017 024000 162702 000060          SUB     #60,R2
4018 024004 006337 003320          ASL     PSNUM          ;SHIFT CURRENT VALUE TO MAKE ROOM
4019 024010 103437          BCS     7$            ;ERROR IF NUMBER TOO BIG
4020 024012 013737 003320 003316 MOV     PSNUM,PSCNT    ;SAVE FOR LATER IN CASE DECIMAL RADIX
4021 024020 006337 003320          ASL     PSNUM
4022 024024 103431          BCS     7$            ;ERROR IF NUMBER TOO BIG
4023 024026 006337 003320          ASL     PSNUM
4024 024032 103426          BCS     7$            ;ERROR IF NUMBER TOO BIG
4025 024034 123727 003322 000012 CMPB     PSRADX,#10.    ;SEE IF DECIMAL RADIX
4026 024042 001004          BNE     4$            ;BR IF NOT EQUAL

```

4027	024044	063737	003316	003320	ADD	PSCNT,PSNUM		
4028	024052	103416			BCS	7\$		:ERROR IF NUMBER TOO BIG
4029	024054	060237	003320		ADD	R2,PSNUM	4\$:	
4030	024060	103413			BCS	7\$		:ERROR IF NUMBER TOO BIG
4031	024062	005305			DEC	R5		
4032	024064	001344			BNE	3\$		
4033	024066	105737	003323		TSTB	PSRADX+1		:SEE IF NUM WAS PRECEDED BY A - SIGN
4034	024072	001402			BEQ	15\$		: BR IF NO
4035	024074	005437	003320		NEG	PSNUM		: ELSE NEGATE THE NUMBER BEFORE LEAVING
4036	024100	004737	023414		JSR	PC,TRVACT	15\$:	:SINCE NUMERIC FOUND, GO TAKE ACTION
4037	024104	000137	023444		JMP	TRVNOB		:GO POINT R3 TO NEXT NODE
4038								
4039	024110						7\$:	
4040	024110	012746	012175		PRINTF	#CLINBG		:PRINT NUMBER TOO BIG ERROR
4041	024114	012746	000001					MOV #CLINBG,-(SP)
4042	024120	010600						MOV #1,-(SP)
4043	024122	104417						MOV SP,R0
4044	024124	062706	000004					TRAP C\$PNTF
4045	024130	112737	177777	003325				ADD #4,SP
4046	024136	000137	023434		MOVB	#-1,PSGDBD	5\$:	:SET ERROR RETURN FLAG
4047					JMP	TRVBRC		:TAKE 'MISS' BRANCH
4048								
4049	024142	005005			TRVALP:	CLR R5		:CLEAR ALPHA FOUND FLAG
4050	024144	121427	000101		1\$:	CMPB (R4),#101		:SEE IF CHAR. LESS THAN A 'A'
4051	024150	002406			BLT	2\$		:BR IF YES (NOT ALPHA)
4052	024152	121427	000132		CMPB	(R4),#132		:SEE IF CHAR. GREATER THAN A 'Z'
4053	024156	003003			BGT	2\$		:BR IF YES (NOT ALPHA)
4054	024160	005204			INC	R4		:UPDATE CMD LINE PTR TO NEXT CHAR
4055	024162	005205			INC	R5		:INDICATE AN ALPHA WAS FOUND
4056	024164	000767			BR	1\$		:GO LOOK AT NEXT CHAR.
4057	024166	005705			2\$:	TST R5		:SEE IF ANY ALPHA'S WERE FOUND
4058	024170	001404			BEQ	3\$		:BR IF NO
4059	024172	004737	023414		JSR	PC,TRVACT		:IF ANY FOUND TAKE ACTION
4060	024176	000137	023444		JMP	TRVNOB		:THEN UPDATE R3 TO NEXT NODE -NO BRANCH
4061	024202	000137	023434		3\$:	JMP TRVBRC		:NONE FOUND, TAKE MISS BRANCH
4062								
4063	024206	005005			TRVALN:	CLR R5		:CLEAR ALPHANUM FOUND FLAG
4064	024210	121427	000060		10\$:	CMPB (R4),#60		:SEE IF CHAR. LESS THAN A '0'
4065	024214	002417			BLT	2\$		:BR IF YES (NOT NUMERIC OR ALPHA)
4066	024216	121427	000072		CMPB	(R4),#72		:SEE IF CHAR. GREATER THAN A '9'
4067	024222	003003			BGT	1\$		:BR IF YES (NOT NUMERIC)
4068	024224	005204			INC	R4		:UPDATE CMD LINE PTR TO NEXT CHAR.
4069	024226	005205			INC	R5		:INDICATE A NUMERIC FOUND
4070	024230	000767			BR	10\$		:GO LOOK AT NEXT CHAR.
4071	024232	121427	000101		1\$:	CMPB (R4),#101		:SEE IF CHAR. LESS THAN A 'A'
4072	024236	002406			BLT	2\$		:BR IF YES (NOT ALPHA)
4073	024240	121427	000132		CMPB	(R4),#132		:SEE IF CHAR. GREATER THAN A 'Z'
4074	024244	003003			BGT	2\$		:BR IF YES (NOT ALPHA)
4075	024246	005204			INC	R4		:UPDATE CMD LINE PTR TO NEXT CHAR
4076	024250	005205			INC	R5		:INDICATE AN ALPHA FOUND
4077	024252	000756			BR	10\$		:GO LOOK AT NEXT CHAR.
4078	024254	005705			2\$:	TST R5		:SEE IF ANY ALPHANUM'S WERE FOUND
4079	024256	001404			BEQ	3\$		:BR IF NO
4080	024260	004737	023414		JSR	PC,TRVACT		:IF ANY FOUND TAKE ACTION
4081	024264	000137	023444		JMP	TRVNOB		:THEN UPDATE R3 TO NEXT NODE -NO BRANCH
4082	024270	000137	023434		3\$:	JMP TRVBRC		:NONE FOUND, TAKE MISS BRANCH

```

4083
4084
4085
4086 024274 010401          TRVSTR: MOV      R4,R1          ;POINT R1 TO CMD STRING
4087 024276 010305          MOV      R3,R5
4088 024300 062705 000006   ADD      #6,R5          ;POINT R5 TO MATCH STRING FROM CLI NODE
4089 024304 005037 003316   CLR      P$CNT         ;CLEAR CHAR MATCH COUNT
4090 024310 105715          2$:  TSTB     (R5)        ;SEE IF END OF MATCH STRING YET
4091 024312 001411          BEQ     10$            ;BR IF YES
4092 024314 105711          TSTB     (R1)        ;SEE IF END OF CMD LINE YET
4093 024316 001407          BEQ     10$            ;BR IF YES
4094 024320 121115          CMPB     (R1),(R5)    ;SEE IF CHARACTERS MATCH
4095 024322 001005          BNE     10$            ;BR IF NO
4096 024324 005237 003316   INC     P$CNT         ;MATCH -INCREMENT MATCH COUNT
4097 024330 005201          INC     R1            ;UPDATE STRING POINTERS
4098 024332 005205          INC     R5
4099 024334 000765          BR      2$            ;BR TO CONTINUE CHECKING CHARS.
4100
4101 024336 005737 003316   10$:  TST     P$CNT     ;WHEN DONE SEE IF ANY MATCHES FOUND
4102 024342 001406          BEQ     15$            ;BR IF NO, GO TAKE THE MISS BRANCH
4103 024344 010104          MOV     R1,R4         ;POINT CMD POINTER TO END OF STRING &
4104 024346 004737 023414   JSR     PC,TRVACT     ;IF A MATCH FOUND, GO DO MATCH ACTION
4105 024352 066303 000004   ADD     4(R3),R3     ;UPDATE R3 TO NEXT NODE (NO BRANCH)
4106 024356 000207          RTS     PC            ; (NO RETURN THRU TRVNOB SINCE DIFFERNT
4107                                     ; DISPLACEMENT DUE TO MATCH STRING)
4108 024360 000137 023434   15$:  JMP     TRVBRC    ; GO TAKE BRANCH
4109
4110                                     ; (PARSED OK), -1 IF ILL CMD.....
4111
4112

```

4113  
4114  
4115  
4116  
4117  
4118  
4119  
4120  
4121 024364  
4122 024364  
4123  
4124  
4125 024364 004737 021406  
4126  
4127  
4128  
4129  
4130  
4131 024370  
4132 024370  
4133 024370 104425

.SBTTL REPORT CODING SECTION

:+  
: THE REPORT CODING SECTION CONTAINS THE  
: 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.  
:--

BGNRPT

LSRPT::

JSR PC,REPORT

;CALL SUBROUTINE TO DUMP EVENT LOG  
; AND BASE TABLE

ENDRPT

L10011: TRAP CSRPT

4134  
4135  
4136  
4137  
4138  
4139  
4140  
4141  
4142  
4143  
4144  
4145  
4146  
4147  
4148  
4149

.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

024372  
024372  
024372 177777  
024374 177777  
024376 177777  
024400

```

4150          .SBTTL  INITIALIZE SECTION
4151
4152          :++
4153          : THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
4154          : AT THE BEGINNING OF EACH PASS.
4155          :--
4156
4157          BGNINJT
4158          L$INIT::
4159
4160
4161          024400 012737 177777 007216      MOV    #-1,RESFLG      ;SET RESTART FLAG
4162          024406          REDEF  #EF.START      ;IF HERE CAUSE OF START,DO SOME INIT
4163          024406 012700 000040          MOV    #EF.START,RO
4164          024412 104447          TRAP   CSREFG
4165          024414          BCOMPLETE  START
4166          024414 103417          BCS    START
4167          024416          REDEF  #EF.RESTART      ;IF HERE CAUSE OF RESTART, DO SOME INIT
4168          024416 012700 000037          MOV    #EF.RESTART,RO
4169          024422 104447          TRAP   CSREFG
4170          024424          BCOMPLETE  RESTRT
4171          024424 103515          BCS    RESTRT
4172          024426          REDEF  #EF.CONTINUE      ;SEE IF WE'RE HERE CAUSE OF A CONTINUE
4173          024426 012700 000036          MOV    #EF.CONTINUE,RO
4174          024432 104447          TRAP   CSREFG
4175          024434          BNCOMPLETE  S1
4176          024434 103002          BCC    S1
4177          024436 000137 025146          JMP    ENDIT
4178          024442          S1: REDEF  #EF.NEW
4179          024442 012700 000035          ;SEE IF THIS IS A 'NEW PASS'
4180          024446 104447          MOV    #EF.NEW,RO
4181          024450          BCOMPLETE  NEW
4182          024450 103523          TRAP   CSREFG
4183          024452 000525          BCS    NEW
4184          024454          BR      GETPRM
4185          024454 005037 007216          START: CLR  RESFLG
4186          024460 005037 007256          ;CLEAR RESTART FLAG SINCE HERE ON START
4187          024464          ;CLEAR CLK VECTOR PTR. AS A FLAG IN
4188          024464 012702 007252          ; NO CLOCK IS FOUND.
4189          024470          MOV    #CLKCSR,R2
4190          024470 012700 000114          ;SETUP R2 AS A PTR. TO CLOCK INFO BLOCK
4191          024474 104462          ;LOOK FOR A LINE CLOCK
4192          024476 010001          MOV    #'L,RO
4193          024500          BNCOMPLETE  S2
4194          024500 103006          TRAP   CSCLK
4195          024502 004737 020544          MOV    RO,R1
4196          024506 012737 000100 007262          BCC    S2
4197          024514 000461          ; IF NONE THERE GO LOOK FOR A P-CLOCK
4198          024516          JSR    PC,CLKSET
4199          024516          ; GO SET UP CLOCK INFO TABLE & CLK VEC.
4200          024516 012700 000120          ;SETUP THE ENABLE LINE CLOCK DATA
4201          024522 104462          MOV    #'P,RO
4202          024524 010001          TRAP   CSCLK
4203          024526          BNCOMPLETE  S3
4204          024526 103017          MOV    RO,R1
4205          024530 004737 020544          BCC    S3
          ; IF NONE THERE GO SEE IF THIS IS LSI
          ; ELSE GO SET UP CLOCK INFO & VECTOR

```

```

4206 024534 062737 000002 007252      ADD    #2,CLKCSR      ;POINT CLKCSR TO P-CLK COUNT SET REG.
4207 024542 012777 001600 162502      MOV    #PCLKCT,@CLKCSR ;LOAD CLK SET REG. WITH COUNT VALUE
4208 02 550 162737 000002 007252      SUB    #2,CLKCSR      ;POINT CLKCSR BAC TO P-CLK CSR
4209 024556 012737 000111 007262      MOV    #PCLKEN,CLKEN  ;SETUP THE ENABLE THE P-CLK DATA
4210 024564 000435
4211
4212 024566          S3:  READBUS          ;READ BUS TYPE TO SEE IF ON AN LSI
4213 024566 104407          TRAP   CSRDBU
4214 024570          BNCOMPLETE      S4          ;BR IF NOT, NO CHANCE OF A CLOCK
4215 024570 103021          BCC   S4
4216 024572 012737 000100 007256      MOV    #100,CLKVEC    ;LOAD 100 AS CLK VECTOR
4217 024600 005037 007254          CLR    CLKBR          ;LOAD 0 AS CLK INT. LEVEL
4218 024604 012737 007262 007252      MOV    #CLKEN,CLKCSR ;KLUDGE UP THE CSR & ENABLE DATA LOCS
4219 024612          GMANID  L5060,CLKHZ,D,377,50.,60.,YES
4220 024612 104443
4221 024614 000406          TRAP   CS$GMAN
4222 024616 007260          BR    10000$
4223 024620 000052          .WORD CLKHZ
4224 024622 013602          .WORD T$CODE
4225 024624 000377          .WORD L5060
4226 024626 000062          .WORD 377
4227 024630 000074          .WORD T$LOLIM
4228 024632          .WORD T$HILIM
4229 024632 000412          10000$:
4230
4231 024634          S4:  PRINTF  #NOCLK          ;INFORM OPR. NO CLOCK, & EXIT INIT
4232 024634 012746 013712          MOV    #NOCLK,-(SP)
4233 024640 012746 000001          MOV    #1,-(SP)
4234 024644 010600          MOV    SP,R0
4235 024646 104417          TRAP  C$PNTF
4236 024650 062706 000004          ADD    #4,SP
4237 024654          EXIT  INIT
4238 024654 104432          TRAP  C$EXIT
4239 024656 000404          .WORD  L10013-.
4240
4241 024660 005037 007264          RESTRT: CLR  TIMMIN          ;CLEAR TIME SINCE START LOCATIONS
4242 024664 005037 007266          CLR  TIMSEC
4243 024670 013737 007260 007270      MOV    CLKHZ,TIMTCK    ;LOAD TICKS/SEC
4244 024676 012702 007302          MOV    #EVTLOG,R2     ;INIT EVENT TABLE TO ALL 1'S AFTER EACH
4245 024702 010237 007300          MOV    R2,EVTPTR     ; START OR RES AND INIT TABLE POINTER
4246 024706 012722 177777          1$:  MOV    #-1,(R2)+
4247 024712 020227 010204          CMP   R2,#EVTEND
4248 024716 001373          BNE   1$             ;SEE IF REACHED END OF TABLE
4249
4250 024720 012737 177777 007212      NEW:  MOV    #-1,LOGUNT ;INITIALIZE LOGICAL UNIT #
4251
4252 024726 005237 007212          GETPRM: INC  LOGUNT      ;POINT TO NEXT LOGICAL UNIT
4253 024732 023737 007212 002012      CMP   LOGUNT,L$UNIT  ;SEE IF PAST MAX. LOG. UNIT #
4254 024740 002367          BGE   NEW           ;BR IF YES, AND START OVER
4255
4256 024742          GPHARD LOGUNT,R1    ;GET THE P-TABLE FOR THIS LOG. UNIT
4257 024742 013700 007212          MOV    LOGUNT,R0
4258 024746 104442          TRAP  C$GPHRD
4259 024750 010001          MOV    R0,R1
4260 024752          BNCOMPLETE      GETPRM ;IF NO P-TABLE AVAIL., GO GET NEXT ONE
4261 024752 103365          BCC   GETPRM

```



```

4262
4263 024754 011137 007224      MOV      (R1),FHDPLX          ;PUT FULL OR HALF DUPLEX ANSWER IN LOC.
4264
4265
4266                          ;DEVICE DEPENDENT PART OF GETTING INFO FROM P-TABLE
4267
4268 024760 016137 000002 011776      MOV      2(R1),SELO          ;STORE AWAY CSR ADDRESSES
4269 024766 016137 000002 012000      MOV      2(R1),BSEL1
4270 024774 005237 012000      INC      BSEL1
4271 025000 016137 000002 012002      MOV      2(R1),SEL2
4272 025006 062737 000002 012002      ADD      #2,SEL2
4273 025014 016137 000002 012004      MOV      2(R1),BSEL3
4274 025022 062737 000003 012004      ADD      #3,BSEL3
4275 025030 016137 000002 012006      MOV      2(R1),SEL4
4276 025036 062737 000004 012006      ADD      #4,SEL4
4277 025044 016137 000002 012010      MOV      2(R1),BSEL5
4278 025052 062737 000005 012010      ADD      #5,BSEL5
4279 025060 016137 000002 012012      MOV      2(R1),SEL6
4280 025066 062737 000006 012012      ADD      #6,SEL6
4281 025074 016137 000002 012014      MOV      2(R1),BSEL7
4282 025102 062737 000007 012014      ADD      #7,BSEL7
4283
4284 025110 016137 000004 012016      MOV      4(R1),INVEC        ;STORE AWAY INPUT INTERRUPT VECTOR
4285 025116 016137 000004 012020      MOV      4(R1),OUTVEC
4286 025124 062737 000004 012020      ADD      #4,OUTVEC          ;BUILD OUTPUT INTERRUPT VECTOR
4287 025132 016137 000006 012022      MOV      6(R1),INTPRI       ;STORE AWAY INTERRUPT PRIORITY
4288 025140 016137 000012 012024      MOV      12(R1),OPTYP       ;STORE AWAY DEVICE OPTION TYPE
4289
4290 025146                          ENDIT:
4291 025146      SETVEC  CLKVEC,#CLKINT,#340  ;SETUP CLOCK VECTOR
4292 025146 012746 000340
4293 025152 012746 020570      MOV      #340,-(SP)
4294 025156 013746 007256      MOV      #CLKINT,-(SP)
4295 025162 012746 000003      MOV      CLKVEC,-(SP)
4296 025166 104437
4297 025170 062706 000010      MOV      #3,-(SP)
4298                          TRAP    C$SVEC
4299                          ADD     #10,SP
4300                          ;DEVICE DEPENDENT VECTOR SETUP
4301 025174      SETVEC  INVEC,#DVINS,INTPRI  ;SETUP INPUT INTERRUPT VECTOR
4302 025174 013746 012022
4303 025200 012746 034700      MOV      INTPRI,-(SP)
4304 025204 013746 012016      MOV      #DVINS,-(SP)
4305 025210 012746 000003      MOV      INVEC,-(SP)
4306 025214 104437
4307 025216 062706 000010      MOV      #3,-(SP)
4308 025222      SETVEC  OUTVEC,#DVOUTS,INTPRI ;SETUP OUTPUT INTERRUPT VECTOR
4309 025222 013746 012022
4310 025226 012746 034710      MOV      C$SVEC
4311 025232 013746 012020      ADD     #10,SP
4312 025236 012746 000003      MOV      INTPRI,-(SP)
4313 025242 104437
4314 025244 062706 000010      MOV      #DVOUTS,-(SP)
4315                          MOV     OUTVEC,-(SP)
4316 025250      SETPRI  #PRI00              ;SET THE 'RUN' PRIORITY TO 0
4317 025250 012700 000000      MOV     #3,-(SP)
                          TRAP    C$SVEC
                          ADD     #10,SP
                          MOV     #PRI00,R0
  
```

4318 025254 104441  
4319 025256  
4320 025256 104432  
4321 025260 000002  
4322  
4323  
4324  
4325  
4326 025262  
4327 025262  
4328 025262 104411

EXIT INIT  
  
.EVEN  
ENDINIT

TRAP C\$SPRI  
TRAP C\$EXIT  
.WORD L10013-

L10013:  
TRAP C\$INIT

4329  
4330  
4331  
4332  
4333  
4334  
4335  
4336  
4337  
4338 025264  
4339 025264  
4340  
4341  
4342 025264  
4343 025264  
4344 025264 104461

.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.  
:--

BGNAUTO

L\$AUTO::

ENDAUTO

L10014: TRAP C\$AUTO

4345  
4346  
4347  
4348  
4349  
4350  
4351  
4352 025266  
4353 025266  
4354  
4355 025266 005077 161760  
4356 025272  
4357 025272 012700 000340  
4358 025276 104441  
4359 025300  
4360 025300 104433  
4361  
4362 025302  
4363 025302 104432  
4364 025304 000002  
4365  
4366  
4367  
4368  
4369 025306  
4370 025306  
4371 025306 104412

.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  
SETPRI #PRI07

;DISABLE CLOCK  
;SET PROCESSOR PRIORITY BACK TO 7

MOV #PRI07,RO  
TRAP C\$SPRI

BRESET

;CLEAR ALL BEFORE END

TRAP C\$RESET

EXIT CLN

TRAP C\$EXIT  
.WORD L10015-

.EVEN

ENDCLN

L10015:

TRAP C\$CLEAN

4372  
4373  
4374  
4375  
4376  
4377  
4378  
4379 025310  
4380 025310  
4381  
4382  
4383 025310  
4384 025310 000167  
4385 025312 000000  
4386  
4387  
4388  
4389  
4390 025314  
4391 025314  
4392 025314 104453

.SBTTL DROP UNIT SECTION

:+  
: THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE  
: TO NO LONGER BE TESTED.  
:--

BGNDU

LSDU::

EXIT DU

.WORD JSJMP  
.WORD L10016-2-

.EVEN

ENDDU

L10016:  
TRAP CSDU

4393  
4394  
4395  
4396  
4397  
4398  
4399  
4400  
4401 025316  
4402 025316  
4403  
4404  
4405 025316  
4406 025316 000167  
4407 025320 000000  
4408  
4409  
4410  
4411  
4412 025322  
4413 025322  
4414 025322 104452  
4415  
4416

.SBTTL ADD UNIT SECTION

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

BGNAU

LSAU::

EXIT AU

.WORD JSJMP  
.WORD L10017-2-

.EVEN

ENDAU

L10017:  
TRAP CSAU

4417  
4418  
4419  
4420  
4421  
4422  
4423  
4424  
4425  
4426  
4427  
4428  
4429  
4430  
4431  
4432  
4433  
4434  
4435  
4436  
4437  
4438  
4439  
4440  
4441  
4442  
4443  
4444  
4445  
4446  
4447  
4448  
4449  
4450  
4451  
4452  
4453  
4454  
4455  
4456  
4457  
4458  
4459  
4460  
4461  
4462  
4463  
4464  
4465  
4466  
4467  
4468  
4469  
4470  
4471  
4472

025324  
025324  
  
025324 013777 007262 161720  
  
025332  
025332 005001  
025334 012737 000001 007272  
025342 005737 007272  
025346 001412  
025350 005301  
025352 001373  
025354  
025354 012746 013712  
025360 012746 000001  
025364 010600  
025366 104417  
025370 062706 000004  
  
025374 005737 007216  
025400 001117  
  
  
025402 005037 007166  
025406 005037 007122  
025412 005037 007106  
025416 012701 006326  
025422 010137 007100  
025426 005037 007076  
025432 012702 000017  
025436 006302  
025440 006302  
025442 010137 007102  
025446 060237 007102  
  
025452 012737 000005 007154  
025460 013737 002162 007156  
025466 012737 003326 007124  
025474 012737 005326 007110  
  
025502 013737 007124 007164

```

.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.
:--

BGNTST

T1::

.SBTTL PROGRAM SETUP SECTION

MOV CLKEN,@CLKCSR ;ENABLE THE CLOCK

GTXRXB:
GTRA2: CLR R1
MOV #1,TIMER1 ;SET TIMER TO COUNT 1 TICK
TST TIMER1 ;CHECK FOR IT TO BE COUNTED OFF
BEQ GTRA3 ;BRANCH IF CLOCK EXISTS (COUNTED A TICK)
DEC R1
BNE 1$ ;KEEP CHECKING UNTIL R1 DOES FULL COUNTDOWN
PRINTF #NOCLK ;PRINT BAD CLK MSG AND WARN OF HANG IF TIMEOUT
MOV #NOCLK,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #4,SP

GTRA3: TST RESFLG ;SEE IF HERE AFTER A RESTART.
BNE GTRA5 ;BR IF HERE CAUSE OF A RESTART

; CLEAR COUNTS AND SET UP DEFAULTS

GTRA4: CLR TOTCC ;CLEAR TOTAL CHAR. COUNT TEMP. LOC.
CLR TTOTCC ; CLEAR TOTAL CHAR. COUNT FOR TX BUFF
CLR CTOTCC ; CLEAR TOTAL CHAR. COUNT FOR CMP BUFF
MOV #PTRTAB,R1 ;INIT TRANSMIT MESSAGE POINTER
MOV R1, TXPTR
CLR RXPTR ; ZERO RX POINTER
MOV #MSGLIM,R2
ASL R2
ASL R2
MOV R1,CMPPTR
ADD R2,CMPPTR ;INIT COMPARE MESSAGE POINTER

MOV #5,MSGTYP ;SET UP DEFAULT MSG TYPE (QUICK FOX - ITEMP MSG)
MOV MSG5C,CURCC ;SET UP DEFAULT CHAR COUNT
MOV #TXBUF,TCURAD ;SET UP CURRENT ADD TO START OF TX BUFFER
MOV #CMPBUF,CCURAD ;SET UP CURRENT ADD TO START OF CMP BUFFER

MOV TCURAD,CURADD ;SETUP CURRENT ADDR TO START OF TXBUF

```

```

4473 025510 013737 007100 007162      MOV      TXPTR,CPTR      ;SETUP CURRENT POINTER TABLE POINTER FOR TXBUF
4474 025516 004737 022676              JSR      PC,BLDBUF      ; GO BUILD POINTER TABLE AND BUFFER
4475 025522 012737 000001 007120      MOV      #1,TXMTOT      ;BUMP TOTAL MESSAGE COUNT
4476
4477 025530 013737 007102 007162      MOV      CMPPTR,CPTR    ;SET UP START OF COMPARE POINTER TABLE
4478 025536 013737 007110 007164      MOV      CCRAD,CURADD   ;SET UP CURRENT ADDR. TO START OF CMPBUF
4479 025544 012737 000005 007154      MOV      #5,MSGTYP
4480 025552 013737 002162 007156      MOV      MSG5C,CURCC
4481 025560 004737 022676              JSR      PC,BLDBUF      ;PUT DEFAULT MESSAGE INTO CMPBUF
4482 025564 012737 000001 007104      MOV      #1,CMPTOT      ;BUMP THE COMP MESSG COUNT
4483 025572 012737 000003 007220      MOV      #ACT,MODTYP    ;SET DEFAULT MODE= ACTIVE
4484 025600 005037 007222              CLR      MLTYP          ;SET DEFAULT MAINTENANCE LOOP MODE =NONE
4485 025604 012737 000001 007230      MOV      #1,RPASS      ;SET UP DEFAULT 'RUN PASS' COUNT TO 1
4486 025612 012737 000002 007226      MOV      #2,PARAM      ;SET UP PROG. PARAMETERS - DATACHECKING ENABLED
4487
4488 025620              PRINTF  #HLP0          ;
4489 025620 012746 012442              MOV      #HLP0,-(SP)
4490 025624 012746 000001              MOV      #1,-(SP)
4491 025630 010600              MOV      SP,RO
4492 025632 104417              TRAP    C$PNTF
4493 025634 062706 000004              ADD     #4,SP
4494 025640 013737 007220 010334  GTRAS: MOV      MODTYP,DEV1
4495 025646 013737 007222 010336      MOV      MLTYP,DEV2
4496 025654 013737 007230 010340      MOV      RPASS,DEV3
4497 025662 013737 007226 010342      MOV      PARAM,DEV4
4498 025670 004737 023022              JSR      PC,SHWOP
4499
4500 025674              MANUAL              ;SEE IF MANUAL INTERVENTION ALLOWED
4501 025674 104450              BCOMPLETE  GETCL    ; BR IF YES (UAM=0 AND NOT CHAINED)
4502 025676              TRAP    C$MANI
4503 025676 103412              BCS      GETCL
4504 025700 005737 007230      TST      RPASS        ;SEE IF THIS IS FIRST 'DCLT PASS'
4505 025704 001002              BNE      1$           ; BR IF NOT COMPLETED 1 PASS
4506 025706              EXIT                ; IF DONE 1 PASS IN UNATTENDED MODE - EXIT
4507 025706 104432              TRAP    C$EXIT
4508 025710 010050              .WORD   L10020-
4509 025712 012737 000001 007222  1$:  MOV      #TTL,MLTYP    ;SET UP DEFAULT FOR UNATTENDED MODE
4510 025720 000137 030634      JMP      GTR9         ; 'R M=ACT/LO=I/PAS=1/NOST/CH' AND RUN
4511
4512              .SBTTL              COMMAND LINE FETCH & INTERPRETATION SECTION
4513
4514 025724 105037 003325      GETCL:  CLRB   PSGDBD    ;CLEAR CMD LINE PARSING ERROR FLAGS
4515 025730 105037 003324      CLRB   PSNUF
4516 025734              GMANID  CLISPM,CMDBUF,A,0,1,72.,NO ;GET A COMMAND LINE FROM OPR.
4517 025734 104443              TRAP    C$GMAN
4518 025736 000406              BR      10000$
4519 025740 003062              .WORD  CMDBUF
4520 025742 000142              .WORD  TSCODE
4521 025744 012114              .WORD  CLISPM
4522 025746 000000              .WORD  0
4523 025750 000001              .WORD  T$LOLIM
4524 025752 000110              .WORD  T$HILIM
4525 025754              10000$:
4526 025754 012737 003062 003310      MOV      #CMDBUF,PSBUFA
4527 025762 012737 010344 003312      MOV      #CLITRE,PS$TREE
4528 025770 012737 026664 003314      MOV      #CLIACT,PS$ACT

```



4529	025776	005037	003206		CLR	QUALFG		:CLEAR QUALIFIER FLAG LOCATION
4530	026002	004737	023274		JSR	PC,PSTRV		:GO PARSE COMMAND LINE
4531	026006	105737	003325		TSTB	PSGDBD		:SEE IF PARSED OK OR AN ERROR
4532	026012	001412			BEQ	1\$		
4533	026014				PRINTF	#CLIERM		
4534	026014	012746	012122					
4535	026020	012746	000001				MOV	#CLIERM,-(SP)
4536	026024	010600					MOV	#1,-(SP)
4537	026026	104417					MOV	SP,RO
4538	026030	062706	000004				TRAP	C\$PNTF
4539	026034	000137	025724				ADD	#4,SP
4540	026040	105737	003324	1\$:	JMP	GETCL		
4541	026044	001412			TSTB	PSANUF		:SEE IF INCOMPLETE COMMAND TYPED
4542	026046				BEQ	10\$		
4543	026046	012746	012152		PRINTF	#CLINUF		
4544	026052	012746	000001				MOV	#CLINUF,-(SP)
4545	026056	010600					MOV	#1,-(SP)
4546	026060	104417					MOV	SP,RO
4547	026062	062706	000004				TRAP	C\$PNTF
4548	026066	000137	025724				ADD	#4,SP
4549					JMP	GETCL		
4550	026072	023727	003204	000005	10\$:	CMP	KEYWD1,#HLP	:SEE IF HELP WAS TYPED
4551	026100	001711			BEQ	GETCI		:GO GET CMD AGAIN IF YES
4552	026102	023727	003204	000055	CMP	KEYWD1,#PRNT		:SEE IF PRINT WAS TYPED
4553	026110	001705			BEQ	GETCL		:GO GET CMD AGAIN IF YES
4554	026112	023727	003204	000004	CMP	KEYWD1,#RUN		:SEE IF RUN WAS TYPED
4555	026120	001002			BNE	11\$		: BR IF NO
4556	026122	000137	030634		JMP	GTR9		: START EXEC. IF YES
4557	026126	023727	003204	000052	11\$:	CMP	KEYWD1,#DMPS	:SEE IF DUMP WAS TYPED
4558	026134	001004			BNE	12\$		: BR IF NO
4559	026136	004737	022442		JSR	PC,DUMPSR		: ELSE, DUMP PART OF MEMORY
4560	026142	000137	025724		JMP	GETCL		: THEN RETURN TO GET ANOTHER CMD.
4561	026146	023727	003204	000001	12\$:	CMP	KEYWD1,#CLEAR	:SEE IF CLEAR WAS TYPED
4562	026154	001663			BEQ	GETCL		: IF YES, BACK TO GET ANOTHER CMD.
4563	026156	023727	003204	000002	CMP	KEYWD1,#SHOW		:SEE IF SHOW WAS TYPED
4564	026164	001657			BEQ	GETCL		: IF YES, BACK TO GET ANOTHER CMD.
4565	026166	023727	003204	000010	4\$:	CMP	KEYWD1,#SETEXP	:SEE IF SET EXPECTED
4566	026174	001512			BEQ	2\$		: BR IF YES (A SETEXP WAS TYPED)
4567	026176	013737	007122	007166	5\$:	MOV	TTOTCC,TOTCC	
4568	026204	023727	007166	001000	CMP	TOTCC,#BUFLIM		:SEE IF BUFFER ALREADY FULL
4569	026212	002414			BLT	15\$		: BR IF NOT FULL (BUFLIM # OF CHARS.)
4570	026214				PRINTF	#MSGTRN,#BUFEX		: ELSE TELL OPR. AND DON'T BUILD MSG.
4571	026214	012746	014033				MOV	#BUFEX,-(SP)
4572	026220	012746	014051				MOV	#MSGTRN,-(SP)
4573	026224	012746	000002				MOV	#2,-(SP)
4574	026230	010600					MOV	SP,RO
4575	026232	104417					TRAP	C\$PNTF
4576	026234	062706	000006				ADD	#6,SP
4577	026240	000137	025724		JMP	GETCL		: THEN GO GET A NEW COMMAND
4578	026244	005737	007122	15\$:	TST	TTOTCC		:IF FIRST 'SET' THEN GET RID OF DEFAULT
4579	026250	001002			BNE	6\$		
4580	026252	005037	007120		CLR	TXMTOT		
4581	026256	012737	006326	007100	6\$:	MOV	#PTRTAB,TXPTR	:GET POSITION OF END OF TX LIST
4582	026264	013701	007120		MOV	TXMTOT,R1		
4583	026270	020127	000017		CMP	R1,#MSG LIM		:SEE IF MSG COUNT EXCEEDED.
4584	026274	002414			BLT	17\$		: BR IF NO

Line	Address	Offset	Value	Label	Instruction	Comment	
4585	026276						
4586	026276	012746	013773				
4587	026302	012746	014051				
4588	026306	012746	000002				
4589	026312	010600					
4590	026314	104417					
4591	026316	062706	000006				
4592	026322	000137	025724				
4593	026326	006301		17\$:	JMP GETCL	; THEN GO GET A NEW COMMAND.	
4594	026330	006301			ASL R1	;# OF MSGS *4 = NEXT FREE PTR BLOCK	
4595	026332	060137	007100		ASL R1		
4596	026336	013737	007100	007162	ADD R1, TXPTR		
4597	026344	013737	007124	007164	MOV TXPTR, CPTR	; SETUP CHAR. COUNT, CURRENT ADDR, & PTR	
4598	026352	004737	022600		MOV TCURAD, CURADD		
4599	026356	004737	022676		JSR PC, ADDCC	; ADD IN CHAR. COUNT AND CHECK TOTAL	
4600	026362	013737	007162	007100	JSR PC, BLDBUF	; GO BUILD MESSAGE IN BUFFER AND PTRS.	
4601	026370	013737	007166	007122	MOV CPTR, TXPTR		
4602	026376	013737	007164	007124	MOV TOTCC, TTOTCC	; UPDATE CHAR. COUNT, CURR ADDR, & PTR	
4603	026404	005237	007120		MOV CURADD, TCURAD		
4604	026410	005337	003210		INC TXMTOT		
4605	026414	001270			DEC QUALVL	; DEC THE COPY COUNT	
4606	026416	000137	025724		BNE 5\$		
4607					JMP GETCL		
4608	026422	013737	007106	007166	2\$:	MOV CTOTCC, TOTCC	; SETUP CHAR. COUNT, CURR. ADDR. & PTR
4609	026430	023727	007166	001000	CMP TOTCC, #BUFLIM	; SEE IF BUFFER ALREADY FULL	
4610	026436	002414			BLT 16\$	; BR IF NOT FULL (BUFLIM # OF CHARS.)	
4611	026440				PRINTF #MSGTRN, #BUFEX	; ELSE TELL OPR. AND DON'T BUILD MSG.	
4612	026440	012746	014033				
4613	026444	012746	014051				
4614	026450	012746	000002				
4615	026454	010600					
4616	026456	104417					
4617	026460	062706	000006				
4618	026464	000137	025724				
4619	026470	005737	007106				
4620	026474	001002		16\$:	JMP GETCL	; THEN GO GET A NEW COMMAND	
4621	026476	005037	007104		TST CTOTCC	; IF FIRST 'SET' THEN GET RID OF DEFAULT	
4622	026502	012701	006326		BNE 7\$		
4623	026506	012702	000017		CLR CMPTOT		
4624	026512	006302		7\$:	MOV #PTRTAB, R1		
4625	026514	006302			MOV #MSGGLIM, R2		
4626	026516	010137	007102		ASL R2		
4627	026522	060237	007102		ASL R2		
4628	026526	013701	007104		MOV R1, CMPPTR	; INIT COMPARE MESSAGE POINTER	
4629	026532	020127	000017		ADD R2, CMPPTR		
4630	026536	002414			MOV CMPTOT, R1		
4631	026540				CMP R1, #MSGGLIM	; SEE IF MSG COUNT EXCEEDED.	
4632	026540	012746	013773		BLT 18\$	; BR IF NO	
4633	026544	012746	014051		PRINTF #MSGTRN, #TABEX	; ELSE TELL OPR. AND DON'T BUILD MSG.	
4634	026550	012746	000002				
4635	026554	010600					
4636	026556	104417					
4637	026560	062706	000006				
4638	026564	000137	025724				
4639	026570	006301					
4640	026572	006301		18\$:	JMP GETCL	; THEN GO GET A NEW COMMAND.	
					ASL R1	;# OF MSGS *4 = NEXT FREE PTR BLOCK	
					ASL R1		

4641 026574 060137 007102  
4642 026600 013737 007102 007162  
4643 026606 013737 007110 007164  
4644 026614 004737 022600  
4645 026620 004737 022676  
4646 026624 013737 007162 007102  
4647 026632 005237 007104  
4648 026636 013737 007164 007110  
4649 026644 013737 007166 007106  
4650 026652 005337 003210  
4651 026656 001261  
4652 026660 000137 025724  
4653  
4654  
4655  
4656  
4657

ADD R1,CMPPTR  
MOV CMPPTR,CPTR  
MOV CCURAD,CURADD  
JSR PC,ADDCC ;ADD IN XHAR. COUNT AND CHECK TOTAL  
JSR PC,BLDBUF  
MOV CPTR,CMPPTR  
INC CMPTOT  
MOV CURADD,CCURAD ;UPDATE CHAR. COUNT, CURR ADDR. & PTR  
MOV TOTCC,CTOTCC  
DEC QUALVL ;IF COPY WAS GIVEN, PUT MSG IN BUFF  
BNE 2\$ ; AGAIN  
JMP GETCL ;GO BACK UNTIL GET A 'RUN'

4658					
4659					
4660			.SBTTL	ACTION TABLE AND ROUTINES	
4661			:	USER MUST CLEAR/SET PSGDBD IF USE 'CLIBIF' IN CONNECTION WITH ACTION	
4662	026664		:	R2 WILL HOLD ACTION CODE FROM PARSING (CLI) NODE	
4663	026664	006302	CLIACT:	ASL	R2
4664	026666	016202		MOV	10\$(R2),R2 ;FORM ADDRESS OF ACTION ROUTINE
4665	026672	062702		ADD	#10\$,R2
4666	026676	004712		JSR	PC,(R2)
4667	026700	000207		RTS	PC
4668					
4669	026702	000142	10\$:	.WORD	ACTNUL-10\$
4670	026704	000144		.WORD	ACTCLR-10\$
4671	026706	000154		.WORD	ACTSHO-10\$
4672	026710	001530		.WORD	ACTCHK-10\$
4673	026712	000244		.WORD	ACTRUN-10\$
4674	026714	000164		.WORD	ACTHLP-10\$
4675	026716	000270		.WORD	ACTCSE-10\$
4676	026720	000416		.WORD	ACTCST-10\$
4677	026722	000740		.WORD	ACTSTE-10\$
4678	026724	000750		.WORD	ACTSTT-10\$
4679	026726	000766		.WORD	ACTSZE-10\$
4680	026730	000776		.WORD	ACTCOP-10\$
4681	026732	001006		.WORD	ACTNUM-10\$
4682	026734	001100		.WORD	ACTOPM-10\$
4683	026736	001536		.WORD	ACTSTS-10\$
4684	026740	001120		.WORD	ACTEQO-10\$
4685	026742	001200		.WORD	ACTMSO-10\$
4686	026744	001206		.WORD	ACTMS1-10\$
4687	026746	001216		.WORD	ACTMS2-10\$
4688	026750	001226		.WORD	ACTMS3-10\$
4689	026752	001236		.WORD	ACTMS4-10\$
4690	026754	001246		.WORD	ACTMS5-10\$
4691	026756	001264		.WORD	ACTMS6-10\$
4692	026760	001314		.WORD	ACTATV-10\$
4693	026762	001324		.WORD	ACTPAS-10\$
4694	026764	001344		.WORD	ACTREC-10\$
4695	026766	001352		.WORD	ACTLIS-10\$
4696	026770	001362		.WORD	ACTDLL-10\$
4697	026772	001372		.WORD	ACTTRA-10\$
4698	026774	001402		.WORD	ACTTAL-10\$
4699	026776	001430		.WORD	ACTNO-10\$
4700	027000	001440		.WORD	ACTECH-10\$
4701	027002	001544		.WORD	ACTCRC-10\$
4702	027004	001552		.WORD	ACTPRO-10\$
4703	027006	001604		.WORD	ACTRPS-10\$
4704	027010	001614		.WORD	ACTMOP-10\$
4705	027012	001624		.WORD	ACTTLP-10\$
4706	027014	001634		.WORD	ACTCLP-10\$
4707	027016	001644		.WORD	ACTLLP-10\$
4708	027020	001654		.WORD	ACTRLP-10\$
4709	027022	000134		.WORD	ACTNUF-10\$
4710	027024	001156		.WORD	ACTBCR-10\$
4711	027026	000674		.WORD	ACTDMS-10\$
4712	027030	000724		.WORD	ACTDME-10\$
4713	027032	000716		.WORD	ACTDMQ-10\$

CZCLKAO DMR,DMC-11 DATA COMM. LINK TEST  
CZCLKA.P11 18-APR-80 09:24

MACY11 30A(1052) <sup>E 9</sup> 18-APR-80 09:24 PAGE 109  
ACTION TABLE AND ROUTINES

SEQ 0108

714 027034 000230  
15  
4716

.WORD ACTPRT-10\$

4717										
4718	027036	112737	177777	003324	ACTNUF: MOV	#-1,PSNNUF				:SET FLAG TO SAY NEED MORE OF COMMAND
4719	027044	000207			ACTNUL: RTS	PC				:RETURN TO PARSER
4720										
4721	027046	012737	000001	003204	ACTCLR: MOV	#CLEAR,KEYWD1				:SET LOC TO SAY A CLEAR WAS TYPED
4722	027054	000207			RTS	PC				
4723										
4724	027056	012737	000002	003204	ACTSHO: MOV	#SHOW,KEYWD1				:SET LOC. TO SAY A SHOW WAS TYPED
4725	027064	000207			RTS	PC				
4726										
4727	027066	012702	003212		ACTHLP: MOV	#HLPTAB,R2				:SETUP R2 AS A POINTER TO HELP MSG TABLE
4728	027072				1\$: PRINTF	#HLPF,(R2)+				:PRINT HELP INFORMATION MESSAGES
4729	027072	012246								MOV (R2)+,-(SP)
4730	027074	012746	012520							MOV #HLPF,-(SP)
4731	027100	012746	000002							MOV #2,-(SP)
4732	027104	010600								MOV SP,R0
4733	027106	104417								TRAP C\$PNTF
4734	027110	062706	000006							ADD #6,SP
4735	027114	020227	003230		CMP	R2,#HLPEND				:SEE IF ALL INFO PRINTED YET
4736	027120	001364			BNE	1\$				:IF NO KEEP PRINTING
4737	027122	012737	000005	003204	MOV	#HLP,KEYWD1				:SET LOC. TO SAY A HELP WAS TYPED
4738	027130	000207			RTS	PC				
4739										
4740	027132	012737	000055	003204	ACTPRT: MOV	#PRNT,KEYWD1				:SET LOC. TO SAY A HELP WAS TYPED
4741	027140	004737	021406		JSR	PC,REPORT				:CALL ROUTINE TO PRINT EVENT LOG AND BASE TABLE
4742	027144	000207			RTS	PC				
4743										
4744	027146	012737	000004	003204	ACTRUN: MOV	#RUN,KEYWD1				:SET RUN FLAG
4745	027154	112737	177777	003324	MOV	#-1,PSNNUF				:SET FLAG TO SAY NEED MORE OF COMMAND
4746	027162	012737	000001	007230	MOV	#1,RPASS				:SET DEFAULT RUN 'PASS' TO 1
4747	027170	000207			RTS	PC				
4748										
4749	027172	012701	006326		ACTCSE: MOV	#PTRTAB,R1				
4750	027176	012702	000017		MOV	#MSGLIM,R2				
4751	027202	006302			ASL	R2				
4752	027204	006302			ASL	R2				
4753	027206	010137	007102		MOV	R1,CMPPTR				
4754	027212	060237	007102		ADD	R2,CMPPTR				:INIT COMPARE MESSAGE POINTER
4755	027216	013701	007102		MOV	CMPPTR,R1				
4756										
4757	027222	013702	007104		MOV	CMPTOT,R2				
4758	027226	105037	003324		CLRB	PSNNUF				:FLAG THAT HAVE VALID COMMAND AT THIS PT.
4759	027232	023727	003204	000002	CMP	KEYWD1,#SHOW				:SEE IF A CLEAR OR SHOW WAS TYPED
4760	027240	001500			BEQ	ACTSHW				:BR IF A SHOW WAS TYPED
4761	027242	012737	000001	007104	MOV	#1,CMPTOT				:CLEAR COMPARE MESSAGE COUNT, CHAR. COUNT
4762	027250	005037	007106		CLR	CTOTCC				: AND RESET POINTER
4763										
4764	027254	012701	006326		MOV	#PTRTAB,R1				
4765	027260	012702	000017		MOV	#MSGLIM,R2				
4766	027264	006302			ASL	R2				
4767	027266	006302			ASL	R2				
4768	027270	010137	007102		MOV	R1,CMPPTR				
4769	027274	060237	007102		ADD	R2,CMPPTR				:INIT COMPARE MESSAGE POINTER
4770	027300	013737	007102	007162	MOV	CMPPTR,CPTR				:SET UP TO FILL IN DEFAULT MESSAGE
4771	027306	012701	005326		MOV	#CMPBUF,R1				
4772	027312	010137	007110		MOV	R1,CCURAD				

4773	027316	000431		BR	ACTCLB		
4774							
4775	027320	012701	006326	ACTCST:	MOV #PTRTAB,R1		
4776	027324	013702	007120		MOV TXMTOT,R2		
4777	027330	105037	003324		CLRB PSNUF		:FLAG THAT HAVE VALID COMMAND AT THIS PT.
4778	027334	023727	003204	000002	CMP KEYWD1,#SHOW		:SEE IF A CLEAR OR SHOW WAS TYPED
4779	027342	001437			BEQ ACTSHW		:BR IF A SHOW WAS TYPED
4780	027344	012737	000001	007120	MOV #1,TXMTOT		:CLEAR TRANSMIT MESSAGE COUNT, CHAR. COUNT
4781	027352	005037	007122		CLR TTOTCC		: AND RESET POINTER
4782	027356	012737	006326	007100	MOV #PTRTAB, TXPTR		
4783	027364	013737	007100	007162	MOV TXPTR, CPTR		
4784	027372	012701	003326		MOV #TXBUF, R1		
4785	027376	010137	007124		MOV R1,TCURAD		
4786							
4787	027402	012702	001000		ACTCLB: MOV #BUFLIM,R2		
4788	027406	010137	007164		MOV R1,CURADD		:SET UP TO PUT DEFAULT MSG IN LIST AFTER 033'S
4789	027412	012737	000005	007154	MOV #5,MSGTYP		
4790	027420	013737	002162	007156	MOV MSG5C,CURCC		
4791	027426	105021			1\$: CLRB (R1)+		:FILL EXPT OR TRAN BUFFER WITH 0'S IF A CLEAR
4792	027430	005302			DEC R2		:DO 'BUFLIM' NUMBER OF BYTE LOCATIONS
4793	027432	001375			BNE 1\$		
4794	027434	004737	022676		JSR PC,BLDBUF		: 'CLEAR' REALLY MEANS TO PUT DEFAULT MSG IN
4795	027440	000207			RTS PC		:WHEN DONE, RETURN TO PARSER
4796							
4797							
4798	027442	012705	003250		ACTSHW: MOV #SHTAB,R5		
4799	027446	122571	000000	5\$:	CMPB (R5)+,@(R1)		:LOOK AT FIRST BYTE OF MSG TO DECIPHER TYPE
4800	027452	001404			BEQ 6\$		
4801	027454	020527	003257		CMP R5,#SHTEND		:SEE IF LOOKED AT ALL OF DEFAULTS YET
4802	027460	001372			BNE 5\$		
4803	027462	005205			INC R5		:MUST BE OPR. SPEC'D THEN
4804	027464	162705	003251	6\$:	SUB #SHTAB+1,R5		
4805	027470	006305			ASL R5		
4806	027472	016137	000002	007172	MOV 2(R1),TEMP		
4807	027500				PRINTF #SHMSG,SHTYTB(R5),TEMP		:PRINT MSG SIZE & TYPE
4808	027500	013746	007172			MOV TEMP,-(SP)	
4809	027504	016546	003230			MOV SHTYTB(R5),-(SP)	
4810	027510	012746	013251			MOV #SHMSG,-(SP)	
4811	027514	012746	000003			MOV #3,-(SP)	
4812	027520	010600				MOV SP,R0	
4813	027522	104417				TRAP C\$PNTF	
4814	027524	062706	000010			ADD #10,SP	
4815	027530	062701	000004				
4816	027534	005302			ADD #4,R1		:BUMP R1 TO NEXT SET OF POINTERS
4817	027536	001341			DEC R2		
4818	027540	013737	007220	010334	BNE ACTSHW		
4819	027546	013737	007222	010336	MOV MODTYP,DEV1		
4820	027554	013737	007230	010340	MOV MLTYP,DEV2		
4821	027562	013737	007226	010342	MOV RPASS,DEV3		
4822	027570	004737	023022		MOV PARAM,DEV4		
4823	027574	000207			JSR PC,SHWOP		:SHOW THE OPERATOR THE CURRENT MODE..... ALSO
4824					RTS PC		
4825	027576	013737	003320	007146	ACTDMS: MOV PSNUM,STADD		:SETUP STARTING ADDRESS FOR DUMP
4826	027604	005037	007152		CLR BYTBIT		:SET DEFAULT OF WORD DUMP
4827	027610	012737	000052	003204	MOV #DMPS,KEYWD1		:FLAG THAT A DUMP WAS TYPED
4828	027616	000403			BR ACTDME		

4829								
4830	027620	012737	177777	007152	ACTDMQ: MOV	#-1,BYTBIT		:SET DUMP FLAG TO 'DUMP-WORD'
4831	027626	013737	003320	007150	ACTDME: MOV	PSNUM,ENADD		:SETUP END ADDRESS FOR DUMP (=START IF NO 'EEE'
4832	027634	105037	003324		ACTDMX: CLRB	PSNNUF		:CLEAR NOT-ENOUGH FLAG, 'DUMP N-N/B' IS VALID
4833	027640	000207			RTS	PC		
4834								



```

4835
4836
4837 027642 012737 000010 003204 ACTSTE: MOV #SETEXP,KEYWD1
4838 027650 000403 BR ACTSTX
4839
4840 027652 012737 000011 003204 ACTSTT: MOV #SETTPN,KEYWD1
4841 027660 012737 000001 003210 ACTSTX: MOV #1,QUALVL ;SET UP DEFAULT COPY TO 1 (/COPY=0)
4842 027666 000207 RTS PC
4843
4844 027670 012737 000012 003206 ACTSIZE: MOV #SIZE,QUALFG
4845 027676 000207 RTS PC
4846
4847 027700 012737 000013 003206 ACTCOP: MOV #QCOPY,QUALFG
4848 027706 000207 RTS PC
4849
4850 027710 023727 003206 000012 ACTNUM: CMP QUALFG,#SIZE ;SEE IF A SIZE OR COPY TYPED
4851 027716 001023 BNE 1$ ;BR IF IT WAS A COPY
4852 027720 005737 003320 TST PSNUM ;CHECK TO BE SURE DIDN'T TRY SIZE=0
4853 027724 001014 BNE 3$ ; BR IF NO
4854 027726 PRINTF #CLISEO
4855 027726 012746 012411 MOV #CLISEO,-(SP)
4856 027732 012746 000001 MOV #1,-(SP)
4857 027736 010600 MOV SP,R0
4858 027740 104417 TRAP C$PNTF
4859 027742 062706 000004 ADD #4,SP
4860 027746 112737 177777 003325 MOVB #-1,PSGDBD ;SEE ERROR-IN-CMD FLAG
4861 027754 000411 BR 2$
4862 027756 013737 003320 007156 3$: MOV PSNUM,CURCC ;IF A SIZE LOAD CURCC WITH BYTE COUNT
4863 027764 000405 BR 2$
4864 027766 013737 003320 003210 1$: MOV PSNUM,QUALVL ;IF A COPY, LOAD COPY COUNT
4865 027774 005237 003210 INC QUALVL ;INCREMENT SO FIRST DEC MAKES IT REAL #
4866 030000 000503 2$: BR ACTMEX
4867
4868 030002 012737 000007 007154 ACTOPM: MOV #7,MSGTYP
4869 030010 010437 007172 MOV R4,TEMP ;KEEP TRACK OF START OF QUOTED TEXT
4870 030014 005237 007172 INC TEMP ; SO CAN CALC OPCNT AT END OF QUOTES
4871 030020 000207 RTS PC
4872
4873 030022 010402 ACTEQO: MOV R4,R2
4874 030024 163702 007172 SUB TEMP,R2
4875 030030 010237 007156 MOV R2,CURCC ;CALC BYTE COUNT FOR QUOTED TEXT
4876 030034 010237 002166 MOV R2,OPCNT
4877 030040 013701 007172 MOV TEMP,R1
4878 030044 012705 002524 MOV #OPBUF,R5
4879 030050 112125 1$: MOVB (R1)+,(R5)+ ;COPY QUOTED TEXT TO OPBUF
4880 030052 005302 DEC R2
4881 030054 001375 BNE 1$
4882 030056 000454 BR ACTMEX
4883
4884 030060 ACTBCR: PRINTF #CLIBCR ;BAD CHAR. IN OPR. QUOTED STRING
4885 030060 012746 012344 MOV #CLIBCR,-(SP)
4886 030064 012746 000001 MOV #1,-(SP)
4887 030070 010600 MOV SP,R0
4888 030072 104417 TRAP C$PNTF
4889 030074 062706 000004 ADD #4,SP
4890 030100 000207 RTS PC
  
```



4912	030216	012737	000003	007220	ACTATV: MOV	#ACT,MODTYP	
4913	030224	000432			BR	ACTM2X	
4914							
4915	030226	012737	000002	007220	ACTPAS: MOV	#PAS,MODTYP	
4916	030234	105037	003324		CLRB	PSNNUF	:CLEAR NOT-ENOUGH FLAG
4917	030240	005037	007222		CLR	MLTYP	:CLEAR MAINT LOOP TYPE
4918	030244	000207			RTS	PC	
4919							
4920	030246	005037	007220		ACTREC: CLR	MODTYP	
4921	030252	000417			BR	ACTM2X	
4922							
4923	030254	012737	000006	007220	ACTLIS: MOV	#LIS,MODTYP	
4924	030262	000413			BR	ACTM2X	
4925							
4926	030264	012737	000004	007220	ACTDLL: MOV	#DOW,MODTYP	
4927	030272	000407			BR	ACTM2X	
4928							
4929	030274	012737	000001	007220	ACTTRA: MOV	#TRA,MODTYP	
4930	030302	000403			BR	ACTM2X	
4931							
4932	030304	012737	000005	007220	ACTTAL: MOV	#TAL,MODTYP	
4933							
4934	030312	042737	000004	007226	ACTM2X: BIC	#ECHOB,PARAM	:DISABLE /ECHO (ALL BUT PASSIVE MODE)
4935	030320	105037	003324		CLRB	PSNNUF	:CLEAR NOT-ENOUGH FLAG
4936	030324	005037	007222		CLR	MLTYP	:CLEAR MAINT LOOP TYPE
4937	030330	000207			RTS	PC	
4938							

4939	030332	012737	000036	003206	ACTNO:	MOV	#NO,QUALFG		
4940	030340	000207				RTS	PC		
4941									
4942	030342	022737	000036	003206	ACTECH:	CMP	#NO,QUALFG		
4943	030350	001422				BEQ	1\$		
4944	030352	052737	000004	007226		BIS	#ECHOB,PARAM		
4945	030360	022737	000002	007220		CMP	#PAS,MODTYP		:BE SURE IN PASSIVE MODE IF
4946	030366	001416				BEQ	2\$		:IF TRYING TO SET /ECHO
4947	030370					PRINTF	#CLINPS		
4948	030370	012746	012301						MOV #CLINPS,-(SP)
4949	030374	012746	000001						MOV #1,-(SP)
4950	030400	010600							MOV SP,R0
4951	030402	104417							TRAP C\$PNTF
4952	030404	062706	000004						ADD #4,SP
4953	030410	112737	177777	003325		MOVB	#-1,\$GDBD		
4954	030416	042737	000004	007226	1\$:	BIC	#ECHOB,PARAM		
4955	030424	005037	003206		2\$:	CLR	QUALFG		:CLEAR 'NO' OUT OF QUALIFIER FLAG
4956	030430	000476				BR	ACTLXX		
4957									
4958	030432	012701	000002		ACTCHK:	MOV	#DATCKB,R1		:SET DATA CHECK BIT
4959	030436	000410				BR	ACTQFG		
4960									
4961	030440	012701	000001		ACTSTS:	MOV	#STATB,R1		:SET THE STATUS BIT
4962	030444	000405				BR	ACTQFG		
4963									
4964	030446	012701	000020		ACTCRC:	MOV	#CRCB,R1		:SET THE CRC BIT
4965	030452	000402				BR	ACTQFG		
4966									
4967	030454	012701	000040		ACTPRO:	MOV	#PROTOB,R1		:SET THE PROTOCOL BIT
4968									
4969	030460	050137	007226		ACTQFG:	BIS	R1,PARAM		
4970	030464	022737	000036	003206		CMP	#NO,QUALFG		
4971	030472	001002				BNE	1\$		
4972	030474	040137	007226			BIC	R1,PARAM		
4973	030500	005037	003206		1\$:	CLR	QUALFG		:CLEAR 'NO' OUT OF QUALIFIER FLAG
4974	030504	000450				BR	ACTLXX		
4975									
4976	030506	013737	003320	007230	ACTRPS:	MOV	PSNUM,RPASS		:GET NUMBER OF 'RUN PASSES'
4977	030514	000444				BR	ACTLXX		
4978									
4979	030516	012737	000005	007222	ACTMOP:	MOV	#5,MLTYP		
4980	030524	000417				BR	ACTLPX		
4981	030526	012737	000001	007222	ACTTLP:	MOV	#1,MLTYP		
4982	030534	000413				BR	ACTLPX		
4983	030536	012737	000002	007222	ACTCLP:	MOV	#2,MLTYP		
4984	030544	000407				BR	ACTLPX		
4985	030546	012737	000003	007222	ACTLLP:	MOV	#3,MLTYP		
4986	030554	000403				BR	ACTLPX		
4987	030556	012737	000004	007222	ACTRLP:	MOV	#4,MLTYP		
4988									
4989	030564	022737	000003	007220	ACTLPX:	CMP	#ACT,MODTYP		:BE SURE IN ACTIVE IF TRYING TO SET LOOP
4990	030572	001415				BEQ	ACTLXX		: BR IF IN ACTIVE
4991	030574	112737	177777	003325		MOVB	#-1,\$GDBD		
4992	030602	005037	007222			CLR	MLTYP		:CLEAR ANY LOOP TYPE THAT MAY HAVE GOT SET
4993	030606					PRINTF	#CLIBDL		
4994	030606	012746	012237						MOV #CLIBDL,-(SP)

4995	030612	012746	000001
4996	030616	010600	
4997	030620	104417	
4998	030622	062706	000004
4999	030626	105037	003324
5000	030632	000207	
5001			

ACTLXX: CLR8 PSNUF  
RTS PC

;CLEAR NOT-ENOUGH FLAG

MOV	#1,-(SP)
MOV	SP,R0
TRAP	C\$PNTF
ADD	#4,SP



5038  
5039  
5040  
5041  
5042  
5043  
5044  
5045  
5046  
5047  
5048  
5049  
5050  
5051  
5052  
5053  
5054  
5055  
5056  
5057  
5058  
5059  
5060

.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  
:  
:RXONLY:  
:RXON2: MOV RXPTR,CPTRR  
: MOV RXMTOT,DVRCT ;SET UP MESSAGE COUNT  
: BIS #QRX+#ERX,FLAG ;SET UP RX QUE  
: CLR CPTR ;CLEAR THE TX PCINTER  
: JMP ALLTR ;GO RX.

031014  
031014 013737 007076 007160  
031022 013737 007134 007132  
031030 052737 000104 007232  
031036 005037 007162  
031042 000137 031200

5061  
5062  
5063  
5064  
5065  
5066  
5067  
5068  
5069  
5070  
5071  
5072  
5073  
5074  
5075  
5076  
5077 031046 042737 000002 007226  
5078 031054 013737 007100 007162  
5079 031062 013737 007120 007116  
5080 031070 052737 000210 007232  
5081 031076 005037 007160  
5082 031102 000137 031200

.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  
:  
: TXONLY: BIC #DATCKB,PARAM ;SET NOCHECK  
: TXON2: MOV TXPTR,CPTR  
: MOV TXMTOT,DVTCT ;COPY COUNTER FOR THIS PASS  
: BIS #QTX+#ETX,FLAG ;SET THE QUE TX FLAG  
: CLR CPTRR ;CLEAR RX POINTER  
: JMP ALLTR ;GO TX.



5083  
5084  
5085  
5086  
5087  
5088  
5089  
5090  
5091  
5092  
5093  
5094  
5095  
5096  
5097  
5098  
5099  
5100  
5101  
5102  
5103  
5104  
5105  
5106  
5107

.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

PLCK:

PLCK2: MOV TXMTOT,DVTCT ;SET UP THE TRANSMIT COUNT  
MOV TXPTR,CPTR ;SET UP CPTR TO TRANSMIT POINTER  
PLCK3: MOV RXPTR,CPTR ;SET UP CPTR TO REC POINTER  
BIS #QRX+#ERX,FLAG ;SET UP Q AND EXPECT RX  
JMP ALLTR ;AND GO RX FIRST MSG.

031106  
031106 013737 007120 007116  
031114 013737 007100 007162  
031122 013737 007076 007160  
031130 052737 000104 007232  
031136 000137 031200

5108  
5109  
5110  
5111  
5112  
5113  
5114  
5115  
5116  
5117  
5118  
5119  
5120  
5121  
5122  
5123  
5124  
5125  
5126  
5127  
5128  
5129  
5130  
5131  
5132  
5133  
5134  
5135

.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

--  
ALCK: MOV TXMTOT,DVICT  
MOV TXPTR,CPTR ;SET UP TX COUNTS  
MOV RXMTOT,DVRCT ;SET UP COUNTS  
MOV RXPTR,CPTRR  
BIS #QRX+#QTX+#ETX+#ERX,FLAG

5136  
 5137  
 5138  
 5139  
 5140  
 5141  
 5142  
 5143  
 5144  
 5145  
 5146  
 5147  
 5148  
 5149  
 5150  
 5151  
 5152  
 5153  
 5154  
 5155  
 5156  
 5157  
 5158  
 5159  
 5160  
 5161  
 5162  
 5163  
 5164  
 5165  
 5166  
 5167  
 5168  
 5169  
 5170  
 5171  
 5172  
 5173  
 5174  
 5175  
 5176  
 5177  
 5178  
 5179  
 5180  
 5181  
 5182  
 5183  
 5184  
 5185  
 5186  
 5187  
 5188  
 5189  
 5190  
 5191

.SBTTL

TRANSMIT - RECEIVE FOR ALL STANDARD MODES

..

FUNCTIONAL DESCRIPTION:

THIS CODE PERFORMS THE FOLLOWING FUNCTIONS

- 1.) IF RX BUFFERS ARE TO BE QUED, TELL DEVICE CODE TO QUE THEM, LOG RECEIVE QUED.
- 2.) IF TX BUFFERS ARE TO BE QUED, TELL DEVICE CODE TO QUE THEM, LOG TRANSMIT QUED.
- 3.) WAIT FOR EITHER RECIVE BUFFER OR TRANSMIT BUFFER OR BOTH TO COMPLETE
- 4.) IF RECEIVE COMPLETE LOG IT UPDATE RX TABLE IF DATA CHECKING.
- 5.) IF TRANSMIT COMPLETE LOG IT.
- 6.) WHEN BOTH TRANSMIT AND RECIEVE LISTS ARE DONE GO TO THE COMPARE BUFFER CODE

SUBORDINATE ROUTINES USED:

- 'DVRXQ' -QUE RECEIVE BUFFER SPACE TO DEVICE
- 'LOGRXQ' -LOG RECEIVE BUFFER SPACE TO EVENT LOG
- 'LOGTXQ' -LOG TRANSMIT BUFFER QUED TO EVENT LOG
- 'DVTXRX' -QUE TRANSMIT BUFFER AND WAIT FOR RX OR TX TO COMPLETE
- 'LOGRXC' -LOG RECEIVE BUFFER COMPLETED TO EVENT LOG
- 'LOGTXC' -LOG TRANSMIT BUFFER COMPLETED TO EVENT LOG

USE OF FLAG BITS:

- QRX - SET ON INPUT TO ALLTR IF REC IS TO BE QUED TO DEVICE. CLEARED BY DVRXQ AND THEN SET BY DVTXRX WHEN RX BUFFER IS COMPLETED.
- QTX - SET ON INPUT TO ALLTR IF TRANSMIT IS TO BE QUED TO DEVICE. CLEARED ON ENTRY TO DVTXRX AND SET BY DVTXRX WHEN TX BUFFER IS COMPLETED.
- ETX - USED BY DVTXRX TO DETERMINE IF TX BUFFER COMPLETED IS EXPECTED.
- ERX - USED BY DVTXRX TO DETERMINE IF RX BUFFER COMPLETED IS EXPECTED.

CALLING SEQUENCE:

JMP ALLTR ;GO TO TRANSMIT-RECEIVE FOR ALL STANDARD MODES

ALLTR:

031200  
 031200 032737 000004 007232  
 031206 001420  
 031210 013702 007160  
 031214 011237 007176  
 031220 012237 007126  
 031224 011237 007200  
 031230 011237 007130  
 031234 010237 007160  
 031240 004737 034120  
 031244 004737 020746  
 031250 032737 000010 007232

ALCK5:

BIT #QRX,FLAG  
 BEQ ALCK1 ;IF NOT RX GO TO TX'S

MOV CPTRR,R2  
 MOV (R2),TEMP2  
 MOV (R2)+,DVRXA  
 MOV (R2),TEMP3  
 MOV (R2),DVRCC  
 MOV R2,CPTRR  
 JSR PC,DVRXQ ;GO QUE DEVICE  
 JSR PC,LOGRXQ ;LOG REC QUED  
 BIT #QTX,FLAG

ALCK1:

5192	031256	001416			BEQ	ALCK2			;IF NO TX'S GO TO 2
5193	031260	013702	007162		MOV	CPTR,R2			
5194	031264	011237	007176		MOV	(R2),TEMP2			
5195	031270	012237	007112		MOV	(R2)+,DVTXA			
5196	031274	011237	007200		MOV	(R2),TEMP3			
5197	031300	012237	007114		MOV	(R2)+,DVTCC			
5198	031304	010237	007162		MOV	R2,CPTR			
5199	031310	004737	020712		JSR	PC,LOGTXQ			
5200									
5201	031314	004737	034200		ALCK2:	JSR	PC,DVTXRX		;GO TO TX AND RX SUB ROUT.
5202									
5203	031320	032737	000004	007232	BIT	#QRX,FLAG			;CHECK FOR REC. MSG.
5204	031326	001514			BEQ	ALCK3			
5205	031330	013737	007126	007176	MOV	DVRXA,TEMP2			
5206	031336	013737	007130	007200	MOV	DVRCC,TEMP3			
5207	031344	004737	020764		JSR	PC,LOGRXC			;LOG REC COMPLETE
5208	031350	032737	000004	007226	UPTABL:	BIT	#ECHOB,PARAM		;IS THIS ECHO MODE(PASSIVE)
5209	031356	001406			BEQ	UPTA4			;IF NOT GO TO 4
5210	031360	013702	007162		MOV	CPTR,R2			;ELSE SET R2 TO PRESENT TX TABL
5211	031364	013722	007176		MOV	TEMP2,(R2)+			;STORE OFF RX ADD
5212	031370	013712	007200		MOV	TEMP3,(R2)			;AND CC
5213	031374	032737	000002	007226	UPTA4:	BIT	#DATCKB,PARAM		;IS DATA CHECKING ASKED FOR
5214	031402	001015			BNE	UPTA1			;IF SO GO TO 1
5215	031404	012737	000001	007132	MOV	#01,DVRCT			;ELSE SET DVRCT TO A 1
5216	031412	013737	007076	007160	MOV	RXPTR,CPTRR			;RESET POINTER
5217	031420	022737	000003	007220	CMP	#ACT,MODTYP			;IS THIS ACTIVE
5218	031426	001002			BNE	UPTA3			
5219	031430	005237	007132		INC	DVRCT			;IF YES BUMP COUNT
5220	031434	000424			UPTA3:	BR	UPTEX		
5221	031436	013702	007160		UPTA1:	MOV	CPTRR,R2		
5222	031442	011237	007172		MOV	(R2),TEMP			;LOAD TEMP WITH PREV. COUNT
5223	031446	163737	007200	007172	SUB	TEMP3,TEMP			;LOAD TEMP WITH PREV.COUNT-CURRENT
5224	031454	013722	007200		MOV	TEMP3,(R2)+			
5225	031460	063737	007200	007176	ADD	TEMP3,TEMP2			
5226	031466	013722	007176		MOV	TEMP2,(R2)+			;STORE OF NEW ADD
5227	031472	013712	007172		MOV	TEMP,(R2)			;AND NEW CC
5228	031476	162702	000002		SUB	#2,R2			;PUT POINTER BACK TO ADDR.
5229	031502	010237	007160		MOV	R2,CPTRR			;AND RESTORE IT.
5230	031506				UPTEX:				
5231	031506	022737	000002	007220	CMP	#PAS,MODTYP			
5232	031514	001007			BNE	ALCK2A			;IF NOT PASSIVE LOOP THEN GO TO 2A
5233	031516	042737	000104	007232	BIC	#QRX+#ERX,FLAG			;CLEAR BOTH EXPECTED AND COMPLETED FLAGS
5234	031524	052737	000210	007232	BIS	#QTX+#ETX,FLAG			;SET THE TX FLAGS
5235	031532	000646			BR	ALCK1			
5236									
5237	031534	005337	007132		ALCK2A:	DEC	DVRCT		;DEC REC COUNT
5238	031540	005737	007132		TST	DVRCT			;IS IT ALL DONE
5239	031544	001005			BNE	ALCK3			;NO. GO CHECK TX
5240	031546	042737	000004	007232	BIC	#QRX,FLAG			;CLEAR THE RX FLAG
5241	031554	005037	007160		CLR	CPTRR			;YES. CLEAR POINTER
5242	031560	032737	000010	007232	ALCK3:	BIT	#QTX,FLAG		;IS IT TX
5243	031566	001447			BEQ	ALCK4			;IF NOT TX THEN GO BACK
5244	031570	013737	007112	007176	MOV	DVTXA,TEMP2			
5245	031576	013737	007114	007200	MOV	DVTCC,TEMP3			;LOG TX COMPLETED
5246	031604	004737	020730		JSR	PC,LOGTXC			
5247	031610	005337	007116		DEC	DVTCT			;DEC TX COUNT

5248	031614	022737	000002	007220		CMP	#PAS,MODTYP	
5249	031622	001013				BNE	ALCK3A	;IF NOT PASSIVE MODE GO TO 3A
5250	031624	042737	000210	007232		BIC	#QTX+ETX,FLAG	;CLEAR THE TX FLAGS
5251	031632	052737	000104	007232		BIS	#QRX+ERX,FLAG	;AND SET THE RX FLAGS
5252	031640	005737	007116			TST	DVTCT	
5253	031644	001005				BNE	ALCK3C	;IF MORE RX'S DO IT
5254	031646	000137	031726			JMP	CMPSR	; ELSE COMPARE
5255	031652	005737	007116		ALCK3A:	TST	DVTCT	;IS IT ALL DONE
5256	031656	001402				BEQ	ALCK3B	;IF NOT GO BACK TO 5
5257	031660	000137	031200		ALCK3C:	JMP	ALCK5	
5258	031664	005037	007162		ALCK3B:	CLR	CPTR	;IF SO CLEAR POINTER
5259	031670	042737	000010	007232		BIC	#QTX,FLAG	;CLEAR TX FLAG
5260	031676	032737	000002	007226		BIT	#DATCKB,PARAM	;IS IT DAT CK
5261	031704	001403				BEQ	ALCK4A	;IF NOT THEN END WO CKING RX.
5262	031706	005737	007160		ALCK4:	TST	CPTRR	
5263								
5264	031712	001362				BNE	ALCK3C	;IF SOME RX'S LEFT GO BACK
5265	031714	005737	007162		ALCK4A:	TST	CPTR	
5266	031720	001402				BEQ	ALCK4B	;BRANCH IF ANY TX'S LEFT
5267	031722	000137	031314			JMP	ALCK2	
5268	031726				ALCK4B:			
5269								
5270								
5271								

5272  
5273  
5274  
5275  
5276  
5277  
5278  
5279  
5280  
5281  
5282  
5283  
5284  
5285  
5286  
5287  
5288  
5289  
5290  
5291  
5292  
5293  
5294  
5295  
5296  
5297  
5298  
5299  
5300  
5301  
5302  
5303  
5304  
5305  
5306  
5307  
5308  
5309  
5310  
5311  
5312  
5313  
5314  
5315  
5316  
5317  
5318  
5319  
5320  
5321  
5322  
5323  
5324  
5325  
5326  
5327

.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

---

CMPSR: BIT #DATCKB,PARAM ;IS DATA CHECKING TO BE DONE  
BEQ CMPSEX ;IF NOT THEN EXIT  
MOV RXPTR,CPTR ;PUT START OF RX POINTERS TO CPTR  
MOV CMPPTR,CPTRR ; AND START OF COMPARE POINTS TO CPTRR  
MOV RXMTOT,DVRCCT  
  
CMPS3:  
MOV CPTR,R2 ;MOVE CURRET RX PT.TO R2  
MOV (R2),TEMP2 ;MOVE RX ADD TO EVENT LOG  
MOV (R2)+,R1 ;SET R1 TO START ADD OF RX  
MOV (R2)+,TEMP3 ;SET CHAR COUNT TO EVENT LOG  
MOV R2,CPTR ;RESTORE RX POINT  
  
MOV CPTRR,R2 ;PUT R2 AT COMPARE TABLE  
MOV (R2)+,R3 ;SET R3 TO COMPARE ADD  
MOV (R2)+,R4 ;SET R4 TO COMP CC  
MOV R2,CPTRR ;RESTORE POINTER  
MOV R4,TEMP4  
JSR PC,LOGCMP ;LOG COMPARE START.  
  
CMP R4,TEMP3 ;IS COMPARE COUNT = TO RX COUNT  
BEQ CMPS7 ;IF SO GO TO 7  
INC ERRCNT  
ERRSOFT 1,EDDLE,ERR10 ;PRINT ERROR



5368  
5369  
5370  
5371  
5372  
5373  
5374  
5375  
5376  
5377  
5378  
5379  
5380  
5381  
5382  
5383  
5384  
5385  
5386  
5387  
5388  
5389  
5390  
5391  
5392  
5393  
5394  
5395  
5396  
5397  
5398

.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 PROM1  
IN NOT = TO 0 THEN GO BACK TO MODE DISPATCHER

SUBORDINATE ROUTINES USED:

-----  
'LOGEOP' - LOG END OF PASS TO EVENT LOG

```
CMPSEX: INC      PSCNT          ;BUMP PASS COUNT
MOV      NOBUF,TEMP4
MOV      PSCNT,TEMP2
MOV      ERRCNT,TEMP3
JSR      PC,LOGEOP          ;LOG END OF PASS

CMP      #-1,RPASS         ;SEE IF RPASS=-1
BEQ      1$                ;IF IT IS DON'T DECRMNT, LOOP FOREVER
DEC      RPASS             ;DEC PASS COUNT
BEQ      2$                ;IF DONE EXIT TEST
1$:      JMP      GTRX2     ;ELSE GO BACK AND DISPATCH
2$:      JMP      GTRAS     ;WHEN RPASS=0 GO BACK TO 'DCLT>'
```



5399  
 5400  
 5401  
 5402  
 5403  
 5404  
 5405  
 5406  
 5407  
 5408  
 5409  
 5410  
 5411  
 5412  
 5413  
 5414  
 5415  
 5416  
 5417  
 5418  
 5419  
 5420  
 5421  
 422  
 5423  
 5424  
 5425  
 5426  
 5427  
 5428  
 5429  
 5430  
 5431  
 5432  
 5433  
 5434  
 5435  
 5436  
 5437  
 5438  
 5439  
 5440  
 5441  
 5442  
 5443  
 5444  
 5445  
 5446  
 5447  
 5448  
 5449  
 5450  
 5451  
 5452  
 5453  
 5454

.SBTTL DOWN-LINE-LOAD SECTION

..\*\*  
 : FUNCTIONAL DESCRIPTION:  
 : DOWN-LINE-LOAD SECTION  
 : IN THIS MODE OF TESTING THE 'HOST' OR ORIGINATING STATION  
 : REQUESTS THE 'SATELLITE' OR BOOT STATION TO ENTER MOP MODE.  
 : THE BOOT STATION THEN SENDS A 'REQUEST PROGRAM MESSAGE'.  
 : THE 'HOST' THEN SENDS A 'MEMORY LOAD WITH TRANSFER ADDRESS'  
 : THAT CONTAINS IMAGE DATA TO BE LOADED BY THE BOOT STATION'S  
 : M9312 STARTING AT LOC. 0. THIS IMAGE DATA WILL CONTAIN A  
 : PROGRAM THAT WILL PRINT A MSG THAT DOWN-LINE-LOAD WAS SUCESSFUL.

: SUBORDINATE ROUTINES USED:  
 :  
 : 'DLTXRX' - SPECIAL TX RX ROUTINE FOR DLL  
 : 'DVRXQ' - QUE RX BUFFER SPACE TO DEVICE  
 : 'LOGRXQ' - LOG RX SPACE QUED TO EVENT LOG  
 : 'LOGTXQ' - LOG TX BUFFER QUED TO EVENT LOG  
 : 'DVTXRX' - QUE TX BUFFER AND WAIT FOR RX OR TX TO COMPLETE  
 : 'LOGTXC' - LOG TX COMPLETED TO EVENT LOG  
 : 'LOGRXC' - LOG RX COMPLETED TO EVENT LOG

: CALLING SEQUENCE:  
 : JMP @MODE(R2) ;DISPATCH TO MODE BASED ON MODE TYPE IN R2

DLL: BIS #ERX,FLAG ;SET EXPECTED TO RX  
 BIC #DATCKB,PARAM ;CLEAR NOCHECK  
 MOV #DLLM1,CURADD ;SET THE DOWN LINE LOAD MSG TO #1  
 MOV DLLM1C,CURCC ;SET THE CC  
 JSR PC,DLTXRX ;GO TO THE DOWN LINE TX RX ROUTINE

;RETURN WHEN TX AND RX ARE COMPLETED

MOV #DLLM2,CURADD ;SET THE DOWN LINE LOAD MSG TO #2  
 MOV DLLM2C,CURCC ;SET CC  
 BIC #DLLGA,FLAG ;CLEAR THE GO AHEAD FLAG  
 JSR PC,DLTXRX ;GO TO THE DOWN LINE TX RX ROUTINE

; RETURN WHEN TX AND RX ARE COMPLETED

DLLPRI: PRINTF #DLLCM  
 MOV #DLLCM,-(SP)  
 MOV #1,-(SP)  
 MOV SP,R0  
 TRAP C\$PNTF  
 ADD #4,SP  
 JMP GTRAS

DLLEA: ERRHRD 20,DLLAB,ERR14  
 TRAP C\$ERHRD  
 .WORD 20  
 .WORD DLLAB  
 .WORD ERR14

```

5455
5456 032400 000137 025640      JMP      GTRAS          ;PRINT ABORT AND EXIT
5457
5458
5459
5460 032404                      DLTXRX:
5461 032404 052737 000004 007232  BIS      #QRX,FLAG      ;SET THE QUE RX FLAG
5462 032412 012737 004326 007126  MOV      #RXBUF,DVRYA   ;SET THE DEVICE RX BUFFER TO RXBUF
5463 032420 012737 004326 007176  MOV      #RXBUF,TEMP2   ;SET UP FOR LOG
5464 032426 012737 000400 007130  MOV      #256.,DVRCC    ;SET UP FOR CC OF 256
5465 032434 012737 000400 007200  MOV      #256.,TEMP3    ;SET UP FOR LOG
5466 032442 004737 034120          JSR      PC,DVRYQ      ; GO QUE RX
5467 032446 004737 020746          JSR      PC,LOGRXQ    ;AND LOG IT...
5468
5469 032452 013737 007164 007112  MOV      CURADD,DVTXA   ;SET UP FOR TX
5470 032460 013737 007164 007176  MOV      CURADD,TEMP2   ;AND LOG
5471 032466 013737 007156 007114  MOV      CURCC,DVTCC    ;SE UP FOR TX COUNT
5472 032474 013737 007156 007200  MOV      CURCC,TEMP3    ;AND LOG IT
5473 032502 004737 020712          JSR      PC,LOGTXQ    ;LOG THE TX QUEUED
5474 032506 052737 000210 007232  BIS      #QTX+#ETX,FLAG ;SET UP TO QUE AND EXPECTED
5475 032514 004737 034200          JSR      PC,DVTXRX    ;GO TO DEVICE ROUTINE
5476 032520 032737 000400 007232  BIT      #DLLGA,FLAG    ;TEST FOR GO AHEAD BIT
5477 032526 001047          BNE     DLLE1          ;IF SET GO TO ONE
5478 032530 032737 000010 007232  BIT      #QTX,FLAG      ;ELSE CHECK FOR TX DONE
5479 032536 001020          BNE     DLLE6          ;IF DONE THEN BRANCH
5480
5481 032540 012737 020062 007204  MOV      #TXNC,CONOTM   ;
5482 032546 013737 004326 007200  DLLE7:  MOV      RXBUF,TEMP3   ;
5483 032554 013737 003326 007202  MOV      TXBUF,TEMP4    ;
5484 032562 012737 017130 007176  MOV      #DLLAB,TEMP2   ;
5485 032570 004737 020774          JSR      PC,LGDVE     ;LOG ERROR
5486 032574 000137 032370          JMP      DLLEA        ;ABORT TEST
5487
5488 032600 013737 007112 007176  DLLE6:  MOV      DVTXA,TEMP2   ;
5489 032606 013737 007114 007200  MOV      DVTCC,TEMP3    ;
5490 032614 004737 020730          JSR      PC,LOGTXC   ;LOG TX DONE
5491 032620 042737 000210 007232  BIC      #QTX+#ETX,FLAG ;CLEAR QUE AND EXPECTED
5492 032626 052737 000400 007232  BIS      #DLLGA,FLAG    ;SET THE GO AHEAD BIT
5493 032634 023737 002174 007114  CMP      DLLM2C,DVTCC   ;
5494 032642 001441          BEQ     DLLE5          ;EXIT IF SECOND MSG.
5495 032644 000723          BR     DLLE2          ;AND GO BACK TO 2
5496 032646 032737 000004 007232  DLLE1:  BIT      #QRX,FLAG    ;IS THE A RX COMPLETED
5497 032654 001004          BNE     DLLE8          ;IF SO GO TO 8
5498 032656 012737 020102 007204  MOV      #RXNC,CONOTM   ;ELSE SET UP ERROR AND ABORT.
5499 032664 000730          BR     DLLE7
5500 032666 013737 007126 007176  DLLE8:  MOV      DVRYA,TEMP2   ;
5501 032674 013737 007130 007200  MOV      DVRCC,TEMP3    ;
5502 032702 004737 020764          JSR      PC,LOGRXC   ;LOG RECEIVE COMPLETE
5503 032706 022737 006010 004326  CMP      #6010,RXBUF    ;CHECK FOR FIRST WORD OF RX
5504
5505 032714 001404          BEQ     DLLE3          ;SEC BOOT MSG.
5506 032716 012737 020122 007204  DLLE4:  MOV      #RXM1,CONOTM  ;SET UP MMSG AND ABORT
5507 032724 000710          BR     DLLE7          ;ABORT TEST
5508
5509 032726 022737 000001 004330  DLLE3:  CMP      #1,RXBUF+2    ;IS SECOND WORD 1
5510 032734 001404          BEQ     DLLE5          ;IF OK RETURN

```

CZCLKAO DMR,DMC-11 DATA COMM. LINK TEST  
CZCLKA.P11 18-APR-80 09:24

MACY11 30A(1052) <sup>N 10</sup> 18-APR-80 09:24 PAGE 131  
DOWN-LINE-LOAD SECTION

SEQ 0130

5511 032736 012737 020145 007204  
5512 032744 000700  
5513  
5514 032746 000207  
5515  
5516  
5517

MOV #RXM2,CONOTM  
BR DLLE7 ;SET UP MESSAGE AND ABORT  
DLLES: RTS PC ;RETURN TO CALLER

5518  
 5519  
 5520  
 5521  
 5522  
 5523  
 5524  
 5525  
 5526  
 5527  
 5528  
 5529  
 5530  
 5531  
 5532  
 5533  
 5534  
 5535  
 5536  
 5537 032750  
 5538 032750 042737 000002 007226  
 5539 032756 012702 002524  
 5540 032762 012722 177777  
 5541 032766 022702 002646  
 5542 032772 001373  
 5543 032774  
 5544 032774 104443  
 5545 032776 000406  
 5546 033000 002524  
 5547 033002 000142  
 5548 033004 013575  
 5549 033006 000000  
 5550 033010 000001  
 5551 033012 000110  
 5552 033014  
 5553 033014 005002  
 5554 033016 122762 000377 002524 2\$:  
 5555 033024 001402  
 5556 033026 005202  
 5557 033030 000772  
 5558 033032 010237 002166 3\$:  
 5559  
 5560 033036 012737 002524 007112  
 5561 033044 012737 002524 007176  
 5562 033052 013737 002166 007200  
 5563 033060 013737 002166 007114  
 5564 033066 004737 020712  
 5565 033072 052737 000210 007232  
 5566 033100 005037 007160  
 5567  
 5568 033104 004737 034200  
 5569  
 5570 033110 013737 007112 007176  
 5571 033116 013737 007114 007200  
 5572 033124 004737 020730  
 5573 033130 022737 054105 002524

.SBTTL TALK MODE SECTION

++  
 FUNCTIONAL DESCRIPTION:  
 TALK MODE SECTION  
 IN THIS MODE, THE 'TALK' END OF THE LINK TRANSMITS OPERATOR  
 SPECIFIED MESSAGES UNTIL A 'EXIT' MESSAGE IS TYPE. AT THAT POINT,  
 THIS END OF THE LINK GOES INTO 'LISTEN' MODE.

SUBORDINATE ROUTINES USED:

'LOGTXQ' - LOG TX BUFFER QUED TO EVENT LOG  
 'DVTXRX' - QUE TX BUFFER TO DEVICE AND WAIT FOR COMPLETE  
 'LOGTXC' - LOG TX COMPLETE TO EVENT LOG

CALLING SEQUENCE:

JMP @MODE(R2) ;DISPATCH TO MODE BASED ON MODE TYPE IN R2

TALCK:

BIC :DATCKB,PARAM ;SET NOCHECK  
 MOV #OPBUF,R2  
 1\$: MOV #-1,(R2)+ ;CLEAR OUT OPBUFFER FIRST  
 CMP #OPEND,R2  
 BNE 1\$  
 GMANID OPRMM,OPBUF,A,0,1,72.,NO ;GET TALK MESSAGE

TRAP  
 BR 10001\$  
 .WORD OPBUF  
 .WORD TSCODE  
 .WORD OPRMM  
 .WORD 0  
 .WORD TSLOLIM  
 .WORD TSHILIM

10001\$:

CLR R2 ;NOW GET CHAR COUNT  
 CMPB #377,OPBUF(R2)  
 BEQ 3\$  
 INC R2  
 BR 2\$  
 2\$: MOV R2,OPCNT  
 3\$: MOV #OPBUF,DVTXA ;SET UP TX ADDR.  
 MOV #OPBUF,TEMP2  
 MOV OPCNT,TEMP3  
 MOV OPCNT,DVTCC ;SET UP TX CC  
 JSR PC,LOGTXQ  
 BIS #QTX+#ETX,FLAG ;SET UP FLAGS  
 CLR CPTRR ;CLEAR RX POINTER  
 JSR PC,DVTXRX  
 MOV DVTXA,TEMP2  
 MOV DVTCC,TEMP3  
 JSR PC,LOGTXC  
 CMP #'EX,OPBUF ;CHECK FOR EXIT

5574	033136	001304		
5575	033140	022737	052111	002526
5576	033146	001300		
5577	033150	042737	000210	007232
5578	033156	012737	000006	007220
5579	033164	000137	030746	

BNE	TALCK	
CMP	#'IT,OPBUF+2	
BNE	TALCK	
BIC	#QTX+#ETX,FLAG	:CLEAR THE TX BITS
MOV	#LIS,MODTYP	:CHANGE TO LISTEN MODE
JMP	STRX2	:AND GO BACK TO DISPATCH

5580  
 5581  
 5582  
 5583  
 5584  
 5585  
 5586  
 5587  
 5588  
 5589  
 5590  
 5591  
 5592  
 5593  
 5594  
 5595  
 5596  
 5597  
 5598  
 5599  
 5600  
 5601  
 5602  
 5603  
 5604  
 5605  
 5606  
 5607  
 5608  
 5609  
 5610  
 5611  
 5612  
 5613  
 5614  
 5615  
 5616  
 5617  
 5618  
 5619  
 5620  
 5621  
 5622  
 5623  
 5624  
 5625  
 5626  
 5627  
 5628  
 5629  
 5630  
 5631  
 5632  
 5633  
 5634  
 5635

.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

```

LISCK: BIC #DATCKB,PARAM ;CLEAR CHECK BIT
        PRINTF #LISP ;PRINT PROMPT FOR OPR.

LISCKA: MOV #OPBUF,DVRXA ;SET DEVICE UP TO REC AT OPBUF
        MOV #OPBUF,TEMP2
        MOV #82.,DVRCC ;SET UP CHAR COUNT TO 82.
        MOV #82.,TEMP3
        BIS #URX+MERX,FLAG ;SET UP FLAG
        CLR CPTR ;CLEAR THE TX.

        JSR PC,DVRXQ ;QUE RX
        JSR PC,LOGRXQ

        JSR PC,DVTXRX ;GO TO DEVICE RX. SUBROUTINE

        MOV DVRXA,TEMP2
        MOV DVRCC,TEMP3 ;SET UP ADDR.AND CC.
        JSR PC,LOGRXC ;LOG COMPLETED
        ADD DVRXA,DVRCC
        CLRB @DVRCC
        PRINTF #OPBFPT

        CMP #'EX,OPBUF ;COMPARE FOR EX OF 'EXIT'
        BNE LISCKA ;IF NOT EXIT THEN GO BACK
        CMP #'IT,OPBUF+2 ;IF FIRST HALF OK CHECK NEXT PART
        BNE LISCKA ;IF NOT EXIT THE GO BACK
        MOV #TAL,MODTYP ;CHANGE MODE TO TALK
        JMP GTRX2 ;RETURN TO DISPATCHER
  
```

```

MOV #LISP,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #4,SP

MOV #OPBFPT,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #4,SP
  
```

CZCLYAO DMR,DMC-11 DATA COMM. LINK TEST  
CZCLKA.P11 18-APR-80 09:24

MACY11 30A(1052) <sup>E 11</sup> 18-APR-80 09:24 PAGE 135  
LISTEN MODE SECTION

SEQ 0134

5636  
5637

```

5638 .SBTTL          DEVICE FUNCTION SUBROUTINES
5639
5640
5641
5642
5643 .SBTTL          DEVICE INIT SUBROUTINE
5644
5645
5646
5647 :++
5648 : FUNCTIONAL DESCRIPTION:
5649 :   DVINIT- DEVICE INIT ROUTINE
5650 :   THIS ROUTINE IS DEVICE DEPENDENT CODE THAT INITIS
5651 :   THE DEVICE BEING TESTED. (I.E. FULL/HALF DUPLEX BAUD RATE, MAINT MODE.)
5652 :
5653 : INPUTS:          'FHDPLX' INDICATES IF MODE IS FULL OR HALF DUPLEX. (1=FULL)
5654 :                  ADDRESS POINTERS (SELO,...) ALREADY POINT TO DEVICE'S REG.S
5655 :
5656 : SUBORDINATE ROUTINES USED:
5657 :
5658 :   'LGDVE' - LOG DEVICE ERROR TO EVENT LOG
5659 :   'TOORIO' - TIME OUT OR INPUT INTERRUPT OR OUTPUT INTERRUPT
5660 :   'CLRWA' - CLEAR RQI AND WAIT FOR RDI TO GO AWAY
5661 :
5662 : CALLING SEQUENCE:
5663 :   JSR          PC,DVINIT
5664 :--
5665
5666 033400 DVINIT:
5667          ;MASTER CLEAR DEVICE
5668
5669 033400 012737 000100 007272      MOV      #100,TIMER1      ;SET UP TIMER 1 FOR 100(OCTAL) TICKS
5670 033406 005077 156400           CLR      @SEL6
5671 033412 005077 156370           CLR      @SEL4
5672 033416 012777 040000 156352      MOV      #MCLR,@SELO      ;DO A MASTER CLEAR
5673
5674 033424 022737 000004 012024      CMP      #DMRC6,OPTYP     ;IS THIS A 8206
5675 033432 001003           BNE     DVIN6            ;IF NOT GO TO 6
5676 033434 112777 000200 156336      MOVB    #200,@BSEL1      ;SET RUN FOR 8206
5677 033442 022737 000006 012024 DVIN6:  CMP      #DMR6,OPTYP     ;IS THIS AN 8206 DMR
5678 033450 001003           BNE     DVIN2            ;IF NOT GO TO 2
5679 033452 112777 000200 156320      MOVB    #200,@BSEL1      ;SET RUN BIT FOR 8206
5680
5681 033460 005777 156312           DVIN2: TST      @SELO      ;IS RUN BIT SET
5682 033464 100426           BMI     DVIN1            ;IF YES GO TO 1 ELSE...
5683 033466           BREAK
5684 033466 104422
5685 033470 005737 007272           TST     TIMER1          ;SEE IF TIME HAS EXPIRED
5686 033474 001371           BNE     DVIN2            ;IF NOT GO BACK AND CHECK
5687                                     ;AGAIN ELSE...PRINT ERROR
5688 033476 012737 016340 007176      MOV     #DVEM3,TEMP2
5689 033504 017737 156266 007200      MOV     @SELO,TEMP3
5690 033512 017737 156264 007202      MOV     @SEL2,TEMP4
5691 033520 004737 020774           JSR     PC,LGDVE
5692 033524 005237 007144           INC     ERRCNT
5693 033530           ERRSOFT 11,DVEM3,ERR13
  
```

TRAP C\$BRK





```
5750 034006 052777 002400 155776      BIS      #MAINTB+HALFDB,@SEL6      ;IF SO SET MAINT MODE BIT
5751 034014 000406                                BR      DVIN4                    ; AND FORCE HALF DUPLEX
5752
5753 034016 005737 007224              DVIN5:  TST      FHDPLX          ;IS THIS A HALF/DUP
5754 034022 001003                                BNE     DVIN4                    ;IF NOT GO TO 4
5755 034024 052777 002000 155760      BIS      #HALFDB,@SEL6          ;ELSE SET HALF/DUP
5756
5757 034032 017737 155754 007206      DVIN4:  MOV      @SEL6,CONTIN    ;SET UP CONTROL IN FOR MODS
5758 034040 004737 034720                                JSR     PC,CLRAW                 ;GO CLEAR RQI AND WAIT
5759                                ;FOR RDI TO GO AWAY.
5760 034044 023727 012024 000006      CMP      OPTYP,#6              ;IS THIS DMR
5761 034052 002403                                BLT     DVINEX                  ;IF NOT EXIT
5762 034054 052737 001000 007232      BIS      #DMRRUN,FLAG          ;SET RUN OUTPUT EXPECTED BIT
5763
5764 034062 000207              DVINEX:  RTS      PC              ;RETURN TO CALLER
5765
5766
5767
5768
5769
```

5770 .SBTTL DEVICE GET MODEM STATUS SUBROUTINE

5771  
5772  
5773  
5774  
5775  
5776  
5777  
5778  
5779  
5780  
5781  
5782  
5783  
5784  
5785  
5786  
5787  
5788  
5789  
5790  
5791  
5792  
5793  
5794  
5795  
5796  
5797  
5798  
5799  
5800  
5801  
5802

..\*\*  
: FUNCTIONAL DESCRIPTION:  
: 'DVMODS' GET MODEM STATUS  
:  
: IMPLICIT INPUTS:  
: THE BIT POSITION AND AVAILABILITY OF THE MODEM SIGNALS CTS,DSR,...RI,..  
: IN THE DEPENDENT POSITION OF THE GLOBAL EQUATES SECTION.  
:  
: OUTPUTS:  
: CURRENT MODEM SIGNAL VALUES IN 'MODS'  
:  
: SUBORDINATE ROUTINES USED:  
:  
: 'TOORIO' - TIME OUT OR INPUT INTERRUPT OR OUTPUT INTERRUPT  
: 'CLR' - CLEAR RQI AND WAIT FOR RDI TO CLEAR  
:  
: CALLING SEQUENCE:  
: JSR PC,DVMODS  
:--

034064	112777	000141	155704	DVMODS: MOV	#141,@BSEL0	:SET UP CONTROL IN
034072	004737	035026		JSR	PC,TOORIO	:GO TIME OUT CHECK
034076	017737	155704	010206	MOV	@SEL4,MODS	:SET UP MODEM STATUS
034104	013777	007206	155700	MOV	CONTIN,@SEL6	:SET UP OLD CONTROL IN
034112	004737	034720		JSR	PC,CLR	
034116	000207			RTS	PC	:RETURN TO CALLER

5803  
5804  
5805  
5806  
5807  
5808  
5809  
5810  
5811  
5812  
5813  
5814  
5815  
5816  
5817  
5818  
5819  
5820  
5821  
5822  
5823  
5824  
5825  
5826  
5827  
5828  
5829  
5830  
5831  
5832  
5833  
5834  
5835  
5836  
5837  
5838  
5839  
5840  
5841

```
.SBTTL                DEVICE QUEUE RECEIVE SPACE SUBROUTINE
:++
:FUNCTIONAL DESCRIPTION:
:   DVRXQ - THIS SUB ROUTINE QUES THE REC BUFFER SPACE TO THE
:           DEVICE, THEN CLEARS THE QRX BIT OF THE FLAG WORD.
:
:INPUTS:
:   DVRXA = ADDRESS OF RX BUFFER SPACE
:   DVRCC = BYTE CHAR COUNT OF RX BUFFER
:   QRX FLAG BIT = SET BY CALLING ROUTINE
:
:OUTPUTS:
:   QRX FLAG BIT = CLEARED BY ROUTINE
:
:SUBORDINATE ROUTINES USED:
:
:   'TOORIO' - TIME OUT OR OUTPUT INTERRUPT OR INPUT INTERRUPT
:   'CLRAW'  - CLEAR RQI AND WAIT FOR RDI TO CLEAR
:
:CALLING SEQUENCE:
:   JSR      PC,DVRXQ
:--
```

```
034120
034120 032737 000004 007232
034126 001423
034130 042737 000004 007232
034136 112777 000144 155632
034144 004737 035026
034150 017737 155632 010206
034156 013777 007126 155622
034164 013777 007130 155620
034172 004737 034720
034176 000207
```

```
DVRXQ:
BIT      #QRX,FLAG
BEQ      DVREX          ;IF NOT RX THEN EXIT
                        ;ELSE QUE RX
                        ;CLEAR FLAG FOR RX
BIC      #QRX,FLAG
MOVB    #144,@BSELO
JSR     PC,TOORIO      ;GO CHECK FOR IN OR OUT
MOV     @SEL4,MODS     ;SET UP NEW MOD STATUS
MOV     DVRXA,@SEL4
MOV     DVRCC,@SEL6
JSR     PC,CLRAW
DVREX:  RTS           ;LOAD CC AND ADDR
                        ;CLEAR AND WAIT
                        ;RETURN TO CALLER
```

```

5842 .SBTTL                DEVICE TRANSMIT AND RECEIVE SUBROUTINE
5843
5844
5845
5846 :++
5847 : FUNCTIONAL DESCRIPTION:
5848 : DVTXRX-DEVICE TRANSMIT AND RECEIVE ROUTINE
5849 : THIS CODE QUES THE TRANSMIT BUFFER TO THE DEVICE
5850 : IF NEEDED. THE CODE THEN WAITS FOR A TX COMPLE,
5851 : RX COMPLETE OR BOTH. THE CODE REPORTS A TIME OUT
5852 : ERROR IF NO BACC OUTPUT INTERRUPT IS RECIEVED BEFORE
5853 : 60 SECONDS. AFTER REPORTING ERROR TIMER IS RE STARTED
5854 : AND DEVICE WILL CONTINUE TO WAIT FOR INTERRUPT. CODE
5855 : ALSO REPORTS ERROR IF INPUT INTERRUPT OCCURS WHEN
5856 : EXPECTING OUTPUT INTERRUPT;WHEN RX BACC OCCURS WHEN
5857 : EXPECTING TX,AND WHEN TX INT. OCCURS WHEN EXPECTING
5858 : RECIEVE.
5859
5860 : INPUTS:
5861 : 'DVTXA' = ADDRESS OF TRANSMIT MSG.
5862 : 'DVTCC' = BYTE COUNT OF TRANSMIT MSG.
5863 : 'QTX' BIT = SET IF TRANSMIT REQUESTED
5864 : 'ETX' BIT = SET IF TRNASMIT EXPECTED
5865 : 'ERX' BIT = SET IF RECIEVE EXPECTED
5866
5867 : OUTPUTS:
5868 : 'DVTXA' = ADDRESS OF TX MSG. COMPLETED
5869 : 'DVTCC' = BYTE COUNT OF TX MSG. COMPLETED
5870 : 'QTX' = SET IF TX COMPLETED
5871 : 'DVRXA' = ADDRESS OF RX MSG. COMPLETED
5872 : 'DVRCC' = BYTE COUNT OF RX MSG. COMPLETED
5873 : 'QRX' = SET IF RX COMPLETED
5874
5875 : SUBORDINATE ROUTINES USED:
5876
5877 : 'TOORIO' - TIME OUT OR OUTPUT INTERRUPT OR INTPUT INTERRUPT
5878 : 'CLRWA' - CLEAR RQI AND WAIT FOR RDI TO CLEAR
5879 : 'LGDVE' - LOG DEVICE ERROR TO EVENT LOG
5880 : 'OUTHDL' - OUTPUT INTERRUPT HANDLER CODE
5881
5882 : CALLING SEQUENCE:
5883 : JSR PC,DVTXRX
5884 :--
5885
5886 034200 032737 000010 007232 DVTXRX: BIT #QTX,FLAG ;ANY TX TO QUE
5887 034206 001423 BEQ DVTR3 ;IF NOT GO WAIT FOR OUPUT
5888 034210 042737 000010 007232 BIC #QTX,FLAG ;CLEAR FLAG
5889 034216 112777 000140 155552 MOVB #140,@BSEL0
5890 034224 004737 035026 JSR PC,TOORIO ;GO CHECK FOR IN OR OUT
5891 034230 017737 155552 010206 MOV @SEL4,MODS ;PUT IN NEW MOD STAT
5892 034236 013777 007112 155542 MOV DVTXA,@SEL4
5893 034244 013777 007114 155540 MOV DVTCC,@SEL6
5894 034252 004737 034720 JSR PC,CLRWA ;CLEAR RQI ANDWAIT
5895 034256
5896 034256 012737 000074 007276 DVTR3: MOV #60.,TIMERS ;SET TIMER FOR 60 SECS
5897 034264 032737 000060 007232 TOINOT: BIT #CRX+#CTX,FLAG ;IS IT TX OR RX COMP ALREADY?

```

5898	034272	001071			BNE	DVTR4		;IS SO EXIT		
5899										
5900	034274	005737	007276		TST	TIMERS		;IS TIMER EXPIRED		
5901	034300	001022			BNE	TOIN1				
5902	034302	012737	016424	007176	MOV	#DVEM4,TEMP2				
5903	034310	017737	155462	007200	MOV	@SELO,TEMP3				
5904	034316	017737	155460	007202	MOV	@SEL2,TEMP4				
5905	034324	004737	020774		JSR	PC,LGDVE				
5906	034330	005237	007144		INC	ERRCNT				
5907	034334				ERRSOFT	12,DVEM4,ERR13				
5908	034334	104457							TRAP	C\$ERSOFT
5909	034336	000014							.WORD	12
5910	034340	016424							.WORD	DVEM4
5911	034342	020450							.WORD	ERR13
5912	034344	000744			BR	DVTR3		;RETURN TO CHECK TIMER		
5913										
5914										
5915	034346				TOIN1:	BREAK				
5916	034346	104422							TRAP	C\$BRK
5917	034350	032737	000001	007232	BIT	#ININT,FLAG		;IS IT INPUT INTERRUPT		
5918	034356	001425			BEQ	TOIN2		;IF SO LOG ERROR		
5919										
5920	034360	012737	016516	007176	MOV	#DVEM5,TEMP2				
5921	034366	017737	155404	007200	MOV	@SELO,TEMP3				
5922	034374	017737	155402	007202	MOV	@SEL2,TEMP4				
5923	034402	004737	020774		JSR	PC,LGDVE				
5924	034406	042737	000001	007232	BIC	#ININT,FLAG		;CLEAR BIT		
5925	034414	005237	007144		INC	ERRCNT				
5926	034420				ERRSOFT	13,DVEM5,ERR13				
5927	034420	104457							TRAP	C\$ERSOFT
5928	034422	000015							.WORD	13
5929	034424	016516							.WORD	DVEM5
5930	034426	020450							.WORD	ERR13
5931	034430	000715			BR	TOINOT				
5932										
5933	034432	032737	000002	007232	TOIN2:	BIT	#OTINT,FLAG			
5934	034440	001711			BEQ	TOINOT		;IF NOT OUTPUT GO BACK AND		
5935								;CHECK TIMER AGAIN		
5936	034442	004737	035150		JSR	PC,OUTHDL		;ELSE HANDLE OUTPUT AND RETURN		
5937	034446	032737	000060	007232	BIT	#CTX+#CRX,FLAG		;IS IT TX OR RX		
5938	034454	001703			BEQ	TOINOT		;IF NOT GO BACK AND TRY AGAIN		
5939	034456	032737	000020	007232	DVTR4:	BIT	#CTX,FLAG	;IS IT TX		
5940	034464	001440			BEQ	DVTR5		;IF NOT TRY RX		
5941	034466	032737	000200	007232	BIT	#ETX,FLAG		;IF SO SHOULD IT BE		
5942	034474	001020			BNE	DVTR4A		;IF IT SHOULD GO TO 4A		
5943	034476	012737	017041	007176	MOV	#DVEM9,TEMP2		;ELSE LOG ERROR		
5944	034504	013737	035746	007200	MOV	TSEL4,TEMP3				
5945	034512	013737	035750	007202	MOV	TSEL6,TEMP4				
5946	034520	004737	020774		JSR	PC,LGDVE				
5947	034524				ERRSOFT	14,DVEM9,ERR13		;REPORT ERROR		
5948	034524	104457							TRAP	C\$ERSOFT
5949	034526	000016							.WORD	14
5950	034530	017041							.WORD	DVEM9
5951	034532	020450							.WORD	ERR13
5952										
5953	034534	000411			BR	DVTR4B		;THEN CLEAR COMPL.FLAG		

```
5954 034536 013737 035746 007112 DVTR4A: MOV TSEL4,DVTXA
5955 034544 013737 035750 007114 MOV TSEL6,DVTCC
5956 034552 052737 000010 007232 BIS #QTX,FLAG ;AND SET TX COMPL FLAG
5957 034560 042737 000020 007232 DVTR4B: BIC #CTX,FLAG ;ELSE CLEAR FLAG
5958 034566 032737 000040 007232 DVTR5: BIT #CRX,FLAG ;IS IT RX TOO?
5959 034574 001440 BEQ DVTR5 ;IF NOT THEN EXIT.
5960 034576 032737 000100 007232 BIT #ERX,FLAG ;TEST IS THIS SUPPOSED TO BE RX
5961 034604 001020 BNE DVTR5A ;IF YES PROCESS AS SUCH
5962 034606 012737 016752 007176 MOV #DVEM8,TEMP2
5963 034614 013737 035752 007200 MOV RSEL4,TEMP3
5964 034622 013737 035754 007202 MOV RSEL6,TEMP4 ;ELSE
5965 034630 004737 020774 JSR PC,LGDVE ;LOG ERROR
5966 034634 ERRSOFT 15,DVEM8,ERR13
5967 034634 104457 TRAP CSERSOFT
5968 034636 000017 .WORD 15
5969 034640 016752 .WORD DVEM8
5970 034642 020450 .WORD ERR13
5971
5972 034644 000411 BR DVTRX1 ;AND EXIT
5973 034646 013737 035752 007126 DVTR5A: MOV RSEL4,DVRXA
5974 034654 013737 035754 007130 MOV RSEL6,DVRCC
5975 034662 052737 000004 007232 BIS #QRX,FLAG
5976 034670 042737 000040 007232 DVTRX1: BIC #CRX,FLAG ;CLEAR FLAG FOR RX DONE
5977 034676 000207 DVTR5: RTS PC ;AND EXIT
5978
```

5979  
 5980  
 5981  
 5982  
 5983  
 5984  
 5985  
 5986  
 5987  
 5988  
 5989  
 5990  
 5991  
 5992  
 5993  
 5994  
 5995  
 5996  
 5997  
 5998  
 5999  
 6000  
 6001  
 6002  
 6003  
 6004  
 6005  
 6006  
 6007  
 6008  
 6009  
 6010  
 6011  
 6012  
 6013  
 6014  
 6015  
 6016  
 6017  
 6018  
 6019  
 6020  
 6021  
 6022  
 6023  
 6024  
 6025  
 6026  
 6027  
 6028  
 6029  
 6030  
 6031  
 6032  
 6033  
 6034

; DEVICE DEPENDENT SUBROUTINES

.SBTTL DEVICE INTERRUPT SERVICE ROUTINES

034700  
 034700  
 034700 052737 000001 007232  
 034706  
 034706 000002  
 034710  
 034710  
 034710 052737 000002 007232  
 034716  
 034716 000002

BGNSRV DVINS  
 BIS #ININT,FLAG  
 ENDSRV  
 BGNSRV DVOUTS  
 BIS #OTINT,FLAG  
 ENDSRV

DVINS::  
 L10021:  
 RTI  
 DVOUTS::  
 L10022:  
 RTI

++  
 : FUNCTIONAL DESCRIPTION:  
 : CLRAW - CLEAR RQI AND WAIT FOR RDI TO GO AWAY  
 : THIS CODE CLEARS THE INPUT REQUEST BIT(RQI) SETS A  
 : TIMER UP TO TIME 50(OCTAL) TICKS AND MAKES SURE  
 : RDI CLEARS BEFORE TIMER EXPIRES. IF TIMER EXPIRES  
 : CODE REPORTS ERROR AND SETS UP TIMER AND WAITS AGAIN.

: SUBORDINATE ROUTINES USED:  
 : 'LGDVE' - LOG DEVICE ERROR (TIME OUT)

: CALLING SEQUENCE:  
 : JSR PC,CLRAW  
 :--

034720 011637 007214  
 034724 042777 000040 155044  
 034732 012737 000050 007272  
 034740 005737 007272  
 034744 001406  
 034746 104422  
 034750 032777 000200 155020  
 034756 001370  
 034760 000207  
 034762 012737 016166 007176  
 034770 017737 155002 007200  
 034776 017737 155000 007202  
 035004 004737 020774  
 035010 005237 007144

CLRAW: MOV (SP),PCADD ;SAVE PC OF CALLING ROUTINE  
 BIC #RQI,@SELO  
 CLRA3: MOV #50,TIMER1 ;SET UP TIMER FOR 50(OCTAL) TICKS  
 CLRA1: TST TIMER1  
 BEQ CLRA2 ;IF TIMER EXPIRED ERROR  
 BREAK  
 BIT #RDI,@SELO ;IS RDI CLEAR TRAP CSBRK  
 BNE CLRA1 ;IF NOT GO CHECK TIMER  
 ; ELSE  
 ; RETURN TO CALLER  
 CLRA2: RTS PC  
 MOV #DVEMO,TEMP2  
 MOV @SELO,TEMP3  
 MOV @SEL2,TEMP4  
 JSR PC,LGDVE ;LOG DEVEICE EVENT 0  
 INC ERRCNT



6035	035014	
6036	035014	104457
6037	035016	000020
6038	035020	016166
6039	035022	020372
6040	035024	000742

ERRSOFT 16,DVEMO,ERR9 ;WHILE WAITING FOR RDI

TRAP	C\$ERSOFT
.WORD	16
.WORD	DVEMO
.WORD	ERR9

BR CLRA3 ;RESET TIMER AND CONTINUE

6041 .SBTTL TIME OUT OR INPUT INT. OR OUTPUT INT.  
 6042  
 6043  
 6044

++  
 : FUNCTIONAL DESCRIPTION:  
 : TOORIO - TIME OUT OR INPUT INTERRUPT OR OUTPUT INTERRUPT  
 : THIS ROUTINE SETS UP A TIMER FOR 100 (OCTAL) TICKS  
 : THEN CHECKS FOR TIME OUT,OR INPUT INTERRUPT,OR OUTPUT  
 : INTERRUPT. IF TIME OUT OCCURS IT REPORTS ERROR AND  
 : RESTARTS TIMER. IF INPUT INTERRUPT OCCURS RETURN TO CALLER  
 : IF OUTPUT INTERRUPT OCCURS LOG IT AND CONTINUE WAITING FOR  
 : INPUT INTERRUPT.

: USE OF FLAGS:  
 : 'OTINT' - SET BY OUTPUT INT ROUTINE  
 : 'ININT' - SET BY INPUT INT. ROUTINE  
 : CLEARED BY THIS ROUTINE.

: SUBORDINATE ROUTINES USED:  
 : 'OUTHDL' - OUTPUT INTERRUPT HANDLER

: CALLING SEQUENCE:  
 : JSR PC,TOORIO

```

6066
6067 035026 011637 007214 TOORIO: MOV (SP),PCADD ;SAVE ADDR. OF CALLING ROUTINE
6068 035032 012737 000100 007272 MOV #100,TIMER1 ;SET UP TIMER
6069 035040 005737 007272 TOOR3: TST TIMER1 ;IS TIME EXPIRED
6070 035044 001022 BNE TOOR1 ;IF NOT CONTINUE
6071 ;IF YES ERROR
6072 035046 012737 016254 007176 MOV #DVEM1,TEMP2
6073 035054 017737 154722 007202 MOV @SEL2,TEMP4
6074 035062 017737 154710 007200 MOV @SEL0,TEMP3
6075 035070 004737 020774 JSR PC,LGDVE
6076 035074 005237 007144 INC ERRCNT
6077 035100 ERRSOFT 17,DVEM1,ERR9
6078 035100 104457 TRAP CSERSOFT
6079 035102 000021 .WORD 17
6080 035104 016254 .WORD DVEM1
6081 035106 020372 .WORD ERR9
6082 035110 000746 BR TOORIO
6083
6084 035112 TOOR1: BREAK TRAP CSBRK
6085 035112 104422
6086 035114 032737 000002 007232 BIT #OTINT,FLAG ;IS THERE AN OUTPUT
6087 ;PENDING
6088 035122 001402 BEQ TOOR2 ;IF NOT GO TO 2
6089 ;ELSE GO HANDL IT
6090 035124 004737 035150 TOOR2: JSR PC,OUTHDL
6091 035130 032737 000001 007232 BIT #ININT,FLAG ;IS THERE AN INPUT PENDING
6092 035136 001740 BEQ TOOR3 ;IF NOT GO BACK TO TIMER CK.
6093 035140 042737 000001 007232 BIC #ININT,FLAG ;ELSE CLEAR THE INPUT PEND FLAG
6094 035146 000207 RTS PC ;AND RETURN TO CALLER
6095
  
```

6096  
 6097  
 6098  
 6099  
 6100  
 6101  
 6102  
 6103  
 6104  
 6105  
 6106  
 6107  
 6108  
 6109  
 6110  
 6111  
 6112  
 6113  
 6114  
 6115  
 6116  
 6117  
 6118  
 6119  
 6120  
 6121  
 6122  
 6123  
 6124  
 6125  
 6126  
 6127  
 6128  
 6129  
 6130  
 6131  
 6132  
 6133  
 6134  
 6135  
 6136  
 6137  
 6138  
 6139  
 6140  
 6141  
 6142  
 6143  
 6144  
 6145  
 6146  
 6147  
 6148  
 6149  
 6150  
 6151

.SBTTL

OUTPUT INTERRUPT HANDLER

..\*\*

FUNCTIONAL DESCRIPTION:

OUTHDL - OUTPUT INTERRUPT HANDLER  
 THIS ROUTINE IS CALLED WHEN AN OUTPUT INTERRUPT HAS SET  
 THE 'OTINT' BIT IN THE 'FLAG' WORD. IT CHECKS FOR  
 AN RDO SIGNAL IF NO RDO THEN REPORT ILLEGAL INTERRUPT.  
 THEN IT CHECKS FOR BACC OUT IF NOT BACC OUT REPORT THE  
 TYPE OF OUTPUT ERROR. IF BACC OUT FIND IF RX OR TX  
 IF RX SET CRX BIT AND MOVE ADDR AND BYTE COUNT TO RSEL4  
 AND RSEL6. IF TX SET CTX BIT AND MOVE ADDR AND BYTE COUNT  
 TO TSEL4 AND TSEL6. CLEAR OTINT FLAG AND RETURN TO CALLER.

USE OF FLAGS:

'OTINT' - SET BY OUPUT ROUTINE  
 CLEARED BY THIS ROUTINE  
 'DMRRUN' - SET BY DVINIT ROUTINE IF THIS IS DMR  
 CHECKED AND CLEARED BY THIS ROUTINE.  
 'CTX' - SET IF TRANSMIT COMPLETED  
 'CRX' - SET IF RECIEVE COMPLETED

SUBORDINATE ROUTINES USED:

'LGDVE' -LOG DEVICE ERRORS TO EVENT LOG

CALLING SEQUENCE

JSR PC,OUTHDL

---

```

OUTHDL: MOV      (SP),PCADD      ;SAVE ADDR. OF CALLING ROUTINE
        BIC      #OTINT,FLAG
        BIT      #RDO,@SEL2    ;CLEAR PEND FLAG AND CHK FOR RDO
        BNE     OUTH1         ;IF RDO OK ...ELSE LOG ERROR
        MOV      #DVEM6,TEMP2
        MOV      @SEL2,TEMP3
        MOV      @SEL6,TEMP4
        JSR     PC,LGDVE      ;GO LOG ERROR
        INC     ERRCNT
        ERRSOF 18,DVEM6,ERR9
  
```

```

TRAP  C$ERSOF 18
.WORD 18
.WORD DVEM6
.WORD ERR9
  
```

;EXIT TEST IF ERROR

ESCAPE TST

```

TRAP  C$ESCAPE
.WORD L10020-
  
```

```

OUTH1: BIT      #BACC,@SEL2    ;IS THE OUTPUT BACC
        BNE     1$            ; BR IF NO
        JMP     OUTH2         ;IF SO GO TO 2
  
```

```

035150 011637 007214
035154 042737 000002 007232
035162 032777 000200 154612
035170 001023
035172 012737 016610 007176
035200 017737 154576 007200
035206 017737 154600 007202
035214 004737 020774
035220 005237 007144
035224 104457
035226 000022
035230 016610
035232 020372
035234
035234 104410
035236 000522
035240 032777 000001 154534
035246 001002
035250 000137 035660
  
```

```

6152                                     ;ELSE LOG ERROR AND PRINT IT
6153 035254 017737 154532 007202 1$:  MOV  @SEL6,TEMP4
6154                                     ; IF NO BUFFER OUTPUT JUST COUNT THEM
6155
6156 035262 032737 000004 007202      BIT  #BIT2,TEMP4
6157 035270 001404                                     BEQ  OUTH6
6158                                     ;IF NO BUFF INC COUNT AND EXIT
6159 035272 005237 007140                                     INC  NOBUF
6160 035276 000137 035736                                     JMP  OUTH6X
6161
6162 035302 023727 012024 000006  OUTH6:  CMP  OPTYP,#6
6163 035310 002420                                     BLT  51$
6164 035312 032737 000040 007202      BIT  #BIT5,TEMP4
6165 035320 001414                                     BEQ  51$
6166 035322 032737 001000 007232      BIT  #DMRRUN,FLAG
6167 035330 001405                                     BEQ  52$
6168 035332 042737 001000 007232      BIC  #DMRRUN,FLAG
6169 035340 000137 035736                                     JMP  OUTH6X
6170 035344 012737 020000 007204 52$:  MOV  #RUNSBM,CONOTM
6171 035352 012737 016665 007176 51$:  MOV  #DVEM7,TEMP2
6172 035360 017737 154416 007200      MOV  @SEL2,TEMP3
6173
6174 035366 004737 020774                                     JSR  PC,LGDVE
6175 035372 012737 013456 007204      MOV  #LPO,CONOTM
6176 035400 032737 000001 007202      BIT  #BIT0,TEMP4
6177 035406 001403                                     BEQ  1$
6178 035410 012737 017340 007204      MOV  #DATCKM,CONOTM
6179 035416 032737 000002 007202 1$:  BIT  #BIT1,TEMP4
6180 035424 001403                                     BEQ  2$
6181 035426 012737 017327 007204      MOV  #TIMOM,CONOTM
6182 035434 032737 000010 007202 2$:  BIT  #BIT3,TEMP4
6183 035442 001403                                     BEQ  4$
6184 035444 012737 017307 007204      MOV  #DDCMRM,CONOTM
6185 035452 032737 000020 007202 4$:  BIT  #BIT4,TEMP4
6186 035460 001403                                     BEQ  5$
6187 035462 012737 017275 007204      MOV  #LOSDAM,CONOTM
6188 035470 032737 000100 007202 5$:  BIT  #BIT6,TEMP4
6189 035476 001403                                     BEQ  6$
6190 035500 012737 017262 007204      MOV  #DISCOM,CONOTM
6191 035506 032737 000200 007202 6$:  BIT  #BIT7,TEMP4
6192 035514 001403                                     BEQ  7$
6193 035516 012737 017242 007204      MOV  #DDCSRPM,CONOTM
6194 035524 032737 000400 007202 7$:  BIT  #BIT8,TEMP4
6195 035532 001403                                     BEQ  8$
6196 035534 012737 017224 007204      MOV  #NXMM,CONOTM
6197 035542 032737 001000 007202 8$:  BIT  #BIT9,TEMP4
6198 035550 001403                                     BEQ  9$
6199 035552 012737 017204 007204      MOV  #PROEM,CONOTM
6200 035560 023727 012024 000006 9$:  CMP  OPTYP,#6
6201 035566 002416                                     BLT  11$
6202 035570 032737 002000 007202      BIT  #BIT10,TEMP4
6203 035576 001403                                     BEQ  10$
6204 035600 012737 020022 007204      MOV  #RXIDM,CONOTM
6205 035606 032737 004000 007202 10$:  BIT  #BIT11,TEMP4
6206 035614 001403                                     BEQ  11$
6207 035616 012737 020046 007204      MOV  #CTSFM,CONOTM

```

```
6208 035624 032737 010000 007202 11$: BIT #BIT12,TEMP4 ;IS THIS CD GLITCHED
6209 035632 001403 BEQ 12$ ;BR IF NO
6210 035634 012737 020032 007204 MOV #CDGLM,CONOTM ;IF SO SET UP MESSAGE
6211
6212 035642 005237 007144 12$: INC ERRCNT
6213 035646 ERRSOFT 19,DVEM7,ERR8
6214 035646 104457
6215 035650 000023 TRAP C$ERSOFT
6216 035652 016665 .WORD 19
6217 035654 020310 .WORD DVEM7
6218 035656 000427 .WORD ERR8
6219 BR OUTHEX ;CLEAR RDO AND RETURN TO CALLER
6220 035660 OUTH2:
6221 035660 032777 000004 154114 BIT #RXBIT,@SEL2 ;IS THIS RX BACC OUT
6222 035666 001012 BNE OUTH3 ;IF NOT THEN IT MUST BE TX.
6223 035670 052737 000020 007232 BIS #CTX,FLAG
6224 035676 017737 154104 035746 MOV @SEL4,TSEL4
6225 035704 017737 154102 035750 MOV @SEL6,TSEL6
6226 035712 000411 BR OUTHEX
6227
6228 035714 052737 000040 007232 OUTH3: BIS #CRX,FLAG ;SET RX COMPL
6229 035722 017737 154060 035752 OUTH4: MOV @SEL4,RSEL4 ;THEN MOVE TO TEMP
6230 035730 017737 154056 035754 MOV @SEL6,RSEL6 ;AND SEL6 TO TEMP
6231 035736 042777 000200 154036 OUTHEX: BIC #RDO,@SEL2 ;CLEAR RDO
6232 035744 000207 RTS PC ;RETURN TO CALLER
6233 035746 000000 TSEL4: .WORD 0
6234 035750 000000 TSEL6: .WORD 0
6235 035752 000000 RSEL4: .WORD 0
6236 035754 000000 RSEL6: .WORD 0
6237
6238 035756 000207 RTS PC
6239
6240
```

6241  
6242  
6243  
6244 035760  
6245 035760  
6246 035760 104401  
6247

.EVEN  
ENDTST

L10020: TRAP C\$ETST

CZCLPAO DMR,DMC-11 DATA COMM. LINK TEST  
CZCLKA.P11 18-APR-80 09:24

MACY11 30A(1052) H 12 18-APR-80 09:24 PAGE 151  
OUTPUT INTERRUPT HANDLER

SEQ 0150

6248  
6249

6250  
6251  
6252  
6253  
6254  
6255  
6256  
6257  
6258  
6259  
6260  
6261  
6262  
6263  
6264  
6265  
6266  
6267  
6268  
6269  
6270  
6271  
6272  
6273  
6274  
6275  
6276  
6277  
6278  
6279  
6280  
6281  
6282  
6283  
6284  
6285  
6286  
6287  
6288  
6289  
6290  
6291  
6292  
6293  
6294  
6295  
6296  
6297  
6298  
6299  
6300  
6301  
6302  
6303  
6304  
6305

.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.  
:--

BGNHRD

035762  
035762 000025  
035764

.WORD L10023-L\$HARD/2  
L\$HARD::

.SBTTL DEVICE INDEPENDENT SECTION

GPRML DPLX,0,1,YES

035764  
035764 000130  
035766 036036  
035770 000001

.WORD T\$CODE  
.WORD DPLX  
.WORD 1

.SBTTL DEVICE DEPENDENT SECTION

GPRMA CSRADR,2,0,160000,177776,YES

035772  
035772 001031  
035774 036067  
035776 160000  
036000 177776

.WORD T\$CODE  
.WORD CSRADR  
.WORD T\$LOLIM  
.WORD T\$HILIM

GPRMA VECTOR,4,0,300,776,YES

036002  
036002 002031  
036004 036115  
036006 000300  
036010 000776

.WORD T\$CODE  
.WORD VECTOR  
.WORD T\$LOLIM  
.WORD T\$HILIM

GPRMD PRIOR,6,0,340,4,7,YES

036012  
036012 003032  
036014 036150  
036016 000340  
036020 000004  
036022 000007

.WORD T\$CODE  
.WORD PRIOR  
.WORD 340  
.WORD T\$LOLIM  
.WORD T\$HILIM

: GPRMD DEVPRM,10,D,17,0,15.,YES

GPRMD OPTN,12,0,7,0,7,YES

036024  
036024 005032  
036026 036176  
036030 000007  
036032 000000  
036034 000007

.WORD T\$CODE  
.WORD OPTN  
.WORD 7  
.WORD T\$LOLIM  
.WORD T\$HILIM

: GPRMD BAUD,14,0,7,0,7,YES  
: GPRMD LININ,16,0,7,0,7,YES



6306 036036  
6307  
6308 036036  
6309  
6310

ENDHRD

L10023: .EVEN

.NLIST BEX

;DEVICE INDEPENDENT QUESTIONS

036036 052506 046114 042040 DPLX: .ASCIZ /FULL DUPLEX OPERATION : /

;DEVICE DEPENDENT QUESTION

036067 104 053105 041511 CSRADR: .ASCIZ /DEVICE CSR ADDRESS : /  
036115 111 052116 051105 VECTOR: .ASCIZ /INTERRUPT VECTOR ADDRESS: /  
036150 047111 042524 051122 PRIOR: .ASCIZ /INTERRUPT PRIORITY : /  
036176 042504 044526 042503 OPTN: .ASCIZ /DEVICE OPTION TYPE : (0=DMC,5=DMR-DMC MODE ,7=DMR)/  
;DEVPRM: .ASCIZ /DEVICE PARMETER WORD (ENABLE CRC,STRIP SYNC...) : /  
;BAUD: .ASCII /BAUD RATE TO USE ('0' FOR 2.4K, '1' FOR 4.8K;/  
: .ASCII <15><12>/ '2' FOR 9.6K, '3' FOR 19.2K, '4' FOR 56K;/  
: .ASCII <15><12>/ '5' FOR 250K, '6' FOR 500K, '7' FOR 1 MEG : /  
;LININ: .ASCIZ /LINE INTERFACE (0=423,1=...)/

6311 036262  
6312  
6313

.LIST BEX  
.EVEN

6314  
6315  
6316  
6317  
6318  
6319  
6320  
6321  
6322  
6323  
6324  
6325  
6326  
6327  
6328  
6329  
6330  
6331  
6332  
6333  
6334  
6335  
6336  
6337 036262  
6338 036262 000030  
6339  
6340  
6341 036342  
6342  
6343 036342 000000  
6344 036344 000000  
6345 036346  
6346 036346  
6347  
6348 000001

:.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 :::

SPATCH:  
.BI.KW 30

LASTAD

LSLAST::  
ENDMOD

.END

.EVEN  
.WORD 0  
.WORD 0

ACT = 000003	1660#	4483	4912	4989	5217		
ACTATV 030216	4692	4912#					
ACTBCR 030060	4710	4884#					
ACTCHK 030432	4672	4958#					
ACTCLB 027402	4773	4787#					
ACTCLP 030536	4706	4983#					
ACTCLR 027046	4670	4721#					
ACTCOP 027700	4680	4847#					
ACTCRC 030446	4701	4964#					
ACTCSE 027172	4675	4749#					
ACTCST 027320	4676	4775#					
ACTDLL 030264	4696	4926#					
ACTDME 027626	4712	4828	4831#				
ACTDMQ 027620	4713	4830#					
ACTDMS 027576	4711	4825#					
ACTDMX 027634	4832#						
ACTECH 030342	4700	4942#					
ACTEQO 030022	4684	4873#					
ACTHLP 027066	4674	4727#					
ACTLIS 030254	4695	4923#					
ACTLLP 030546	4707	4985#					
ACTLPX 030564	4980	4982	4984	4986	4989#		
ACTLXX 030626	4956	4974	4977	4990	4999#		
ACTMEX 030210	4866	4882	4904	4909#			
ACTME1 030202	4893	4895	4897	4899	4901	4908#	
ACTMOP 030516	4704	4979#					
ACTMSO 030102	4685	4892#					
ACTMS1 030110	4686	4894#					
ACTMS2 030120	4687	4896#					
ACTMS3 030130	4688	4898#					
ACTMS4 030140	4689	4900#					
ACTMS5 030150	4690	4902#					
ACTMS6 030166	4691	4905#					
ACTM2X 030312	4913	4921	4924	4927	4930	4934#	
ACTNO 030332	4699	4939#					
ACTNUF 027036	4709	4718#					
ACTNUL 027044	4669	4719#					
ACTNUM 027710	4681	4850#					
ACTOPM 030002	4682	4868#					
ACTPAS 030226	4693	4915#					
ACTPRO 030454	4702	4967#					
ACTPRT 027132	4714	4740#					
ACTQFG 030460	4959	4962	4965	4969#			
ACTREC 030246	4694	4920#					
ACTRLP 030556	4708	4987#					
ACTRPS 030506	4703	4976#					
ACTRUN 027146	4673	4744#					
ACTSHO 027056	4671	4724#					
ACTSHW 027442	4760	4779	4798#	4817			
ACTSTE 027642	4677	4837#					
ACTSTS 030440	4683	4961#					
ACTSTT 027652	4678	4840#					
ACTSTX 027660	4838	4841#					
ACTSZE 027670	4679	4844#					
ACTTAL 030304	4698	4932#					
ACTTLP 030526	4705	4981#					

ACTTRA	030274	4697	4929#					
ADDCC	022600	3718#	4598	4644				
ADDCC1	022674	3720	3735#					
ADR	= 000020	1633#						
ALCK	031142	2100	5127#					
ALCK1	031250	5182	5191#	5235				
ALCK2	031314	5192	5201#	5267				
ALCK2A	031534	5232	5237#					
ALCK3	031560	5204	5239	5242#				
ALCK3A	031652	5249	5255#					
ALCK3B	031664	5256	5258#					
ALCK3C	031660	5253	5257#	5264				
ALCK4	031706	5243	5262#					
ALCK4A	031714	5261	5265#					
ALCK4B	031726	5266	5268#					
ALCK5	031200	5181#	5257					
ALLTR	031200	5059	5082	5106	5180#			
ASSEMB	= 000010	1403						
ATVMOD	= 000027	1759#	2239					
BACC	= 000001	1803#	6149					
BAD	007211	2071#	3068	5343*				
BADCHR	= 000051	1777#	2331					
BASE	017370	3018#	3451	3452	5708			
BASM1	015010	2746#	3662					
BASM1A	014755	2738#	3445					
BASM2	015001	2744#	3681					
BASM3	014772	2742#	3673					
BASN1	021470	3449	3455#					
BIT0	= 000001	1606#	1680	1791	1803	6176		
BIT00	= 000001	1595#	1606					
BIT01	= 000002	1594#	1605					
BIT02	= 000004	1593#	1604					
BIT03	= 000010	1592#	1603					
BIT04	= 000020	1591#	1602					
BIT05	= 000040	1590#	1601					
BIT06	= 000100	1589#	1600					
BIT07	= 000200	1588#	1599					
BIT08	= 000400	1587#	1598					
BIT09	= 001000	1586#	1597					
BIT1	= 000002	1605#	1681	6179				
BIT10	= 002000	1585#	1807	6202				
BIT11	= 004000	1584#	1805	6205				
BIT12	= 010000	1583#	6208					
BIT13	= 020000	1582#						
BIT14	= 040000	1581#	1794	1804				
BIT15	= 100000	1580#						
BIT2	= 000004	1604#	1682	1789	1808	5727	5730	6156
BIT3	= 000010	1603#	1790	5727	5733	6182		
BIT4	= 000020	1602#	1683	6185				
BIT5	= 000040	1601#	1684	1792	1800	6164		
BIT6	= 000100	1600#	1809	6188				
BIT7	= 000200	1599#	1793	1801	1802	6191		
BIT8	= 000400	1598#	1806	6194				
BIT9	= 001000	1597#	1795	6197				
BLDBEX	023006	3779	3783#					
BLDBUF	022676	3761#	4474	4481	4599	4645	4794	5033







DEV4	010342	2193#	3542*	3834	3838	3842	4497*	4821*
DFPTBL	002130 G	1531#						
DIAGMC=	000000	1403						
DISCOM	017262	2999#	6190					
DLE =	000020	1704#	3367					
DLL	032262	2101	5426#					
DLLAB	017130	2981#	5453	5484				
DLLCM	013640	2616#	5442					
DLLEA	032370	5449#	5486					
DLLE1	032646	5477	5496#					
DLLE2	032514	5475#	5495					
DLLE3	032726	5505	5509#					
DLLE4	032716	5506#						
DLLE5	032746	5494	5510	5514#				
DLLE6	032600	5479	5488#					
DLLE7	032546	5482#	5499	5507	5512			
DLLE8	032666	5497	5500#					
DLLGA =	000400	1718#	5436	5476	5492			
DLLMOD=	000033	1763#	2247					
DLLM1	002647	1831	1914#	5428				
DLLM1C	002172	1831#	5429					
DLLM1E	002654	1831	1919#					
DLLM2	002654	1832	1920#	5434				
DLLM2C	002174	1832#	5435	5493				
DLLM2E	003062	1832	1961#					
DLLPRI	032344	5440#						
DLTXRX	032404	5430	5437	5460#				
DMC =	000000	1688#						
DMPE =	000053	1779#	2277					
DMPQ =	000054	1780#	2279					
DMPS =	000052	1778#	2275	4557	4827			
DMRC6 =	000004	1689#	5674					
DMRC7 =	000005	1690#						
DMRRUN=	001000	1719#	5762	6166	6168			
DMR6 =	000006	1691#	5677					
DMR7 =	000007	1692#						
DMSGAD	002176	1836#	3775					
DMSGCT	002150	1821#	3774					
DOW =	000004	1661#	4926	5748				
DPLX	036036	6271	6310#					
DSR =	000010	1790#	2133					
DUMEX	022576	3687	3693#					
DUMPSR	022442	3454	3658#	4559				
DUM1	022534	3668	3679#					
DUM2	022556	3678	3686#					
DUM3	022472	3667#	3691					
DUM4	022446	3659#	3690					
DVEM0	016166	2890#	6030	6038				
DVEM1	016254	2900#	6072	6080				
DVEM3	016340	2910#	5688	5696				
DVEM4	016424	2920#	5902	5910				
DVEM5	016516	2931#	5920	5929				
DVEM6	016610	2942#	6132	6140				
DVEM7	016665	2950#	6171	6216				
DVEM8	016752	2961#	5962	5969				
DVEM9	017041	2971#	5943	5950				











LISCK	033170	2103	5600#						
LISCKA	033216	5607#	5631	5633					
LISMOD=	000032	1762#	2245						
LISP	013564	2600#	5602						
LMDLOP=	000046	1774#	2354						
LNCNT	007136	2045#	3394	3396*	3405*				
LOE =	040000 G	1643#							
LOGCMD	021114	3369#	5363						
LOGCML	021076	3365#	5332						
LOGCMP	021060	3361#	5322						
LOGDVI	021012	3352#	5026						
LOGEOP	021132	3373#	5390						
LOGEX	021404	3383	3430#						
LOGRXC	020764	3344#	5207	5502	5621				
LOGRXQ	020746	3339#	5190	5467	5615				
LOGS1	021150	3332	3337	3342	3346	3378#			
LOGS2	021376	3424	3428#						
LOGS3	021202	3350	3359	3364	3368	3372	3376	3387#	
LOGS4	021256	3395	3404#						
LOGS5	021302	3389	3391	3412#					
LOGTXC	020730	3334#	5246	5490	5572				
LOGTXQ	020712	3329#	5199	5473	5564				
LOGUNT	007212	2075#	4250*	4252*	4253	4257			
LOOPS	003276	2001#	3819						
LOSDAM	017275	3001#	6187						
LOT =	000010 G	1632#							
LPO	013456	2001	2161	2579#	3818	6175			
LPOO	013457	2580#	3815						
LP1	013466	2002	2582#						
LP2	013477	2003	2584#						
LP3	013505	2004	2586#						
LP4	013520	2005	2588#						
LULoop=	004000	1805#	5715	5718					
LSACP	002110 G	1495#							
LSAPT	002036 G	1453#							
LSAU	025316 G	1480	4402#						
LSAUT	002070 G	1479#							
LSAUTO	025264 G	1496	4339#						
LSCCP	002106 G	1493#							
LSCLEA	025266 G	1494	4353#						
LSCO	002032 G	1449#							
LSDEPO	002011 G	1431#							
LSDESC	012042 G	1486	2421#						
LSDESP	002076 G	1485#							
LSDEVP	002060 G	1471#							
LSDISP	002124 G	1456	1516#						
LSDLY	002116 G	1501#							
LSDTP	002040 G	1455#							
LSDTYP	002034 G	1451#							
LSDU	025310 G	1482	4380#						
LSDUT	002072 G	1481#							
LSDVTY	012026 G	1472	2410#						
LSEF	002052 G	1466#							
LSENV1	002044 G	1459#							
LSETP	002102 G	1489#							
LSEXP1	002046 G	1461#							





NOD114	011414	2315#
NOD115	011430	2316#
NOD116	011434	2317#
NOD117	011450	2318#
NOD12	010444	2224#
NOD120	011454	2319#
NOD121	011470	2320#
NOD122	011474	2321#
NOD123	011510	2322#
NOD124	011514	2325#
NOD125	011520	2326#
NOD126	011524	2327#
NOD127	011530	2328#
NOD13	010450	2225#
NOD130	011534	2329#
NOD131	011540	2330#
NOD132	011544	2331#
NOD133	011546	2334#
NOD134	011552	2335#
NOD135	011556	2336#
NOD136	011572	2337#
NOD137	011576	2338#
NOD14	010464	2226#
NOD140	011612	2339#
NOD141	011616	2342#
NOD142	011622	2343#
NOD143	011626	2344#
NOD144	011632	2347#
NOD145	011636	2350#
NOD146	011660	2351#
NOD147	011664	2352#
NOD15	010470	2227#
NOD150	011700	2353#
NOD151	011704	2354#
NOD152	011726	2355#
NOD153	011732	2356#
NOD154	011754	2357#
NOD155	011760	2360#
NOD156	011764	2361#
NOD157	011770	2362#
NOD16	010474	2228#
NOD160	011774	2367#
NOD17	010506	2229#
NOD2	010354	2216#
NOD20	010512	2230#
NOD21	010524	2231#
NOD22	010530	2232#
NOD23	010532	2236#
NOD24	010536	2237#
NOD25	010552	2238#
NOD26	010556	2239#
NOD27	010574	2240#
NOD3	010356	2217#
NOD30	010600	2241#
NOD31	010616	2242#
NOD32	010622	2243#



NOD33	010640	2244#																			
NOD34	010644	2245#																			
NOD35	010662	2246#																			
NOD36	010666	2247#																			
NOD37	010712	2248#																			
NOD4	010372	2218#																			
NOD40	010716	2249#																			
NOD41	010722	2250#																			
NOD42	010740	2251#																			
NOD43	010744	2252#																			
NOD44	010756	2253#																			
NOD45	010762	2257#																			
NOD46	010766	2258#																			
NOD47	011010	2259#																			
NOD5	010374	2219#																			
NOD50	011012	2260#																			
NOD51	011036	2261#																			
NOD52	011040	2266#																			
NOD53	011044	2267#																			
NOD54	011064	2268#																			
NOD55	011070	2269#																			
NOD56	011112	2270#																			
NOD57	011116	2273#																			
NOD6	010410	2220#																			
NOD60	011122	2274#																			
NOD61	011126	2275#																			
NOD62	011132	2276#																			
NOD63	011136	2277#																			
NOD64	011142	2278#																			
NOD65	011146	2279#																			
NOD66	011152	2280#																			
NOD67	011156	2283#																			
NOD7	010412	2221#																			
NOD70	011162	2284#																			
NOD71	011166	2285#																			
NOD72	011200	2286#																			
NOD73	011204	2287#																			
NOD74	011220	2288#																			
NOD75	011224	2294#																			
NOD76	011242	2295#																			
NOD77	011246	2296#																			
NONE	= 000000	1665#																			
NOTNUF	= 000050	1776#	2223	2226	2227	2237	2276	2278	2284	2335											
NULEVT	015016	2751#	3459																		
NULL	= 000000	1736#																			
NUM	= 000014	1748#	2343																		
NXPPI	017224	2993#	6196																		
N10S	010350	2214#																			
N100S	010762	2226	2229	2256#																	
N102S	010766	2257#																			
N104S	011012	2258	2259#																		
N110S	011040	2231	2265#																		
N111S	011044	2266#																			
N112S	011070	2267	2268#																		
N114S	011162	2283#																			
N115S	011156	2240	2242	2244	2246	2248	2251	2253	2282#	2288	2295	2297	2351	2353							







RXM2	020145	3040#	5511											
RXNC	020102	3033#	5498											
RXON2	031014	2097	5054#											
RXPTR	007076	5055#												
RXQ =	000004	2026#	4460*	5015*	5016*	5031	5055	5104	5130	5216	5306			
SCM	016111	1698#	3341											
SCMD	016144	2875#	3362											
SCML	016133	2881#	3370											
SDVE	016100	2879#	3366											
SDVI	016122	2873#	3348											
SELO	011776	2877#	3353											
SEL2	012002	2373#	4268*	5672*	5681	5689	5715*	5718*	5903	5921	6020*	6026	6031	6074
		2376#	4271*	4272*	5690	5714*	5904	5922	6032	6073	6130	6133	6149	6172
		6221	6231*											
SEL4	012006	2379#	4275*	4276*	5671*	5708*	5797	5836	5837*	5891	5692*	6224	6229	5838*
SEL6	012012	2382#	4279*	4280*	5670*	5710*	5713*	5727*	5747*	5750*	5755*	5757	5798*	
		5893*	6134	6153	6225	6230								
SEOP	016155	2883#	3374											
SETEXP=	000010	1744#	2267	4565	4837									
SETTRN=	000011	1745#	2269	4840										
SHFO	014165	2655#	3826											
SHF1	014223	2663#	3851											
SHMSG	013251	2549#	4810											
SHOW =	000002	1738#	2228	4563	4724	4759	4778							
SHTAB	003250	1987#	4798	4804										
SHTEND	003257	1990#	4801											
SHTYPO	013305	1980	2554#											
SHTYP1	013314	1980	2556#											
SHTYP2	013321	1980	2557#											
SHTYP3	013326	1980	2558#											
SHTYP4	013333	1980	2559#											
SHTYP5	013341	1980	2561#											
SHTYP6	013346	1980	2562#											
SHTYP7	013354	1980	2563#											
SHTYTB	003230	1980#	4809											
SHWOP	023022	3544	3810#	4498	4822									
SIZE =	000012	1746#	2336	4844	4850									
SQD =	040000	1794#	2137											
SRXQ	016067	2871#	3340											
STADD	007146	2049#	3451*	3658	4825*									
START	024454	4166	4185#											
STATB =	000001	1680#	3390	3834	4961									
STATUS=	000016	1750#	2294											
STXC	016056	2869#	3335											
STXQ	016045	2867#	3330											
SVCGBL=	000000	1403#	1420	1429	1431	1433	1435	1437	1439	1441	1443	1445	1447	1449
		1451	1453	1455	1457	1459	1461	1463	1466	1469	1471	1473	1475	1477
		1479	1481	1483	1485	1487	1489	1491	1493	1495	1497	1499	1501	1503
		1516	1530	1531	2410	2421	3065	3082	3095	3109	3131	3152	3166	3265
		4122	4142	4158	4339	4353	4380	4402	5986	5993	6264	6345#	6346	
SVCINS-	000001	1403#	1421	1422	1423	1424	1425	1426	1427	1428	1430	1432	1434	1436
		1438	1440	1442	1444	1446	1448	1450	1452	1454	1456	1458	1460	1462
		1464	1465	1467	1468	1470	1472	1474	1476	1478	1480	1482	1484	1486
		1488	1490	1492	1494	1496	1498	1500	1502	1504	1515	1517	1529	2411
		2413	2422	2429	3067	3068	3069	3070	3071	3072	3073	3074	3075	3076
		3079	3084	3085	3086	3087	3088	3089	3092	3097	3098	3099	3100	3101











BCOMPL	1#	1403#	4165	4170	4181	4502									
BERROR	1#	1403#													
BGNAU	1#	1403#	4401												
BGNAUT	1#	1403#	4338												
BGNCLN	1#	1403#	4352												
BGNDU	1#	1403#	4379												
BGNHRD	1#	1403#	6262												
BGNHW	1#	1403#	1528												
BGNINI	1#	1403#	4157												
BGNMOD	1#	1403#	1404												
BGNMSG	1#	1403#	3064	3081	3094	3108	3130	3151	3165						
BGNPRO	1#	1403#	4141												
BGNPTA	1#	1403#													
BGNRPT	1#	1403#	4121												
BGNSEG	1#	1403#													
BGNSET	1#	1403#													
BGNSFT	1#	1403#													
BGNSRV	1#	1403#	3264	5985	5992										
BGNSUB	1#	1403#													
BGNSW	1#	1403#													
BGNTST	1#	1403#	4428												
BNCOMP	1#	1403#	4175	4193	4203	4214	4260								
BNERRO	1#	1403#													
BREAK	1#	1403#	5683	5915	6024	6084									
BRESET	1#	1403#	4359												
CKLOOP	1#	1403#													
CLI	1407#	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226
	2227	2228	2229	2230	2231	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244
	2245	2246	2247	2248	2249	2250	2251	2252	2256	2257	2258	2259	2260	2265	2266
	2267	2268	2269	2272	2273	2274	2275	2276	2277	2278	2279	2282	2283	2284	2285
	2286	2287	2293	2294	2295	2296	2299	2300	2301	2302	2305	2308	2309	2310	2311
	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2324	2325	2326	2327	2328
	2329	2330	2333	2334	2335	2336	2337	2338	2341	2342	2343	2346	2349	2350	2351
	2352	2353	2354	2355	2356	2359	2360	2361	2366						
CLOCK	1#	1403#	4189	4199											
CLOSE	1#	1403#													
CLRVEC	1#	1403#													
COMMEN	1#	1403#													
DELAY	1#	1403#													
DESCRI	1#	1403#	2420												
DEVTYP	1#	1403#	2409												
DISPAT	1#	1403#	1514												
DISPLA	1#	1403#													
DOCLN	1#	1403#													
DODU	1#	1403#													
DORPT	1#	1403#													
ENDAU	1#	1403#	4412												
ENDAUT	1#	1403#	4342												
ENDCLN	1#	1403#	4369												
ENDCOM	1#	1403#													
ENDDU	1#	1403#	4390												
ENDHRD	1#	1403#	6306												
ENDHW	1#	1403#	1559												
ENDINI	1#	1403#	4326												
ENDMOD	1#	1403#	6346												
ENDMSG	1#	1403#	3077	3090	3104	3126	3147	3161	3176						





	6139#	6140#	6141#	6146#	6147#	6214#	6215#	6216#	6217#	6246#	6263#	6270#	6271	6272	6280#
	6281	6282	6283	6285#	6286	6287	6288	6290#	6291	6292	6293	6294	6298#	6299	6300
	6301	6302	6307#	6342#	6343#	6344#									
MSGNLS	1#	1403#	3441#	3447	4220#	4228	4517#	4525	5544#	5552					
MSGNSU	1#	1403#													
MSGNTA	1#	1403#	1560#	3078#	3091#	3105#	3127#	3148#	3162#	3177#	3290#	4132#	4327#	4343#	4370#
	4391#	4413#	5989#	5996#	6245#	6307#	6308								
MSGNTE	1#	1403#	4429#												
MSHAPT	1#	1403#	1420#												
MSHNAP	1#	1403#	1420#	1459											
MSINCR	1#	1403#	1405#	1529#	3065#	3075#	3079#	3082#	3088#	3092#	3095#	3102#	3106#	3109#	3117#
	3124#	3128#	3131#	3138#	3145#	3149#	3152#	3159#	3163#	3166#	3174#	3178#	3265#	3402#	3410#
	3441#	3448	3462#	3485#	3499#	3512#	3525#	3533#	3557#	3573#	3588#	3606#	3626#	3665#	3676#
	3684#	3728#	3829#	3854#	4000#	4043#	4122#	4133#	4142#	4158#	4164#	4169#	4174#	4180#	4191#
	4201#	4213#	4220#	4229	4235#	4238#	4258#	4296#	4306#	4313#	4318#	4320#	4328#	4339#	4344#
	4353#	4358#	4360#	4363#	4371#	4380#	4392#	4402#	4414#	4429#	4430#	4447#	4492#	4501#	4507#
	4517#	4526	4537#	4546#	4575#	4590#	4616#	4636#	4733#	4813#	4858#	4888#	4951#	4997#	5328#
	5346#	5359#	5445#	5451#	5544#	5553	5605#	5628#	5684#	5694#	5908#	5916#	5927#	5948#	5967#
	5986#	5993#	6025#	6036#	6078#	6085#	6138#	6146#	6214#	6246#	6263#				
MSIOSE	1#	1403#													
MSLDRO	1#	1403#	4163#	4168#	4173#	4179#	4190#	4200#	4257#	4317#	4357#				
MSMASK	1#	1403#													
MSMCHI	1#	1403#													
MSMCLO	1#	1403#													
MSMSK1	1#	1403#													
MSPOP	1#	1403#	1560#	3078#	3091#	3105#	3127#	3148#	3162#	3177#	3290#	4132#	4149#	4327#	4343#
	4370#	4391#	4413#	5989#	5996#	6245#	6307#	6347#							
MSPRIN	1#	1403#	3067#	3084#	3097#	3111#	3120#	3133#	3141#	3154#	3168#	3399#	3407#	3459#	3482#
	3492#	3507#	3521#	3528#	3551#	3567#	3582#	3603#	3623#	3661#	3671#	3680#	3725#	3822#	3848#
	3997#	4040#	4232#	4444#	4489#	4534#	4543#	4571#	4586#	4612#	4632#	4729#	4808#	4855#	4885#
	4948#	4994#	5442#	5602#	5625#										
MSPUSH	1#	1403#	1405#	1529#	3065#	3082#	3095#	3109#	3131#	3152#	3166#	3265#	4122#	4142#	4158#
	4339#	4353#	4380#	4402#	4429#	4430	5986#	5993#	6263#						
MSPUT	1#	1403#	3067#	3084#	3097#	3111#	3120#	3133#	3141#	3154#	3168#	3399#	3407#	3459#	3482#
	3492#	3507#	3521#	3528#	3551#	3567#	3582#	3603#	3623#	3661#	3671#	3680#	3725#	3822#	3848#
	3997#	4040#	4232#	4292#	4302#	4309#	4444#	4489#	4534#	4543#	4571#	4586#	4612#	4632#	4729#
	4808#	4855#	4885#	4948#	4994#	5442#	5602#	5625#	5625#						
MSPUT1	1#	1403#	3067#	3069	3071	3072	3073	3084#	3085	3086	3097#	3098	3099	3100	3111#
	3112	3113	3114	3115	3120#	3121	3122	3133#	3134	3135	3136	3141#	3142	3143	3154#
	3155	3156	3157	3168#	3169	3170	3171	3172	3399#	3400	3407#	3408	3459#	3460	3482#
	3483	3492#	3493	3494	3495	3496	3497	3507#	3508	3509	3510	3521#	3522	3523	3528#
	3529	3530	3531	3551#	3552	3553	3554	3555	3567#	3568	3569	3570	3571	3582#	3583
	3584	3585	3586	3603#	3604	3623#	3624	3661#	3662	3663	3671#	3673	3674	3680#	3681
	3682	3725#	3726	3822#	3823	3824	3825	3826	3827	3848#	3849	3850	3851	3852	3997#
	3998	4040#	4041	4232#	4233	4292#	4293	4294	4295	4302#	4303	4304	4305	4309#	4310
	4311	4312	4444#	4445	4489#	4490	4534#	4535	4543#	4544	4571#	4572	4573	4586#	4587
	4588	4612#	4613	4614	4632#	4633	4634	4729#	4730	4731	4808#	4809	4810	4811	4855#
	4856	4885#	4886	4948#	4949	4994#	4995	5442#	5443	5602#	5603	5625#	5626		
MSRADI	1#	1403#	3444#	4221#	4520#	5547#	6270#	6280#	6285#	6290#	6298#				
MSRBRO	1#	1403#													
MSRNRO	1#	1403#	4190#	4192	4200#	4202	4257#	4259							
MSSETS	1#	1403#	1405#	1529#	3065#	3082#	3095#	3109#	3131#	3152#	3166#	3265#	4122#	4142#	4158#
	4339#	4353#	4380#	4402#	4430#	5986#	5993#	6263#							
MSSTAR	1#	1403#													
MSVC	1#	1403#	3067#	3075	3078#	3079	3084#	3088	3091#	3092	3097#	3102	3105#	3106	3111#
	3117	3120#	3124	3127#	3128	3133#	3138	3141#	3145	3148#	3149	3154#	3159	3162#	3163

	3168#	3174	3177#	3178	3181#	3399#	3402	3407#	3410	3441#	3459#	3462	3482#	3485	3492#
	3499	3507#	3512	3521#	3525	3528#	3533	3551#	3557	3567#	3573	3582#	3588	3603#	3606
	3623#	3626	3661#	3665	3671#	3676	3680#	3684	3725#	3728	3822#	3829	3848#	3854	3997#
	4000	4040#	4043	4132#	4133	4163#	4164	4168#	4169	4173#	4174	4179#	4180	4190#	4191
	4200#	4201	4213#	4220#	4232#	4235	4238#	4257#	4258	4292#	4296	4302#	4306	4309#	4313
	4317#	4318	4320#	4327#	4328	4343#	4344	4357#	4358	4360#	4363#	4370#	4371	4384#	4391#
	4392	4406#	4413#	4414	4444#	4447	4489#	4492	4501#	4507#	4517#	4534#	4537	4543#	4546
	4571#	4575	4586#	4590	4612#	4616	4632#	4636	4729#	4733	4808#	4813	4855#	4858	4885#
	4888	4948#	4951	4994#	4997	5328	5346	5359	5442#	5445	5451	5544#	5602#	5605	5625#
	5628	5684#	5694	5908	5916#	5927	5948	5967	6025#	6036	6078	6085#	6138	6146#	6214
	6245#	6246													
MSLAB	1#	1403#	3075#	3079#	3088#	3092#	3102#	3106#	3117#	3124#	3128#	3138#	3145#	3149#	3159#
	3163#	3174#	3178#	3402#	3410#	3441#	3462#	3485#	3499#	3512#	3525#	3533#	3557#	3573#	3588#
	3606#	3626#	3665#	3676#	3684#	3728#	3829#	3854#	4000#	4043#	4133#	4164#	4169#	4174#	4180#
	4191#	4201#	4213#	4220#	4235#	4238#	4258#	4296#	4306#	4313#	4318#	4320#	4328#	4344#	4358#
	4360#	4363#	4371#	4392#	4414#	4447#	4492#	4501#	4507#	4517#	4537#	4546#	4575#	4590#	4616#
	4636#	4733#	4813#	4858#	4888#	4951#	4997#	5328#	5346#	5359#	5445#	5451#	5544#	5605#	5628#
	5684#	5694#	5908#	5916#	5927#	5948#	5967#	6025#	6036#	6078#	6085#	6138#	6146#	6214#	6246#
MSSTL	1#	1403#	3075#	3079#	3088#	3092#	3102#	3106#	3117#	3124#	3128#	3138#	3145#	3149#	3159#
	3163#	3174#	3178#	3402#	3410#	3441#	3462#	3485#	3499#	3512#	3525#	3533#	3557#	3573#	3588#
	3606#	3626#	3665#	3676#	3684#	3728#	3829#	3854#	4000#	4043#	4133#	4164#	4169#	4174#	4180#
	4191#	4201#	4213#	4220#	4235#	4238#	4258#	4296#	4306#	4313#	4318#	4320#	4328#	4344#	4358#
	4360#	4363#	4371#	4392#	4414#	4447#	4492#	4501#	4507#	4517#	4537#	4546#	4575#	4590#	4616#
	4636#	4733#	4813#	4858#	4888#	4951#	4997#	5328#	5346#	5359#	5445#	5451#	5544#	5605#	5628#
	5684#	5694#	5908#	5916#	5927#	5948#	5967#	6025#	6036#	6078#	6085#	6138#	6146#	6214#	6246#
MSWORD	1#	1403#	1459#	1468	1515#	1517	3181#	3441#	3443	3444#	4220#	4222	4223#	4238#	4320#
	4363#	4384#	4406#	4507#	4517#	4519	4520#	5328#	5329	5330	5331	5346#	5347	5348	5349
	5359#	5360	5361	5362	5451#	5452	5453	5454	5544#	5546	5547#	5694#	5695	5696	5697
	5908#	5909	5910	5911	5927#	5928	5929	5930	5948#	5949	5950	5951	5967#	5968	5969
	5970	6036#	6037	6038	6039	6078#	6079	6080	6081	6138#	6139	6140	6141	6214#	6215
	6216	6217	6270#	6280#	6285#	6290#	6298#	6343	6344						
MSXFER	1#	1403#													
MODCL	1409#	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227
	2228	2229	2230	2231	2232	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245
	2246	2247	2248	2249	2250	2251	2252	2253	2257	2258	2259	2260	2261	2266	2267
	2268	2269	2270	2273	2274	2275	2276	2277	2278	2279	2280	2283	2284	2285	2286
	2287	2288	2294	2295	2296	2297	2300	2301	2302	2303	2306	2309	2310	2311	2312
	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2325	2326	2327	2328	2329
	2330	2331	2334	2335	2336	2337	2338	2339	2342	2343	2344	2347	2350	2351	2352
	2353	2354	2355	2356	2357	2360	2361	2362	2367						
OPEN	1#	1403#													
POINTE	1#	1403#	1415												
PRINTB	1#	1403#	3066	3083	3096	3110	3119	3132	3140	3153	3167				
PRINTF	1#	1403#	3398	3406	3660	3670	3679	3724	3996	4039	4231	4443	4488	4533	4542
	4570	4585	4611	4631	4728	4807	4854	4884	4947	4993	5441	5601	5624		
PRINTS	1#	1403#	3458	3481	3491	3506	3520	3527	3550	3566	3581	3602	3622	3821	3847
PRINTX	1#	1403#													
READBU	1#	1403#	4212												
READEF	1#	1403#	4162	4167	4172	4178									
RFLAGS	1#	1403#													
SETPRI	1#	1403#	4316	4356											
SETVEC	1#	1403#	4291	4301	4308										
SLASH	1#	1403#													
STARS	1#	1403#													
SVC	1#	1403#													
XFER	1#	1403#	3181#	4238#	4320#	4363#	4384#	4406#	4507#						

XFERF 1# 1403#  
XFERT 1# 1403#

. ABS. 036346 000

ERRORS DETECTED: 0

CZCLKA/I,CZCLKA.SEQ/CRF/SOL=SVC34R.MLB,CZCLKA.P11  
RUN-TIME: 21 27 3 SECONDS  
RUN-TIME RATIO: 68/52=1.3  
CORE USED: 19K (37 PAGES)