

DMV 11  
DMP 11

DMP/DMV 11 DCLT  
CZCLMCO

COPYRIGHT (c) 1981-84  
AH-F599C-MC  
FICHE 01 OF 02

JUL 1984  
digital  
Made In USA

The microfiche card displays a grid of 100 frames, arranged in 10 rows and 10 columns. Each frame contains a small table with multiple columns and rows of text and numbers. The data is organized in a regular grid pattern across the entire card. The text is small and difficult to read, but it appears to be a structured dataset, possibly a list of records or a table of values. The frames are separated by thin white lines, and the overall layout is consistent throughout the card.

DMV 11  
DMP 11

DMP/DMV 11 DCLT  
CZCLMCO

COPYRIGHT (c) 1981-84  
RH-F599C-MC  
FICHE 02 OF 02

JUL 1984  
digital  
Made In USA

Microfiche grid containing multiple frames of data, including headers like 'DMV 11' and 'DMP 11'.

Microfiche grid containing multiple frames of data, including headers like 'DMV 11' and 'DMP 11'.

29  
30

.TITLE CZCLMCO DMP/V-11 DCLT

.REM 6

IDENTIFICATION

PRODUCT CODE: AC F597C-MC  
PRODUCT NAME: CZCLMC DMP,DMV-11 DATA COMM. LINK TEST  
PRODUCT DATE: MARCH 1984  
MAINTAINER: MERRIMACK DIAGNOSTIC ENGINEERING  
AUTHOR: BRUCE LUMRS - 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 RESPONSIBLITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1981,1984 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	14 JAN-81	BRUCE RIBOLINI	ORIGINAL ISSUE, DCLT FOR THE DMP.DMV 11
B	26 OCT 81	ERNIE COOPER	ADD "SET E=T COMMAND" ADD ID OF DEVICE REQUESTING DOWNLINELOAD. ADDED NEEDED PATCHES. GENERAL CLEANUP AND ENHANCEMENT OF DOCUMENT.
C	MARCH 1984	ERNIE COOPER	ADD FIXES TO CORRECT DISCONNECT ERROR.

## 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	TRIB COMMANDS
2.5.3	STATISTICAL COMMANDS
2.5.4	RUN COMMANDS
2.5.5	PRINT COMMANDS
2.5.6	DEFAULTS
2.6	QUICK STARTUP PROCEDURE
3.0	ERROR INFORMATION
3.1	TYPES OF ERROR MESSAGES
3.2	SPECIFIC ERROR MESSAGES
3.2.1	COMMAND LINE INTERPRETER ERRORS
3.2.2	DCLT ERROR MESSAGES
3.2.3	DEVICE ERROR MESSAGES
4.0	PERFORMANCE AND PROGRESS REPORTS
4.1	PRINTING EVENT LOG
4.2	OPERATOR STATUS MESSAGES
5.0	DEVICE INFORMATION TABLES
6.0	MODE AND MESSAGE DESCRIPTIONS
6.1	MODE DESCRIPTIONS
6.1.1	TRANSMIT MODE
6.1.2	RECEIVE MODE
6.1.3	PASSIVE MODE
6.1.4	ACTIVE MODE
6.1.5	DOWN LINE LOAD
6.1.6	TALK AND LISTEN
6.1.6.1	TALK MODE
6.1.6.2	LISTEN MODE
6.1.7	MAINTENANCE LOOP SUMMARY
6.1.8	MODE SUMMARY TABLE
6.2	MESSAGE DESCRIPTIONS
7.0	OTHER INFORMATION
7.1	INTERFACING TO AN "ITEP" NODE
7.2	TROUBLESHOOTING HINTS
7.3	EXAMPLES OF COMMANDS
7.4	THINGS TO WATCH OUT FOR

## 1.0 GENERAL INFORMATION

## 1.1 PROGRAM ABSTRACT

THIS DCLT (DATA COMMUNICATION LINK TEST) PROGRAM IS MEANT TO PROVIDE FIELD SERVICE WITH A TOOL TO MAINTAIN DMP,DMV-11 TO DDCMP MULTIPOINT COMMUNICATION LINKS. THIS DCLT PROGRAM WILL PROVIDE THE COVERAGE NECESSARY TO DETECT FAILURES IN 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 CZCLM DCLT PROGRAM, THE FOLLOWING MINIMUM HARDWARE IS REQUIRED:

- A PDP 11 CPU IF DMP OR A LSI-11 CPU IF DMV
- MINIMUM OF 24K WORDS OF MEMORY
- A WORKING, LINE OR REAL-TIME CLOCK
- A CONSOLE TERMINAL
- ANY XXDP+ SUPPORTED LOAD MEDIA
- ONE OF THESE DMP,DMV-11 CONFIGURATIONS:
 

DMV-11 AA	EIA RS232 AND RS423
DMV-11-AB	CCITT AND V.35
DMV-11-AC	INTEGRAL MODEM
DMP-11-AA	EIA RS232 AND RS423 WITH H3251 TURNAROUND
DMP-11 AB	CCITT AND V.35
DMP 11-AC	INTEGRAL MODEM
DMP 11 AE	RS422
DMP-11-AD	DMP WITH TURNAROUND CONN (H3254,H3255)

NOTE: OPTIONS AE,AC,AB,AND AA ALSO CONTAIN AD.

## 1.3 RELATED DOCUMENTS AND STANDARDS

- DMP USERS MANUAL EK-DMP11-UG-001
- DMP TECH MANUAL EK-DMP11-TM-001
- DMV USERS MANUAL EK-DMV11-UG-001
- DMV TECH MANUAL EK-DMV11-TM-001
- XXDP+ USER'S MANUAL (CHQUS?.SEQ WHERE ? IS THE REV. LEVEL OF THE MANUAL "C" IS THE CURRENT REV.).

## 1.4 DIAGNOSTIC HIERARCHY 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 DMP,DMV 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 DMP,DMV-11, BUT TO TEST THE COMMUNICATION LINK TO WHICH THEY ARE CONNECTED.

SOME OF THE DIAGNOSTICS THAT COULD BE RUN IF THE DMP,DMV-11 LOOKS BAD:

CZDMT - FUNCTIONAL DIAGNOSTIC FOR DMP,DMV 11

FOR DMP:

CZDMP - 8207 STATIC #1 (PROCESSOR)  
CZDMQ - 8207 STATIC #2 (PROCESSOR)  
CZDMR - 8203 STATIC #1 (LINE UNIT)  
CZDMS - 8203 STATIC #2 (LINE UNIT)

FOR DMV:

CVDMA - MICRO PROCESSOR #1  
CVDMB - MICRO PROCESSOR #2  
CVDMC - LINE UNIT #1  
CVDMD - LINE UNIT #2  
CVDME - LINE UNIT #3

#### 1.5 ASSUMPTIONS - RESTRICTIONS

IT IS ASSUMED THAT THE COMMUNICATIONS DEVICE (DMP,DMV 11) HAS BEEN TESTED USING THE PREREQUISITE 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.

## 2.0 OPERATING INSTRUCTIONS

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

### 2.1 COMMANDS

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

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

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

### 2.2 SWITCHES

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

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

EXAMPLE OF SWITCH USAGE:



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

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

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

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

### 2.3 FLAGS

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

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

LOT LOOP ON TEST  
 EVL EXECUTE EVALUATION (ON DIAGNOSTICS WHICH  
 HAVE EVALUATION SUPPORT)

\*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

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

/FLAGS:LOE:IER:BOE

#### 2.4 HARDWARE QUESTIONS

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

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

```

# UNITS (D) ? 1<CR>

UNIT 0
FULL DUPLEX OPERATION : (L) Y ?
DEVICE CSR ADDRESS : (0) 160170 ?
INTERRUPT VECTOR ADDRESS: (0) 300 ?
INTERRUPT PRIORITY: (0) 5 ?
OPTION TYPE
0=DMP
1=DMV: (0) 0 ?
IS THIS A MULTIPOINT NETWORK: (L) N ?
IS THIS A CONTROL STATION: (L) N ?

```

NOTE: THE QUESTION ABOUT CONTROL STATION IS ONLY ASKED IF YOU ANSWER YES TO THE MULTIPOINT QUESTION. WHEN YOU COMPLETE THE ABOVE SEQUENCE YOU WILL BE AT THE DCLT> COMMAND LEVEL

```

THIS IS DCLT. TYPE "H" OR "?" FOR DETAILS
MODE=ACTIVE/PASS=00001
/NOSTATUS/CHECK/NOECHO/NOMODEM
DCLT> (A) ?

```

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.

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 INITIALIZES LIST TO ONE DEFAULT ITEP MESSAGE
CLEAR	TRANSMITLIST	FILLS TRANSMITLIST (000'S) AND THEN INITIALIZES LIST TO ONE DEFAULT ITEP MESSAGE
SET	EXPECTMSG=TYPE/QUAL	DEFINE A MESSAGE TO BE PUT ON THE EXPECTED LIST
	WHERE: "TYPE" IS:	
	=ONES	
	=ZEROES	
	=1ALT	
	=OALT	
	=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 EXPECT=TRANSMIT MAKES A COPY OF THE TRANSMIT LIST IN THE EXPECT LIST.
- SET TRANSMITMSG=TYPE/QUAL DEFINE A MESSAGE TO BE PUT ON THE TRANSMIT LIST (SEE DESCRIPT FOR SET EXP)
- SHOW EXPECTLIST LISTS THE MESSAGE SIZE AND TYPE FOR THE MESSAGES IN THE EXPECT LIST
- SHOW TRANSMITLIST LISTS THE MESSAGE SIZE AND TYPE FOR THE MESSAGES IN THE TRANSMIT LIST

## 2.5.2 TRIBUTARY COMMANDS

NOTE: THESE COMMANDS ARE VALID ONLY IF IN MULTIPOINT MODE.

TRIB ESTABLISH=N,N,N/W

ADDS THE DECIMAL TRIBUTARY  
ADDRESSES SPECIFIED IN N TO  
THE TRIB LIST.

IF /W IS USED THEN PROGRAM  
WILL ASK USER FOR POLL PARAMS  
FOR ALL TRIBS THAT HAVE THE /W  
SWITCH APPENDED. AFTER ALL  
TRIB PARAM QUESTIONS HAVE  
BEEN ANSWERED THEN THE PROGRAM  
ASKS THE USER FOR THE GLOBAL  
POLL PARAMS

TRIB KILL=N,N,N OR ALL

REMOVE TRIB ADDRESSES FOR THE  
TRIB LIST IF 'ALL' IS USED ALL  
TRIBS ARE REMOVED.

TRIB SHOW

LISTS ALL TRIBS IN THE TRIB  
ADDRESS LIST.

## 2.5.2 STATISTICAL COMMANDS

HELP

TYPES HELP INFO FOR OPERATOR

?

TYPES HELP INFO FOR OPERATOR

DUMP SSSSSS EEEEE/E

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.

THE "/B" IS OPTIONAL.  
DEFAULT IS PRINT WORDS  
"/B" CAUSES PRINT BYTES

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

## 2.5.3 RUN COMMANDS

COMMAND	DESCRIPTION																								
RUN MODE=MTYPE/QUAL	STARTS DCLT EXECUTING IN THE MODE SPECIFIED																								
<p>NOTE: MODE=ACTIVE IS NOT DEFAULT, A MODE=MTYPE MUST BE TYPED            -- EACH TIME A RUN IS TYPED</p> <p>WHERE THE "MTYPE" IS ANY ONE OF THE FOLLOWING:</p> <table border="0"> <tr> <td>=ACTIVE</td> <td>(FORCES /NOECHO ,NO LOOPING)</td> </tr> <tr> <td>=PASSIVE</td> <td>(FORCES NO LOOPING)</td> </tr> <tr> <td>=RECEIVE</td> <td>(FORCES /NOECHO ,NO LOOPING)</td> </tr> <tr> <td>=LISTEN</td> <td>(FORCES /NOECHO ,NO LOOPING, /NOCHECK)</td> </tr> <tr> <td>=TRANSMIT</td> <td>(FORCES /NOECHO ,NO LOOPING, /NOCHECK)</td> </tr> <tr> <td>=TALK</td> <td>(FORCES /NOECHO ,NO LOOPING, /NOCHECK)</td> </tr> <tr> <td>=DOWNLINELOAD</td> <td>(FORCES /NOECHO ,NO LOOPING, /NOCHECK)</td> </tr> </table> <p>(FORCING NO LOOPING MEANS IT MUST BE SPECIFIED AS A QUALIFIER ANY TIME ITS DESIRED, THERE IS NO DEFAULT)</p> <p>AND OPTIONAL "QUAL" IS ANY COMBINATION OF THE FOLLOWING:</p> <table border="0"> <tr> <td>/CHECK/NOCHECK</td> <td>ENABLES/DISABLES CHECKING OF RECEIVED DATA AGAINST THE EXPECTED DATA</td> </tr> </table> <p>NOTE: IF BOTH MODES IN ACTIVE AND "/NOCHECK" IS USED, END OF PASS IS DEFINED AS RECEIVING THE SAME # OF MESSAGES THAT IS CONTAINED IN THE TX LIST. WITH NO DATA CHECKING, THERE IS NO WAY FOR DCLT TO KNOW HOW MANY MESSAGES IT SHOULD EXPECT TO RECEIVE.</p> <table border="0"> <tr> <td>/STATUS/NOSTATUS</td> <td>ENABLES/DISABLES PRINTING OF PROGRAM STATUS MESSAGES TO THE OPERATOR</td> </tr> <tr> <td>/ECHO/NOECHO</td> <td>ENABLES/DISABLES THE RETRANSMISSION OF THE DATA RECEIVED IN PASSIVE MODE.            NOTE: THIS IS VALID ONLY FOR PASSIVE MODE.            IF THIS SWITCH IS USED THE TRANSMIT LIST WILL HAVE TO BE RE BUILT.</td> </tr> <tr> <td>/MODEM/NOMODEM</td> <td>ENABLES/DISABLES THE REPORTING OF MODEM STATUS INTERRUPT CHANGES.            NOTE: THIS SWITCH CAUSES NO ACTION IN THIS DCLT PROGRAM BUT IT IS INCLUDED BECUASE IT IS USED IN OTHER DCLT PROGRAMS.</td> </tr> <tr> <td>/LOOP=LTYPE</td> <td>SPECIFIES WHICH, IF ANY, TYPE OF MAINTENANCE LOOPBACK IS BEING USED.</td> </tr> </table>		=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)	/CHECK/NOCHECK	ENABLES/DISABLES CHECKING OF RECEIVED DATA AGAINST THE EXPECTED DATA	/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. NOTE: THIS IS VALID ONLY FOR PASSIVE MODE. IF THIS SWITCH IS USED THE TRANSMIT LIST WILL HAVE TO BE RE BUILT.	/MODEM/NOMODEM	ENABLES/DISABLES THE REPORTING OF MODEM STATUS INTERRUPT CHANGES. NOTE: THIS SWITCH CAUSES NO ACTION IN THIS DCLT PROGRAM BUT IT IS INCLUDED BECUASE IT IS USED IN OTHER DCLT PROGRAMS.	/LOOP=LTYPE	SPECIFIES WHICH, IF ANY, TYPE OF MAINTENANCE LOOPBACK IS BEING USED.
=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)																								
/CHECK/NOCHECK	ENABLES/DISABLES CHECKING OF RECEIVED DATA AGAINST THE EXPECTED DATA																								
/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. NOTE: THIS IS VALID ONLY FOR PASSIVE MODE. IF THIS SWITCH IS USED THE TRANSMIT LIST WILL HAVE TO BE RE BUILT.																								
/MODEM/NOMODEM	ENABLES/DISABLES THE REPORTING OF MODEM STATUS INTERRUPT CHANGES. NOTE: THIS SWITCH CAUSES NO ACTION IN THIS DCLT PROGRAM BUT IT IS INCLUDED BECUASE IT IS USED IN OTHER DCLT PROGRAMS.																								
/LOOP=LTYPE	SPECIFIES WHICH, IF ANY, TYPE OF MAINTENANCE LOOPBACK IS BEING USED.																								

(IGNORED IN MODES OTHER THAN ACTIVE)  
MUST BE SPECIFIED EACH TIME ELSE NO  
LOOP IS USED.

"LTYPE" IS:

- INTERNAL TTL    SETS THE L LOOP BIT IN BSEL1 IF DMP  
AND IF DMV ENTERS MAINT LOOP AND  
SETS THE INTERNAL LOOP BIT.
- CABLE            DOES NOT CAUSE ANY BITS TO BE SET OR  
REQUESTS TO BE QUEUED, BUT MAKES FOR  
A NICE BOOKKEEPING FEATURE. "/L-CABLE"  
WILL THEN BE SHOWN WHEN THE COMMAND  
LINE IS TYPED AS A REMINDER OF WHAT  
TYPE OF LOOPING IS BEING ATTEMPTED.  
REMEMBER TO INSTALL ANY CONNECTORS OR  
ENABLE ANY LOOP FEATURES THAT ARE  
NECESSARY TO MAKE CABLE LOOPBACK  
POSSIBLE.

THE FOLLOWING LOOP TYPES ARE NOT SUPPORTED BY THE DMV.  
INCLUDING THESE LOOP TYPES FOR A DMV WILL HAVE NO EFFECT AT ALL.

- LOCAL MODEM  
ALSO CALLED ANALOG-LOOPBACK.  
SETS MM1 AND DSR IN THE MODEM REG  
THIS IS ONLY FOR RS449 MODEMS
- REMOTE MODEM  
ALSO CALLED DIGITAL-LOOPBACK.  
SETS MM2 AND DSR IN THE MODEM REG  
THIS ONLY FOR RS449 MODEMS

/PASS=MN            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"

EXIT

THE EXIT COMMAND RETURNS THE USER TO THE SUPERVISOR DR>  
PROMPT AFTER PRINTING A SUPERVISOR END OF PASS.



2.5.4 PRINT

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

COMMAND	DESCRIPTION
HELP OR ?	PRINTS HELP INFORMATION FOR RPT>
TSS NNN/SW	SHOWS TRIBUTARY STATUS SLOT INFORMATION WHERE NNN IS THE DECIMAL TRIBUTARY ADDR AND SW IS ONE OF THE FOLLOWING SWITCHES
ERROR	INDICATES ONLY ERROR SLOTS ARE TO BE PRINTED
FULL	INDICATES ALL TRIB STATUS SLOTS ARE TO BE PRINTED
OFFSET=NN	INDICATES THE TRIB STATUS SLOT WHOSE OFFSET IS NN IS TO BE PRINTED.
GSS/SW	PRINT THE GLOBAL STATUS INFORMATION SWITCHES ARE THE SAME AS FOR TSS.
LOG	DUMPS THE EVENT LOG
EXIT	EXITS BACK TO THE COMMAND LEVEL THAT YOU ENTERED FROM. (DCLT> OR DR>)

2.5.6 DEFAULTS

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

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

FOR THE RUN COMMAND THE DEFAULTS ARE:

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

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

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

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

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 MODE=ACTIVE<CR>"

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

## 3.0 ERROR INFORMATION

## 3.1 TYPES OF ERROR MESSAGES

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

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

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

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

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

## 3.2 SPECIFIC ERROR MESSAGES

## 3.2.1 COMMAND LINE INTERPRETER ERRORS:

ERROR MESSAGE:	MEANING
?ILL CMD BAD 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.
- ? TRIB CMDS ILLEGAL IN PT-PT MODE? A TRIB COMMAND WAS ISSUED AND THE MODE DEFINED BY THE HARDWARE P-TABLE WAS POINT TO POINT. IF TRIB COMMANDS ARE TO BE USED PROGRAM MUST BE STARTED AGAIN WITH THE MODE SET TO MULTIPOINT
- ? TRANSMIT AND EXPECT LIST MUST BE IDENTICAL FOR LOOP?  
IF RUN COMMAND WITH "/LOOP/CH" IS TYPED TRANSMIT AND EXPECT LISTS MUST BE EQUAL. USE "SE E=T" COMMAND.
- ? TRIB ADDRESS= XXX IS NOT UNIQUE? THE TRIB WHOSE ADDRESS IS XXX IS ALREADY IN THE TRIB LIST
- ? TRIB ADDRESS= XXX NOT FOUND? THE TRIB WHOSE DECIMAL ADDRESS IS XXX WAS NOT FOUND IN THE TRIB LIST WHEN THE TRIB KILL COMMAND WAS EXECUTED
- ? CABLE,LOC,REM LOOP NOT VALID IN "MULTIPT MODE"? A RUN COMMAND WAS ISSUED WITH LOOP= TO CABLE,LOCAL OR REMOTE WHILE THE MODE SET BY THE P-TABLE WAS MULTIPOINT THESE LOOP MODES ARE ONLY VALID FOR POINT TO POINT OPERATION
- ? TRIBS MUST BE ESTABLISHED TO EXECUTE? A RUN COMMAND WAS ISSUED IN MULTIPOINT MODE AND THE TRIB LIST WAS EMPTY. TO USE MULTIPOINT MODE A LEAST ONE TRIB MUST BE ESTABLISHED.
- ? TRIB STATION CANNOT DO LOOP? A RUN COMMAND WAS ISSUED WITH THE LOOP SWITCH AND THE MODE IN P TABLE WAS MULTIPOINT TRIBUTARY. TRIBUTARY

STATIONS CANNOT DO LOOPBACK.  
 ?ONLY ONE TRIB (TRIB ADDR 1) ALLOWED  
 FOR LOOP IN MULTIPoint?  
 A RUN COMMAND WITH LOOP=INTERNAL  
 WAS ISSUED AND THE TRIB LIST DID  
 NOT HAVE ONLY 1 TRIB IN IT. IF IT  
 DID HAVE ONLY 1 TRIB IN IT THE ADDRESS  
 WAS NOT 1.

?TRIB ADDRESS= XXX INVALID?  
 A TRIB COMMAND WAS ISSUED WITH A TRIB  
 ADDRESS NOT IN THE RANGE 1-255

3.2.2 DCLT ERROR MESSAGES:

CLOCK NOT FOUND  
 THIS MEANS THAT NO CLOCK WAS FOUND  
 ON THE SYSTEM THE DIAGNOSTIC WILL  
 STILL RUN BUT NONE OF THE TIME OUT  
 CONDITIONS WILL OCCUR.

BAD CLOCK - PROGRAM WILL HANG ON "TIMEOUT"!!  
 THIS MEANS THAT 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. THE LIMIT  
 IS CURRENTLY 15.

CHAR. COUNT EXCEEDS BUFFER 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 LIMIT IS 512. BYTES.  
 THE MESSAGE WAS TRUNCATED TO COMPLETELY  
 FILL THE BUFFER. NO MORE MESSAGES CAN

TRIB ADDRESS LIST IS EMPTY BE ADDED TO THAT BUFFER.  
THERE ARE NO TRIBS IN THE TRIB LIST WHEN THE THE TRIB SHOW COMMAND WAS EXECUTED.

TRIB ADDRESS LIST FULL - ADDRESS= XXX NOT ADDED  
A TRIB ESTABLISH COMMAND CAUSED THE NUMBER OF TRIBS IN THE LIST TO EXCEED THE MAXIMUM (DMV=12,DMP=32). THIS ERROR MESSAGE IS REPEATED FOR ALL TRIBS IN EXCESS FOR THIS S'IRING. XXX= THE DECIMAL ADDRESS OF THE TRIB

RX BUFFER NOT BIG ENOUGH  
TOO MANY TRIBS OR MSGS

A RUN COMMAND WAS ISSUED WITH DATA CHECKING REQUESTED AND THE NUMBER OF TRIBS TIMES THE NUMBER OF EXPECTED MESSAGES EXCEEDED THE MAXIMUM REC BUFFER TOTAL (2048 BYTES) TO CORRECT FOR THIS EITHER THE NUMBER OF MESSAGES, THE SIZE OF THE MESSAGE OR THE NUMBER OF TRIBS MUST BE DECREASED.

### 3.2.3 DEVICE ERROR MESSAGES:

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

RECVG=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 RECEIVED WAS A DIFFERENT LENGTH THEN THE MESSAGE THAT WAS EXPECTED.

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

DEVICE DID NOT RETURN RUN BIT  
 SEL0 SEL2  
 XXXXXX XXXXXX  
 ;THIS ERROR INDICATES  
 ;THAT THE DEVICE DID  
 ;NOT RETURN THE RUN BIT  
 ;AFTER 1000 TICKS OF THE CLOCK  
 ;COULD INDICATE MICRO DIAG  
 ;FOUND A FAILURE.

FAILURE IN MICRO DIAGNOSTICS  
 SEL0 SEL6  
 XXXXXX XXXXXX  
 ;THIS ERROR INDICATES THAT  
 ;BSEL6 DOES NOT CONTAIN 305  
 ;THIS IS CHECKED AFTER A MASTER  
 ;CLEAR AND THE RUN BIT HAS  
 ;BEEN SET

TIME OUT WAITING FOR TX OR RX TO COMPLETE  
 SEL0 SEL2  
 XXXXXX XXXXXX  
 ;THIS ERROR IS THE MOST POPULAR  
 ;IT INDICATES THAT THE 60 SEC  
 ;TIMER EXPIRED WHEN THE DEVICE  
 ;WAS EXPECTING TO GET A RX OR  
 ;TRANSMIT COMPLETE. AFTER THIS  
 ;ERROR OCCURS THE PROGRAM WILL  
 ;RESET THE TIMER AND LOOP AGAIN

TIME OUT WAITING FOR RDI  
 SEL0 SEL2  
 XXXXXX XXXXXX  
 ;THIS ERROR INDICATES THAT THE  
 ;DEVICE DID NOT RETURN RDI IN  
 ;RESPONSE TO AN RDI BEFORE THE  
 ;TIMER EXPIRED. THE TIMER IS  
 ; 100 TICKS FOR DMP AND 400  
 ; TICKS FOR THE DMV.



CONTROL OR INFORMATION OUT ERROR  
 SEL2 SEL6  
 XXXXXX XXXXXX YYYYYY

; THIS ERROR INDICATES THAT  
 ; A CONTROL OUT ERROR OCCURRED  
 ; OR AN UNEXPECTED INFORMATION  
 ; OUT OCCURRED. THE TYPE OF  
 ; ERROR IS INDICATED  
 ; BY THE ASCII  
 ; STRING YYYYYY WHICH CAN BE ONE  
 ; FROM THE LIST BELOW.  
 ; SOME CONTROL OUTS ARE FATAL  
 ; IF A FATAL ERROR OCCURS THE  
 ; PROGRAM WILL BE FORCED TO THE  
 ; DCLT> PROMPT THE FATAL ERRORS  
 ; ARE INDICATED BELOW

MSG	FATAL	DESCRIPTION
SELECT THRESHOLD	NO	SELECTION TIMER TIMED OUT MORE THAN 7 TIMES
START RXD IN RUN	YES	DDCMP START RX'D WHILE DEVICE WAS IN RUN STATE
MAINT RXD IN RUN	YES	DDCMP MAINT MESSAGE WAS RX'D WHILE DEVICE WAS IN THE RUN STATE
MAINT RXD IN HALT	YES	DDCMP MAINTINANCE MSG RX'D WHEN DEVICE WAS IN HALT STATE.
START RXD IN MAINT	YES	DDCMP START MSG RX'D WHILE DEVICE WAS IN MAINTINANCE MODE
RING DETECTED	NO	RING SIGNAL WAS SET BY MODEM. THIS OUTPUT FOR DMP ONLY.
DEAD TRIB	NO	INDICATES THAT A TRIB NO LONGER RESPONDS WHEN IT IS POLLED
RUN STATE ERR	NO	RUN STATE OUTPUT IS POSTED WHEN DCLT IS NOT EXPECTING IT.
BABBLING TRIB	YES	A TRIBUTARY IS HOGGING THE LINE AND NOT RETURNING THE SELECT FLAG.
STREAMING TRIB	YES	A TRIBUTARY IS SENDING DATA CONSTANTLY.
BUFFER TOO SMALL	YES	MSG WAS RX'D AND THE DEVICE HAS NO BUFFER

		BIG ENOUGH FOR IT. THIS IS PROBABLY OPER RATER ERROR.
NON EXIST MEM	YES	INDICATES THAT DEVICE TRIED TO NPR TO A MEM LOCATION THAT IS NON EXISTENT.
DISCONNECT	YES	INDICATES DEVICE SAW MODEM READY GO AWAY AFTER BEING SET. LOOK FOR CABLE OR MODEM
QUEUE OVER	YES	DEVICE HAS TOO MUCH OUTPUT OR PROGRAM GAVE DEVICE TOO MUCH.
CARRIER LOSS	YES	INDICATES CARRIER SIG WENT AWAY WHILE RX INC

\*\*NOTE THE FOLLOWING ARE PROCEDURE ERRORS IF THEY OCCUR  
 \*\*THE DEVICE IS PROBABLY BAD ALL PROCEDURE ERRORS ARE FATAL

NO MODE DEF	YES	PROCEDURE ERROR
ILLEGAL TYPE CODE		
MODE CHANGE		
CONTROL IN TO UNES. TRIB		
COMMAND TO TRIB 0		
COMMAND TO UNHALTED TRIB		
MAX TRIBS EXCEEDED		
ESTB TO ALREADY ESTABLISHED		
ILLEGAL REQUEST KEY		
ASSIGN BUFF UNEST. TRIB		
ASSIGN BUFF HALTD TRIB		
ASSIGN BUFF BYTE CNT 0		
ASSIGN TX BUFF TRIB 0		
R OR W RESERVED TSS		
USE RESERVED BIT IN BSEL7		
COMMON POOL ERROR		
QUOTA OVERFLOW		

\*\*\*\* END OF PROCEDURE ERRORS\*\*\*\*\*



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" ARE LOGGED IN IN THE EVENT LOG WHEN EACH "DCLT PASS" IS 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 45 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 PRINT COMMAND MUST BE FOLLOWED BY A LOG COMMAND. THE EVENTS ARE PRINTED IN A "LAST-IN FIRST-OUT" ORDER.

EVENT TIME IS TYPED OUT AS MMM:SS:TT (LIKE 254:36:07) WHERE MMM,SS,TT REPRESENT THE NUMBER OF MINUTES, SECONDS, CLOCK TICKS SINCE THE LAST START OR RESTART. IT SHOULD BE NOTED THAT THE TIMES ARE RELATIVE SINCE WHILE THE PROCESSOR IS RUNNING AT PRIORITY 7 THE CLOCK CAN'T INTERRUPT TO KEEP TIME. THIS IS THE CASE WHILE THE PROGRAM IS FETCHING DCLT COMMANDS FROM THE OPERATOR. IT SHOULD ALSO BE NOTED THAT THERE ARE ONLY 8 BITS AVAILABLE 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" "LOG" 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 TRIBUTARY TO/FROM  
ADDRESS OF 1ST BYTE OF MESSAGE,  
TOTAL NO. OF BYTES, MODEM STATUS AT THAT TIME.

TRANSMIT MESSAGE COMPLETED:

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

RECEIVE SPACE QUEUED:

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

RECEIVE MESSAGE COMPLETED:

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

DATA COMPARISON STARTED:

EVENT TIME, ADDRESS OF TRIBUTARY TO/FROM  
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 TRIBUTARY TO/FROM  
 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 TRIBUTARY TO/FROM  
 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,  
 # OF RX THRESHOLD ERRORS, # OF TX THRESHOLD ERRORS

NOTE RX THRESHOLDS AND TX THRESHOLDS OCCUR IF  
 ONE STATION IS STARTED BEFORE THE OTHER  
 OR IF LINKS ARE RUN AT HIGH SPEED

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
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

## 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] DEVICE PARAMS BIT1      BIT0
                   ;           IF A ZERO   TRIB   POINT-POINT
                   ;           IF A ONE   CONTROL MULTIPPOINT
.WORD 0           ;[12] OPTION TYPE 0=DMP 1=DMV
                   ;
```

## 6.0 MODE AND MESSAGE DESCRIPTIONS

THE FOLLOWING ABBREVIATIONS WILL BE USED IN THE MODE DESCRIPTIONS  
MTP/TB - MULTIPOINT TRIBUTARY  
MTP/CS - MULTIPOINT CONTROL STATION  
PTP - POINT TO POINT

## 6.1 MODE DESCRIPTIONS

## 6.1.1 TRANSMIT MODE

-----  
IF PTP OR MTP/TB:

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

IF MTP/CS: THE LIST IS SENT TO EACH TRIBUTARY

## 6.1.2 RECEIVE MODE

-----  
IF PTP OR MTP/TB:

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.

IF MTP/CS: SPACE IS QUED FOR ALL TRIBUTARIES

## 6.1.3 PASSIVE MODE

IF PTP OR MTP/TB:

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.

IF MTP/CS: A MESSAGE IS RECEIVED FROM EACH TRIB AND THEN A MESSAGE IS TRANSMITTED TO EACH TRIB.

## 6.1.4 ACTIVE MODE

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

IF MTP/TB: THE TRANSMIT MESSAGES OF ALL TRIBS MUST BE IDENTICAL IF DATA CHECKING IS ENABLED.

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

## 6.1.5 DOWN LINE-LOAD

\*\*\*\*\*  
 \* NOTE \* - THE SATELLITE IN MTP MODE WILL ALWAYS BE THE FIRST  
 \* \*\*\*\*\*  
 TRIB IN THE TRIB LIST.  
 IF IN PTP MODE, THE SATELLITE WILL ENTER MOP MODE  
 ONLY IF THE PASSWORD SUPPLIED BY THE USER MATCHES  
 THAT SET IN ITS PASSWORD SWITCH PACK.

IF PTP OR MTP/CS:

THE "HOST" REQUESTS THE "SATELLITE" 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 MICRO-CODE INTO MAIN MEMORY STARTING AT LOC. 0. THIS IMAGE DATA WILL CONTAIN CODE THAT PRINTS A MESSAGE STATING 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.

IF MTP/TB:

RUNNING DOWN LINE LOAD MODE IN A MULTIPOINT TRIB JUSTS ENABLES PRIMARY MOP MODE. TRIBS CANNOT BE "HOSTS"

\*\*\*\*\*  
 \* NOTE \* THE SATELLITE MUST HAVE CERTAIN SWITCHES SET ON

\*\*\*\*\* THE LINE UNIT CARD IN ORDER TO ALLOW THE BOOT TO OCCUR, THE MODE ENABLE SWITCH [SW 8 (OF E121)] MUST BE SET TO A 1(OFF). THE MODE MUST BE DEFINED IN THE SWITCHES(SW'S 5 6 AND 7 OF E 121). THE PASSWORD OR TRIB ADDRESS MUST BE SET IN THE SWITCHES(SW'S IN E-134). THIS MUST BE DONE FOR ALL TYPES OF DOWN LINE LOAD, IN ADDITION THE FOLLOWING MUST BE DONE FOR:  
 REMOTE LOAD DETECT:  
 SWITCH 9 OF E-121 TO A ONE (OFF)  
 FOR POWER ON BOOT AND ENTER P MOP  
 SWITCH 10 OF E-121 TO A ZERO (ON)

INCLUDED IN THE "SECONDARY BOOT MESSAGE" IS THE DEVICE TYPE CODE THAT IS DECIPHERED AND INCLUDED IN AN IDENTIFICATION MESSAGE.

EXAMPLE:

SECONDARY BOOT REQ FROM XXX DEVICE TYPE = YY

YY	XXX
--	---
0	DP
2	DU
4	DL
6	DQ
8	DA
10	DUP
12	DMC
14	DN
16	DLV
18	DMP
20	DTE
22	DV
24	DZ
28	KDP
30	KDZ
32	KL
34	DMV

#### 6.1.6 TALK AND LISTEN MODE

-----

\*\*\*\*\*  
 \* NOTE \* - IN MTP MODE TALK AND LISTEN USE ONLY THE FIRST TRIB  
 \*\*\*\*\* IN THE TRIB LIST

##### 6.1.6.1 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

#### 6.1.6.2 LISTEN MODE

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

#### 6.1.7 MAINTENANCE "LOOP" MODES

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

INTERNALTTL                      LOOPS DATA INTERNALLY THIS WILL NOT WORK FOR MTP/TB. IF MTP/CS THEN TRIB 1 MUST BE ESTABLISHED.

THE FOLLOWING ARE ONLY VALID IN PTP MODE.

CABLE                              DOES NOT CAUSE ANY BITS TO BE SET OR REQUESTS TO BE QUEUED, BUT MAKES FOR A NICE BOOKKEEPING FEATURE. "/L=CABLE" WILL THEN BE SHOWN WHEN THE COMMAND LINE IS TYPED AS A REMINDER OF WHAT TYPE OF LOOPING IS BEING ATTEMPTED. REMEMBER TO INSTALL ANY CONNECTORS OR ENABLE ANY LOOP FEATURES THAT ARE NECESSARY TO MAKE CABLE LOOPBACK POSSIBLE.

LOCALMODEM                      SETS MM1 ON INTERFACE ALSO CALLED ANALOG-LOOPBACK.

REMOTEMODEM                      SETS MM2 ON RS449 INTERFACE ALSO CALLED DIGITAL LOOPBACK.

#### 6.1.8 MODE SUMMARY TABLE

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

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

\* = MOST LIKELY TO BE IN THAT MODE

6.2 MESSAGE DESCRIPTIONS

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

## 7.0 OTHER INFORMATION

## 7.1 INTERFACING TO AN "ITEP" NODE

THIS DCLT WILL INTERFACE ONLY TO THE ITEP FOR DMC.  
IF THIS LINK IS NEEDED THEN THE DMP/V-11 MUST BE IN POINT  
TO POINT MODE AND THE FOLLOWING TABLE APPLIES TO THE ITEP  
NODE:

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 INCORPORATED PATCH FROM DEPO# MD 11 DZDMO A1

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

## 7.2 TROUBLESHOOTING HINTS

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

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

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

## 7.2.1 INTERNAL LOOP AT EACH NODE

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

INTERNAL LOOP WORKS ONLY FOR POINT TO POINT OR MULTIPPOINT CONTROL STATIONS. THE SEQUENCE BELOW IS FOR POINT TO POINT IF YOU WISH TO DO MULTIPPOINT ADD THE COMMAND WITH THE \*

```

C E
C T
SE T=ONES/S=20/C=2
* T E=1
R M=A/LO=I/NOCH/STAT

```

WHAT THE ABOVE COMMAND SEQUENCE MEANS:

THE "C E" AND THE "C T" INITIALIZES THE "EXPECT" LIST AND THE "TRANSMIT LIST". THE "SE T=ONES/S=20/C=2" SETS THE TRANSMIT LIST TO CONTAIN 3 MESSAGES. THE MESSAGES CONTAIN DATA OF ALL ONES AND EACH ONE IS 20 BYTES IN LENGTH. THE "T E=1"(ONLY FOR MTP) ESTABLISHES ONE TRIB ,TRIB ADDRESS 1. THE "R M=A/LO=I/NOCH/STAT" SETS THE MODE TO RUN IN TO BE ACTIVE AND LOOP TYPE TO BE INTERNAL TTL. THE PROGRAM WILL NOT BE CHECKING DATA SO THERE WAS NO NEED TO SET UP AN EXPECT LIST. THE PROGRAM WILL BE PRINTING STATUS MESSAGES.

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

```

INI RXQ TXQ RXQ TXC TXQ RXQ TXQ
RXQ TXC EOP
MODE=ACTIVE/LOOP=INTERNAL/PASS=00000

```

/STATUS/NOCHECK/NOECHO/NOMODEM  
DCLT> (A) ?

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

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

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

WHAT THIS SEQUENCE MEANS:

THIS SEQUENCE IS SIMILAR TO THE ONE ABOVE . THE 'SE E=T' MAKES A COPY OF THE TRANSMIT LIST IN THE EXPECT LIST. THE EXPECT LIST NOW CONTAINS 3 MESSAGES (SAME AS TRANSMIT). THE MESSAGES WILL HAVE ALL ONES FOR DATA AND BE 20 BYTES EACH IN LENGTH. THE RUN COMMAND IS THE SAME WITH THE ADDITION OF TWO SWITCHES "/CH/PAS=3". THE "CH" SWITCH TELLS THE PROGRAM TO CHECK THE RECEIVED DATA AGAINST THE "EXPECTED LIST". THE "PAS=3" SWITCH TELLS THE PROGRAM TO RUN 3 PASSES BEFORE RETURNING TO THE DCLT> PROMPT.

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

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

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. THIS COMMAND IS VALID FOR POINT TO POINT STATIONS ONLY. POSSIBLE COMMAND SEQUENCE IS:

R M=A/CH/PAS=3

WHAT THIS SEQUENCE MEANS:

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

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

```
RXQ TXQ TXC RXQ TXQ TXC RXQ TXQ
```

```

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

```

7.2.2 TRANSMIT ON ONE NODE RECEIVE ON THE OTHER

NOW TRY 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:

```

*****
* NOTE *   THESE SEQUENCES ARE FOR POINT TO POINT MODE
*****   IF YOU WISH TO RUN MULTIPPOINT ADD THE COMMAND
          COMMAND LINES MARKED WITH AN *.

```

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

WHAT THIS SEQUENCE MEANS:

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

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

FOR NODE A:

```

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

```

FOR NODE B:

```

INI RXQ EOP RXQ EOP RXQ EOP
      MODE=RECEIVE/PASS=00000

```

/STATUS/NOCHECK/NOECHO/NOMODEM  
DCLT> (A) ?

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

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

WHAT THIS SEQUENCE MEANS:

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

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

NODE A: IS THE SAME AS ABOVE.

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

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

7.2.3

ONE NODE ACTIVE THE OTHER NODE PASSIVE

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

\*\*\*\*\*  
\* NOTE \* THESE SEQUENCES ARE FOR POINT TO POINT MODE  
\*\*\*\*\* IF YOU WISH TO RUN MULTIPPOINT ADD THE COMMAND  
COMMAND LINES MARKED WITH AN \*.

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
* T E=1	T E=1
R M=ACT/NOCH/PAS=3	R M=P/NOCH/PAS=3

WHAT THIS SEQUENCE MEANS:

THE EXECUTION OF THIS SEQUENCE CAUSES THE FOLLOWING THINGS TO HAPPEN ON NODE A. THE TRANSMIT AND EXPECT LISTS ARE INITIALIZED THEN THE TRANSMIT LIST IS SET TO 3 MESSAGES OF 10 BYTES EACH. THE DATA USED IN THE TRANSMIT MESSAGES IS THE CCITT PATTERN. THEN A IF THIS IS A MULTIPOINT NETWORK A TRIB IS ESTABLISHED (TRIB ADDR. 1) THEN NODE A IS RUN IN ACTIVE MODE WITH DATA CHECKING DISABLED AND THE PASS COUNT SET TO THREE. NOTE STATUS WOULD STILL BE PRINTED IF THE PREVIOUS SEQUENCES HAD BEEN RUN, IF YOU ARE RUNNING FROM LOAD TIME YOU WOULD HAVE TO ADD A "/STA TO THE RUN COMMAND LINE. NODE B: THE TRANSMIT AND EXPECT LISTS ARE INITIALIZED THEN THE TRANSMIT LIST IS SET TO 3 MESSAGES OF 20 BYTES EACH. THE DATA FOR EACH MESSAGE IS ALTERNATING 1'S AND 0'S. IF MULTIPPOINT ESTABLISH 1 TRIB. THEN RUN IN PASSIVE MODE WITH DATA CHECKING DISABLED AND THE PASS COUNT SET TO 3.

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

FOR NODE A:

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

FOR NODE B:

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

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

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

#### 7.2.4 BOTH NODES ACTIVE

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



NOTE - THIS IS FOR POINT TO POINT ONLY

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

WHAT THIS SEQUENCE MEANS:

NODE A SETS UP IS TRANSMIT LIST TO BE 3 MESSAGES. MESSAGE 1 IS 10 BYTES LONG AND CONTAINS DATA OF ALTERNATING 0'S AND 1'S. MESSAGE 2 IS 20 BYTES LONG AND CONTAINS DATA OF THE CCITT PATTERN. MESSAGE THREE IS 30 BYTES LONG AND CONTAINS ALPHANUMERIC DATA. THE EXPECT LIST ALSO CONTAINS 3 MESSAGES. MESSAGE 1 IS 11 BYTES LONG AND CONTAINS 0'S FOR DATA. MESSAGE TWO IS 21 BYTES LONG AND CONTAINS 1'S FOR DATA. MESSAGE 3 IS 31 BYTES LONG AND CONTAINS THE ITEP DATA. NODE B HAS THE SAME MESSAGES EXCEPT THAT THE TRANSMIT MESSAGE LIST IS THE EXPECT MESSAGE LIST AND VICE VERSA. BOTH NODES ARE RUN IN THE ACTIVE MODE WITH NO DATA CHECKING AND PASS COUNT EQUAL TO THREE.

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

ON BOTH NODES A AND B:

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

DCLT> (A) ?

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

#### 7.2.5 TALK AND LISTEN MODES FOR COMMUNICATING

TALK AND LISTEN MODES ARE USEFUL IF THE OPERATORS WISH TO COMMUNICATE WITH EACH OTHER. JUST SETUP A TIME THAT EACH WILL GO TO THEIR MODE, TALK OR LISTEN, AND SEND MESSAGES OVER THE LINK. POSSIBLE COMMAND SEQUENCES ARE. WHEN USING TALK AND LISTEN MODES ON MULTIPOINT LINKS REMEMBER THAT YOU CAN ONLY USE THESE MODES FROM THE CONTROL STATION TO THE FIRST TRIBUTARY IN THE TRIB LIST.

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

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

## 7.3 EXAMPLES OF COMMANDS

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

## 7.3.1 EXAMPLES OF MESSAGES COMMANDS

THE CLEAR COMMANDS .

C E  
C T

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

IF THESE COMMANDS ARE FOLLOWED BY A SHOW COMMAND

SH E

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

MSG: TYPE=ITEP/SIZE=58  
MODE=ACTIVE/PASS=00001  
/NOSTATUS/CHECK/NOECHO/NOMODEM

DCLT> (A) ?

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

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

AND FOLLOWED IT WITH A SHOW EXPECT LIST COMMAND

SH E

WHAT YOU WOULD SEE IS

MSG: TYPE=ALPHA/SIZE=35  
MSG: TYPE=ALPHA/SIZE=35  
MSG: TYPE=ALPHA/SIZE=35  
MSG: TYPE=ALPHA/SIZE=35  
MODE=ACTIVE/PASS=00001  
/NOSTATUS/CHECK/NOECHO/NOMODEM

DCLT> (A) ?

## 7.3.1 EXAMPLES TRIBUTARY COMMANDS

WHEN YOU FIRST GET TO THE DCLT> COMMAND LEVEL IN MULTIPOINT MODE AND YOU EXECUTE A TRIB SHOW COMMAND:

T S

WHAT YOU WOULD SEE IS

TRIB ADDRESS LIST IS EMPTY

THEN YOU COULD TO A TRIB ESTABLISH COMMAND

T E=1,2,3,4

THIS WOULD ESTABLISH TRIB ADDRS 1 2 3 AND 4

IF YOU FOLLOWED THIS WITH A TRIB SHOW COMMAND YOU WOULD SEE

TRIB ADDRESS LIST:

1, 2, 3, 4,

IF YOU THEN DID A TRIB KILL COMMAND

T K=3

FOLLOWED BY A TRIB SHOW.

T S

WHAT YOU WOULD SEE IS  
 TRIB ADDRESS LIST:  
 1. 2. 4.  
 IF YOU FOLLOWED THIS WITH A TRIB KILL ALL COMMAND  
 T K=A  
 AND ANOTHER TRIB SHOW  
 T S  
 WHAT YOU WOULD SEE IS  
 TRIB ADDRESS LIST IS EMPTY  
 IS YOU DID A TRIB ESTABLISH WITH A /W SWITCH  
 T E=1/W,2/W  
 WHAT YOU WOULD SEE IS SHOWN BELOW WHEN YOU GET TO THE ?  
 TYPE EITHER THE NEW PARAMATER OR CARRIAGE RETURN FOR  
 DEFALUT.  
 PARAMETERS FOR TRIB 001  
 000000 PRESET VALUE FOR TX DELAY TIMER  
 NEW POLL PARAMETERS (WORD)= (0) 0 ?  
 377 Q VAL FOR ACT  
 000 R VAL FOR ACT  
 NEW POLL PARAMETERS (BYTE LOW) = (0) 377 ?  
 NEW POLL PARAMETERS (BYTE HI) = (0) 0 ?  
 000 Q VAL FOR INACT  
 100 R VAL FOR INACT  
 NEW POLL PARAMETERS (BYTE LOW) = (0) 0 ?  
 NEW POLL PARAMETERS (BYTE HI) = (0) 100 ?  
 000 Q VAL FOR UNRSP  
 020 R VAL FOR UNRSP  
 NEW POLL PARAMETERS (BYTE LOW) = (0) 0 ?  
 NEW POLL PARAMETERS (BYTE HI) = (0) 20 ?  
 010 NDM TO INACT  
 002 \* T-0 TO UNRSP  
 NEW POLL PARAMETERS (BYTE LOW) = (0) 10 ?  
 NEW POLL PARAMETERS (BYTE HI) = (0) 2 ?  
 010 \*T-0 TO DEAD  
 004 MAX MSG COUNT  
 NEW POLL PARAMETERS (BYTE LOW) = (0) 10 ?  
 NEW POLL PARAMETERS (BYTE HI) = (0) 4 ?  
 005670 SELECTION INTERVAL TIMING COUNT  
 NEW POLL PARAMETERS (WORD)= (0) 5670 ?  
 013650 BABBLING TRIB TIMING COUNT  
 NEW POLL PARAMETERS (WORD)= (0) 13650 ?  
 PARAMETERS FOR TRIB 002  
 000000 PRESET VALUE FOR TX DELAY TIMER  
 .  
 .  
 .  
 .  
 THE SAME AS FOR TRIB 1

013650 BABBLING TRIB TIMING COUNT  
NEW POLL PARAMETERS (WORD)= (0) 13650 ?

GLOBAL POLL PARAMETERS  
0000015 NUM SYNC  
NEW POLL PARAMETERS (WORD)= (0) 15 ?

013650 CARRIER WAIT TIMER COUNTER  
NEW POLL PARAMETERS (WORD)= (0) 13650 ?

000062 DELTA T  
NEW POLL PARAMETERS (WORD)= (0) 13650 ?

000000 DEAD T  
NEW POLL PARAMETERS (WORD)= (0) 13650 ?

000000 POLL DELAY  
NEW POLL PARAMETERS (WORD)= (0) 13650 ?

DCLT> (A) ?

### 7.3.1 EXAMPLES STATISTICAL COMMANDS

IF YOU TYPE A HELP COMMAND

HELP

WHAT YOU WILL SEE IS

DCLT CMDS:

CLEAR OR SHOW EXPECTLIST OR TRANSMITLIST

PRINT OR EXIT

DUMP START-END/B

TRIB SHOW, TRIB ESTABLISH=N/W,N(D)..OR TRIB KILL=N,ALL

WHERE W=INDICATES WRITE POLL PARAMS

SET EXPECTMSG OR TRANSMITMSG=TYPE/SIZE=N OR /COPY=N

SET EXPECT=TRANSMIT

TYPE=ONES,ZEROES,1ALT,0ALT,ITEP,CCITT,ALPHA

OR "OPR SPCD=A Z,SP,TAB,0-9 IN QUOTES"

RUN MODE=MTYP/LOOP=LTYP/CHECK,STATUS,ECHO,MODEM,PASS=N

MTYP=TRAN,REC,ACT,PAS,TAL,LIS,DOWN

LTYP=INT,CAB,LOC,REM/

DCLT> (A) ?

THE SAME WILL HAPPEN IF YOU USE THE ?

THE DUMP COMMAND WORKS LIKE THIS

DUM 41260-41300

THIS WILL DUMP THE DATA FROM ADDRESSES 41260 TO  
41300 IN THE FOLLOWING MANNER

41260 104423 000167 177772 021122 012112 006312 006312 C06312

41300 006312

IF YOU HAD USED THE /B SWITCH

CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 20 9

```

DUM 41260 41300/B
WHAT YOU WOULD SEE IS
41260 023 211 167 000 372 377 122 024
41270 112 024 312 014 312 014 312 014
41300 312

```

### 7.3.1 EXAMPLES RUN COMMANDS

YOU CAN FIND SEVERAL EXPAMLES OF THE RUN COMMAND IN TH TROUBLE SHOOTING HINTS SECTION BUT HERE ARE SOME OTHERS.

IF YOU WERE TO EXECUTE THE RUN COMMAND  
R M=TR/NOST/CH/PAS=4  
WHAT WOULD HAPPEN IS AFTER 4 PASSES THE PROGRAM WOULD RETURN TO THE DCLT PROMT. AND PRINT.  
MODE=TRANSMIT/PASS=00000  
/NOSTATUS/CHECK/NOECHO/NOMODEM

DCLT> (A) ?  
IF YOU WERE TO EXECUTE THE RUN COMMAND  
R M=A/LO=I/ST/CH/PAS=4  
WHAT YOU WOULD SEE (IF USING DEFUALT TRANSMIT AND EXPECT MESSAGES) IS  
INI RXQ TXQ TXC CMP EOP RXQ TXQ  
TXC CMP FOP RXQ TXQ TXC CMP EOP  
MODE=ACTIVE/LOOP=INTERNAL/PASS=0000  
/STATUS/CHECK/NOECHO/NOMODEM

DCLT> (A) ?

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

DR>

### 7.3.1 EXAMPLES PRINT COMMANDS

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

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

HERE ARE SOME EXAMPLES OF RPT> LEVEL COMMANDS

THE HELP OR ? COMMAND  
HELP  
OR  
?  
PRODUCES THE FOLLOWING:  
DCLT REPORT CMDS:



```

000     NAK REASON
000     TRIB ADDR
000000  POLL STATUS FLAGS
000     POLL RATE
000     POLL PRIORITY
000     NA
000     MAX MSG COUNTER
000     COMM POLL QUOTA
000     RX THRESH ERRS
000     TX THRESH. ERRS
000     SELECT THRESH. ERRS
000000  DATA MSGS. TX'MITTD
000000  DATA MSGS. RX'CVD
000000  SELECTION INTERVALS
000     DATA ERRORS OUT
000     MBCC 0 BCC 0 REP 0
000     DATA ERRORS IN
000     MBCC 0 BCC 0 REP 0
000     LOCAL BUFFER ERRS
000     TU 0 TS 0
000     REMOTE BUFFER ERRS
000     TU 0 TS 0
000     SELECTION T-0
000     NRTS 0 IRTS 0
000     LOCAL REPLY T-0
000     REMOTE REPLY T-0
000     HIGHEST MSG # TX'D
000     HIGHEST MSG # ACK'D
000     NEXT MSG # TO TX
000     TPTR ADDR OF LKNBK
000     LAST MSG # TX'D
000     XPTR ADDR OF LNKBK
000     CTL X REPLY T-0
000     STRT OF TX BUFF Q
000     END OF TX BUFF Q
000     HIGHEST MSG # RX'D
000     STRT OF RX BUFFQ
000     END OF RX BUFF Q
000000  TX DELAY TIMER
000     NO DATA MSG COUNTER
000     T-0 COUNTER
000000  PRESET VALUE FOR TX DELAY TIMER
000     Q VAL FOR ACT
000     R VAL FOR ACT
000     Q VAL FOR INACT
000     R VAL FOR INACT
000     Q VAL FOR UNPSP
000     R VAL FOR UNRSP
000     NDM TO INACT
000     # T-0 TO UNRSP
000     # T-0 TO DEAD
000     MAX MSG COUNT
000000  SELECTION INTERVAL TIMING COUNT
000000  BABBLING TRIB TIMING COUNT

```

TO GET A SPECIFIC OFFSET LOCATION FROM THE  
TSS USE THE COMMAND

T 1/0=4  
 THIS IS FOR THE VALUES AT OFFSET 4 BUT YOU COULD  
 USE ANY VALUE FROM 0 TO 36 OCTAL  
 THIS WILL PRODUCE:

000 MAX MSG COUNTER  
 000 COMM POLL QUOTA

TO GET THE GLOBAL ERROR COUNTERS USE  
 THE COMMAND

G  
 TO PRODUCE

TO GET THE FULL GSS USE THE COMMAND

G/F  
 TO PRODUCE:

000 POLPTR  
 000 RCVPTR  
 000 XMTPTR  
 000 TSP  
 000 NASP  
 000 BUFPTR  
 000 S-OF  
 000 E-OF  
 000 S-OQ  
 000 E-OQ  
 000 S-OC  
 000 E-OC  
 000 TIMER STATUS  
 000 S-R TIMER [L]  
 000 S-R  
 000 B-CW TIME [H]  
 000 RPM CNTR  
 000000 AACTIM  
 000 MODEM  
 000 MODE  
 000 ALT SW  
 000 XMTQRT  
 000000 RTNADD  
 000 REMOTE STA ERRS  
 OVRN 0 MMFE 0 SEL 0 STR 0  
 000 LOCAL STA ERRS  
 OVRN 0 MMFE 0 UNDR0 OVR 0  
 000 GBL HDR BCC  
 000 MAINT DATA BCC ERR  
 000 TX HDR 1  
 000 TX HDR 2  
 000 TX HDR 3  
 000 TX HDR 4  
 000 TX HDR 5  
 000 TX HDR 6  
 000 RX HDR 1  
 000 PX HDR 2  
 000 RX HDR 3  
 000 RX HDR 4  
 000 RX HDR 5



```

000    RX HDR 6
000000 R TIMER
000000 D TIMER
000000 POLL DELAY TIMER
000    POLL UPDATE PTR
000    DEAD SCAN
000    CARRIER LOSS TIM
000    USART HANG CTR
000000 NUM SYNC
000000 CARRIER WAIT TIMER COUNTER
000000 DELTA T
000000 DEAD T
000000 POLL DELAY

```

```

*****
* NOTE * - DATA DISPLAYED HERE IS ZEROES ACTUAL DATA WILL VARY
*****

```

TO GET AN OFFSET VALUE USE THE COMMAND

```

G/O=4
TO PRODUCE

```

```

000    E-OF
000    S-OQ

```

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

```

EXIT
WILL RETURN YOU TO THE SUPERVISOR
DR>

```

IF YOU ENTERED REPORT FROM THE DCLT LEVEL  
THEN TYPING

```

EXIT
WILL RETURN YOU TO THE DCLT LEVEL
DCLT>

```

## 7.4 THINGS TO WATCH OUT FOR

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

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

IF YOU ARE IN MULTIPOINT MODE AND YOU ARE USING THE DATA CHECK FEATURE ALL TRIBUTARYS MUST USE THE SAME TRANSMIT LIST.

1  
2  
32  
33  
34  
39  
43  
44  
64  
65  
66  
67  
68  
69  
70  
71  
88  
89  
96  
97  
98

.SBTTL PROGRAM HEADER  
BGNMOD

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

POINTER BGNRPT,BGNAU,BGNDU

HEADER CZCLM,C,0,1800.,0,PRI07

002000  
002000 103  
002001 132  
002002 103  
002003 114  
002004 115  
002005 000  
002006 000  
002007 000  
002010  
002010 103  
002011  
002011 060  
002012  
002012 000000  
002014  
002014 003410  
002016  
002016 066710  
002020  
002020 000000  
002022  
002022 002130  
002024  
002024 000000  
002026  
002026 067350  
002030  
002030 000000  
002032  
002032 000000  
002034  
002034 000000  
002036  
002036 000000  
002040

L\$NAME::  
.ASCII /C/  
.ASCII /Z/  
.ASCII /C/  
.ASCII /L/  
.ASCII /M/  
.BYTE 0  
.BYTE 0  
.BYTE 0  
L\$REV::  
.ASCII /C/  
L\$DEPO::  
.ASCII /O/  
L\$UNIT::  
.WORD 0  
L\$TIML::  
.WORD 1800.  
L\$HPCP::  
.WORD L\$HARD  
L\$SPCP::  
.WORD 0  
L\$HPTP::  
.WORD L\$HW  
L\$SPTP::  
.WORD 0  
L\$LADP::  
.WORD L\$LAST  
L\$STA::  
.WORD 0  
L\$CO::  
.WORD 0  
L\$DTYP::  
.WORD 0  
L\$APT::  
.WORD 0  
L\$DTP::

002040 002124  
 002042  
 002042 000340  
 002044  
 002044 000000  
 002046  
 002046 000000  
 002050  
 002050 003  
 002051 003  
 002052  
 002052 000000  
 002054 000000  
 002056  
 002056 000000  
 002060  
 002060 023266  
 002062  
 002062 050736  
 002064  
 002064 000000  
 002066  
 002066 000000  
 002070  
 002070 052056  
 002072  
 002072 052050  
 002074  
 002074 000000  
 002076  
 002076 023304  
 002100  
 002100 104035  
 002102  
 002102 000000  
 002104  
 002104 050752  
 002106  
 002106 051772  
 002110  
 002110 051770  
 002112  
 002112 050744  
 002114  
 002114 000000  
 002116  
 002116 000000  
 002120  
 002120 000000

99

.WORD L\$DISPATCH  
 L\$PRIO:: .WORD  
 .WORD PRI07  
 L\$ENVI:: .WORD  
 .WORD 0  
 L\$EXP1:: .WORD  
 .WORD 0  
 L\$MREV:: .BYTE C\$REVISION  
 .BYTE C\$EDIT  
 L\$EF:: .WORD 0  
 .WORD 0  
 L\$SPC:: .WORD 0  
 .WORD 0  
 L\$DEVP:: .WORD L\$DVTYP  
 .WORD L\$RPT  
 L\$REPP:: .WORD 0  
 .WORD 0  
 L\$EXP4:: .WORD 0  
 .WORD 0  
 L\$EXP5:: .WORD L\$AU  
 .WORD L\$DU  
 L\$AUT:: .WORD 0  
 .WORD L\$DESC  
 L\$DUT:: .WORD E\$LOAD  
 .WORD 0  
 L\$LUN:: .WORD L\$INIT  
 .WORD I\$CLEAN  
 L\$DESP:: .WORD L\$AUTO  
 .WORD L\$PROT  
 L\$LOAD:: EMT  
 .WORD 0  
 L\$ETP:: .WORD  
 .WORD 0  
 L\$ICP:: .WORD  
 .WORD 0  
 L\$CCP:: .WORD  
 .WORD 0  
 L\$ACP:: .WORD  
 .WORD 0  
 L\$PRT:: .WORD  
 .WORD 0  
 L\$TEST:: .WORD  
 .WORD 0  
 L\$DLY:: .WORD  
 .WORD 0  
 L\$HIME:: .WORD

1  
2  
3  
4  
5  
6  
7  
8  
9

.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 052064

.WORD 1  
L\$DISPATCH::  
.WORD T1

```

1          .SBTTL  DEFAULT1 HARDWARE P TABLE
2
3          ;**
4          ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
5          ; THE TEST DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
6          ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P TABLES,
7          ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
8          ;
9
10         BGNHW  DFPTBL
11         002126          .WORD  L10000-L$HW/2
12         002126  000010          DFPTBL::
13         002130
14         002130
15
16         ;INDEPENDENT SECTION
17         ; THE NUMBERS IN BRACKETS ARE THE OFFSET VALUES USED IN THE PARAMETER
18         ; CODING SECTION.
19
20         .WORD  1          ;[0] FULL OR HALF DUPLEX FLAG (BIT0=1 IF FULL)
21
22         ;DEVICE DEPENDENT SECTION
23         ; ADDING OR REMOVING WORDS FROM THIS TABLE EFFECTS THE "GET" CALLS IN
24         ; THE HARDWARE PARAMETER CODING SECTION BY CHANGING "OFFSETS"
25
26         .WORD  160170      ;[2] CSR ADDRESS
27         .WORD  300        ;[4] INTERRUPT VECTOR
28         .WORD  240        ;[6] INTERRUPT PRIORITY (5)
29         .WORD  0          ;[10] DEVICE PARAMETERS WORD
30                          ; BIT0=(1=MULTI 0=POINT TO POINT)
31                          ; BIT1=(1=CONTROL 0=TRIB)
32         .WORD  0          ;[12] DEVICE OPTION TYPE
33                          ; 0=DMP,1=DMV
34         .WORD  4          ;[14]SPARE
35         .WORD  0          ;[16]SPARE
36
37         ENDHW
38
39         L10000:
    
```

1  
2  
19  
42  
43  
44  
45  
55  
56  
57  
58  
59  
60  
61  
76  
77 002150

.SBTTL GLOBAL EQUATES SECTION

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

EQUALS

; BIT DIFINITIONS

100000	BIT15--	100000
040000	BIT14--	40000
020000	BIT13--	20000
010000	BIT12--	10000
004000	BIT11--	4000
002000	BIT10--	2000
001000	BIT09--	1000
000400	BIT08--	400
000200	BIT07--	200
000100	BIT06--	100
000040	BIT05--	40
000020	BIT04--	20
000010	BIT03--	10
000004	BIT02--	4
000002	BIT01--	2
000001	BIT00--	1

001000	BIT9--	BIT09
000400	BIT8--	BIT08
000200	BIT7--	BIT07
000100	BIT6--	BIT06
000040	BIT5--	BIT05
000020	BIT4--	BIT04
000010	BIT3--	BIT03
000004	BIT2--	BIT02
000002	BIT1--	BIT01
000001	BIT0--	BIT00

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

000040	EF.START--	32.	; START COMMAND WAS ISSUED
000037	EF.RESTART--	31.	; RESTART COMMAND WAS ISSUED
000036	EF.CONTINUE--	30.	; CONTINUE COMMAND WAS ISSUED
000035	EF.NEW--	29.	; A NEW PASS HAS BEEN STARTED
000034	EF.PWR--	28.	; A POWER FAIL/POWER UP OCCURRED

0

```

;
; PRIORITY LEVEL DEFINITIONS
;
000340 PRI07-- 340
000300 PRI06-- 300
000240 PRI05-- 240
000200 PRI04-- 200
000140 PRI03-- 140
000100 PRI02-- 100
000040 PRI01-- 40
000000 PRI00-- 0

```

```

;
; OPERATOR FLAG BITS
;
000004 EVL-- 4
000010 LOT-- 10
000020 ADR-- 20
000040 IDU-- 40
000100 ISR-- 100
000200 UAM-- 200
000400 BOE-- 400
001000 PNT-- 1000
002000 PRI-- 2000
004000 IXE-- 4000
010000 IBE-- 10000
020000 IER-- 20000
040000 LOE-- 40000
100000 MOE-- 100000

```

1



```

; INDEPENDENT EQUATES
2
3      001000      BUFLIM=512.      ;MAX BUFFER SIZE IN BYTES
4
5      000017      MSGLIM=15.      ; APPLIES TO TX,RX AND CMP BUFFS
6
7
8
9
10
11     004000      RBFLIM=2048.     ;MAX NUMBER OF BYTES FROM ALL TRIBS
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
;MODE OF OPERATION EQUATES
REC=0      ;RECEIVE MODE
TRA=1      ;TRANSMIT MODE
PAS=2      ;PASSIVE MODE
ACT=3      ;ACTIVE MODE
DOW=4      ;DOWN-LINE-LOAD MODE
TAL=5      ;TALK MODE
LIS=6      ;LISTEN MODE

;MAINT LOOP TYPE EQUATES
NONE= 0    ;NO LOOP
TTL= 1     ;INTERNAL TTL
CABLE= 2   ;CABLE LOOP
MOULOC= 3  ;MODEM LOCAL
MODREM= 4  ;MODEM REMOTE
MOP= 5     ;MOP

;CLOCK ENABLE VALUES TO BE LOADED IN CLK'S CSR
LCLKEN= 100 ;L-CLOCK CSR VALUE TO ENABLE THE CLOCK
PCLKEN= 111 ;P-CLOCK CSR VALUE TO ENABLE THE CLOCK
PCLKCT= 1600 ;P-CLOCK COUNT SET REGISTER FOR COUNTER

;PAPAM WORD EQUATES
STATB= BIT0 ;OPERATOR AWAKE ASKED FOR
DATCKB= BIT1 ;DATA CHECK BIT
ECHOB= BIT2 ;ECHO BIT
MOCHK= BIT3 ;MODEM CHECK/NO CHECK
CRCB= BIT4 ;CRC CALCULATE ASKED FOR
PROTOB= BIT5 ;PROTOCOL PROCESSING ASKED FOR

;OPTION TYPE EQUATES
DMP= 0      ;DMP OPTION
DMV= 1      ;DMV
DMP6= 4     ;DMP 8206
MTP= BIT0   ;MULTIPOINT IF 1 IF PIPT =0
TRBB= BIT1  ;TR'B BIT IF 0-TRIB IF 1-CONTROL

```

```

69      ;EVENT LOG MESSAGE TYPES (USED TO LOCATE EVENT DESCRIPTION IN EVENT TABLE
70      ; AND DISPATCHING TO SEPARATE SECTIONS OF THE EVENT REPORTING SECTION)
71      000000      TXQ= 0      ;TRANSMIT MESSAGE QUEUED
72      000002      TXC= 2      ;TRANSMIT COMPLETE
73      000004      RXQ= 4      ;RECEIVE BUFFER QUEUED
74      000006      RXC= 6      ;RECEIVE COMPLETE
75      000010      DER= 10     ;DEVICE INFORMATION
76      000012      DVI= 12     ;DEVICE ABOUT TO INIT
77      000014      DCK= 14     ;DATA COMPARISON RESULTS
78      000016      MSC= 16     ;MODEM STATUS CHANGE
79      000020      DLE= 20     ;DATA COMPARISON LENGTH ERROR
80      000022      DDE= 22     ;DATA COMPARISON DATA ERROR
81      000024      EOP= 24     ;END OF PASS
82
83      ;EQUATES FOR FLAG WORD
84
85      000001      ININT= BIT0   ;INPUT INT. REC.
86      000002      OTINT= BIT1  ;OUTPUT INT REC
87      000004      ORX=  BIT2   ;RX QUED /COMPL
88      000010      QTX=  BIT3   ;TX QUED/COMPL
89      000100      ERX=  BIT6   ;EXPECT TO GET A RX COMPLETED
90      000200      ETX=  BIT7   ;EXPECT TO GET A TX COMPLETED
91
102
103      000400      RUNST= BIT8   ;INDICATES TRIB COULD GIVE RUN STATE INTERRUPT
104      001000      DLLGA= BIT9  ;INDICATES GO AHEAD FOR DLL.
105      002000      INOVR= BIT10 ;INDICATE DEVICE INITIALIZATION OVER
106
107      ; SPECIAL CLI CODES FOR "CHAR" ARGUMENT IN CLI CALLS
108      ; (COMMAND LINE INTERPRETER DEFINITIONS)
109      000000      CLIERR= 0
110      000001      CLIEXI= 1
111      000002      CLIBR= 2
112      000003      CLIBIF= 3
113      000004      CLISFA= 4
114      000005      CLINUM= 5
115      000006      CLIALP= 6
116      000007      CLIALN= 7
117      000010      CLIOCT= 8.
118      000011      CLIDEC= 9.
119      000012      CLISTR= 10.
120
121      ; DEFS FOR COMMAND LINE INTERPRETATION ACTION VALUES
122      000000      NULL=0
123      000001      CLEAR=1
124      000002      SHOW=2
125      000003      CHECK=3
126      000004      RUN=4
127      000005      HLP=5
128      000006      CSMEXP=6
129      000007      CSMTRN=7
130      000010      SETEXP=10
131      000011      SETTRN=11
132      000012      SIZE=12
133      000013      QCOPY=13
134      000014      NUM=14
135      000015      OPRMSG=15

```

CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 26 2  
 GLOBAL EQUATES SECTION

136	000016	STATUS=16	
137	000017	ENDQ0=17	
138	000020	CMG0=20	
139	000021	CMG1=21	
140	000022	CMG2=22	
141	000023	CMG3=23	
142	000024	CMG4=24	
143	000025	CMG5=25	
144	000026	CMG6=26	
145	000027	ATVMOD=27	
146	000030	PASMOD=30	
147	000031	RECMOD=31	
148	000032	LISMOD=32	
149	000033	DLLMOD=33	
150	000034	TRAMOD=34	
151	000035	TALMOD=35	
152	000036	NO=36	
153	000037	ECHO=37	
154	000040	CRC=40	
155	000041	PROTO=41	
156	000042	PASC=42	
157	000043	MOP=43	
158	000044	TTLLOP=44	
159	000045	CBLLOP=45	
160	000046	LMOLOP=46	
161	000047	RMDLOP=47	
162	000050	NOTNUF=50	
163	000051	BADCHR=51	
164	000052	DMP5=52	
165	000053	DMPE=53	
166	000054	DMPQ=54	
167	000055	PRNT=55	
168	000056	MOSC=56	
169	000057	SLST=57	
170	000060	ETRB=60	
171	000061	KTRB=61	
172	000062	KALL=62	
173	000063	EKTB=63	
174	000064	CTPP=64	
175	000065	ETWS=65	
176	000066	EXIT=66	
177	000067	SETET=67	;REV B EC
178			
179	000001	RPHLP=1	
180	000002	RPEXT=2	
181	000003	RPLOG=3	
182	000004	RPGSS=4	
183	000005	RPTSS=5	
184	000006	RPTSN=6	
185	000007	RPSWE=7	
186	000010	RPSWF=10	
187	000011	RPSWO=11	
188	000012	RNOTNF=12	
189			
190			
202			
203			; DEVICE DEPENDENT EQUATES

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 26 3  
 GLOBAL EQUATES SECTION

```

204      ; MODEM SIGNAL BIT DEFINITIONS
205      ; IF SIGNAL AVAILABLE IN DEVICE, EQUATE NAME TO BIT POSITION,
206      ; ELSE EQUATE IT TO = 0
207      000004      CTS=      BIT2      ;CLEAR TO SEND (CIRCUIT CB)
208      000010      DSR=      BIT3      ;DATA SET READY (CIRCUIT CC)
209      000001      DCD=      BIT0      ;DATA CARRIER DETECT (CIRCUIT CF)
210      000040      RTS=      BIT5      ;REQUEST TO SEND (CIRCUIT CA)
211      000200      RI=       BIT7      ;RING INDICATOR (CIRCUIT CE)
212      040000      SQD=     BIT14     ;SIGNAL QUALITY DETECT (CIRCUIT CG)
213      002000      TM=      BIT10     ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)
214
225
226      ; DEVICE BIT DEFINITIONS
227
228      000200      RQI=      BIT7
229      000200      RDO=      BIT7
230      040000      MCLR=    BIT14
231      000020      RDI=      BIT4
232      000001      IEI=      BIT0
233      000020      IE0=      BIT4
234

```

```

1      .SBTTL GLOBAL DATA SECTION
2      .SBTTL          DEFAULT MESSAGE DEFINITIONS AND TABLES
3
4      ;**
5      ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
6      ; IN MORE THAN ONE TEST.
7      ;
8
9      ;MESSAGE BYTE COUNT TABLE
10
11     DMSGCT:
12     MSG0C: .WORD   EMSG0-MSG0      ;BYTE COUNT OF MESSAGE #0
13     MSG1C: .WORD   EMSG1-MSG1      ;BYTE COUNT OF MESSAGE #1
14     MSG2C: .WORD   EMSG2-MSG2      ;BYTE COUNT OF MESSAGE #2
15     MSG3C: .WORD   EMSG3-MSG3      ;BYTE COUNT OF MESSAGE #3
16     MSG4C: .WORD   EMSG4-MSG4      ;BYTE COUNT OF MESSAGE #4
17     MSG5C: .WORD   EMSG5-MSG5      ;BYTE COUNT OF MESSAGE #5
18     MSG6C: .WORD   EMSG6-MSG6      ;BYTE COUNT OF MESSAGE #6
19     OPCNT: .WORD   0                ;BYTE COUNT FOR OPERATOR SPEC'D MSG.
20     MSG8C: .WORD   EMSG8-MSG8      ;BYTE COUNT OF RECEIVE BUFFER FILL PATTERN
21     DLLM1C: .WORD  DLLM1E-DLLM1    ;DLL MSG 1 COUNT
22     DLLM2C: .WORD  DLLM2E-DLLM2    ;DLL MSG 2 COUNT
23
24
25     ;MESSAGE ADDRESS TABLE
26
27     DMSGAD:
28     MSG0      ;ADDRESS OF MESSAGE #0
29     MSG1      ;ADDRESS OF MESSAGE #1
30     MSG2      ;ADDRESS OF MESSAGE #2
31     MSG3      ;ADDRESS OF MESSAGE #3
32     MSG4      ;ADDRESS OF MESSAGE #4
33     MSG5      ;ADDRESS OF MESSAGE #5
34     MSG6      ;ADDRESS OF MESSAGE #6
35     OPBUF     ;ADDRESS OF OPERATOR SPEC'D MSG.
36     MSG8      ;ADDRESS OF RECEIVE BUFFER FILL PATTERN
37
38     MSG0: .BYTE 000                ;MESSAGE OF ALL 0'S
39     EMSG0:
40     MSG1: .BYTE 377                ;MESSAGE OF ALL 1'S
41     EMSG1:
42     MSG2: .BYTE 252                ;MESSAGE OF ALTERNATING 1'S
43     EMSG2:
44     MSG3: .BYTE 125                ;MESSAGE OF ALTERNATING 0'S
45     EMSG3:
46     MSG4:                                     ;"CCITT" 512-BIT (VS. 511 BITS) TEST PATTERN
47     .WORD 177603,157427,031011,047321,163715,105221,143325,142304
48     .WORD 040041,014116,052606,172334,105025,123754,111337,111523
49     .WORD 030030,145064,137642,143531,063617,135075,066730,026575
50     .WORD 052012,053627,070071,151172,165044,031605,166632,016741
002150 000001
002152 000001
002154 000001
002156 000001
002160 000100
002162 000072
002164 000101
002166 000000
002170 000001
002172 000005
002174 000254
002176 002220
002200 002221
002202 002222
002204 002223
002206 002224
002210 002324
002212 002416
002214 002524
002216 002646
002220 000
002221 377
002222 252
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

```

	002320	166632	016741		
51	002324			EMSG4:	
52	002324			MSG5:	; "INTERPROCESSOR TEST PROGRAM'S (ITEP)" MESSAGE
53					; #1, (DP1:)
54	002324	177	177	044	.ASCII <177><177>/#A THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG./
	002327	101	040	124	
	002332	110	105	040	
	002335	121	125	111	
	002340	103	113	040	
	002343	102	122	117	
	002346	127	116	040	
	002351	106	117	130	
	002354	040	112	125	
	002357	115	120	105	
	002362	104	040	117	
	002365	126	105	122	
	002370	040	124	110	
	002373	105	040	114	
	002376	101	132	131	
	002401	040	104	117	
	002404	107	056		
55	002406	015	012	001	.ASCIZ <15><12><001><177><177><177><177>
	002411	177	177	177	
	002414	177	000		
56	002416				EMSG5:
57	002416				MSG6:
58	002416	043	044	041	.ASCII /#!" &'()*+,-.0123456789:;<=>?@ABCDEFGHIJKLMN OPQRSTUVWXYZ/
	002421	042	040	046	
	002424	047	050	051	
	002427	052	053	054	
	002432	055	056	060	
	002435	061	062	063	
	002440	064	065	066	
	002443	067	070	071	
	002446	072	073	074	
	002451	075	076	077	
	002454	100	101	102	
	002457	103	104	105	
	002462	106	107	110	
	002465	111	112	113	
	002470	114	115	116	
	002473	117	120	121	
	002476	122	123	124	
	002501	125	126	127	
	002504	130	131	132	
59	002507	057	133	134	.ASCIZ ?/[ \ ] ^ _ # ?
	002512	135	136	137	
	002515	045	000		
60	002517				EMSG6:
61					.EVEN
62					
63					; *****
64					; THESE THREE STORAGE AREAS MUST NOT BE SEPARATED !!!!
65					
66	002520	045	116	045	OPBFPT: .ASCII /#N#A/
	002523	101			
67	002524				OPBUF: .BLKB 82. ;BUFFER FOR OPERATOR SPEC'D MESSAGES

```

68 002646          OPEND:
69
70                ; THE ABOVE THREE LINE; MUST BE KEPT TOGETHER
71                ; *****
72
73 002646          MSG8:  .BYTE  33          ;RECEIVE BUFFER FILL PATTERN
74 002647          EMSG8:
75
76                ; DOWN-LINE-LOAD MESSAGE DEFINITIONS
77
78 002647          DLLM1:  .BYTE  6
79 002650          PASS1:  .BYTE  0
80 002651          PASS2:  .BYTE  0
81 002652          PASS3:  .BYTE  0
82 002653          PASS4:  .BYTE  0
83 002654          DLLM1E:
84 002654          DLLM2:  .BYTE  0          ;CODE
85 002655          .BYTE  0          ;LOAD NUMBER
86 002656          .BYTE  6          ;LOAD ADDRESS LSB
87 002657          .BYTE  0
88 002660          .BYTE  0
89 002661          .BYTE  0          ;LOAD ADDRESS
90
91                ; IMAGE DATA
92                ;
93 002662          000240          NOP          ;BYTE COUNT=240 (USED ONLY IN CATS VTC LOADER)
94 002664          005037          000006          CLR          @#6
95 002670          012706          001000          MOV          @1000,SP
96 002674          012701          177560          MOV          @177560,R1          ;SET UP TTY
97 002700          010700          MOV          PC,R0          ;MAKE ADDR.PIC
98 002702          062700          000034          ADD          @<MSG-.>,R0          ;ADDRESS MSG.
99 002706          105761          000004          1$: TSTB          4(R1)          ;TTY READY?
100 002712          100375          BPL          1$          ;WAIT TIL YES
101 002714          112061          000006          MOVB          (R0)+,6(R1)          ;TYPE A CHAR
102 002720          001372          BNE          1$          ;KEEP GOING
103 002722          012737          000026          000024          MOV          @26,@#24          ;SET UP POWER FAIL
104 002730          005037          000026          CLR          @#26          ;MAKE SURE T BIT CLAER
105 002734          000777          BR          ;JUMP ON YOURSELF
106 002736          012          015          102          MSG:  .ASCII  <12><15>/BOOT MESSAGE WAS RECEIVED SUCCESSFULLY  END OF TEST!!!/
      002741          117          117          124
      002744          040          115          105
      002747          123          123          101
      002752          107          105          040
      002755          127          101          123
      002760          040          122          105
      002763          103          105          111
      002766          126          105          104
      002771          040          123          125
      002774          103          103          105
      002777          123          123          106
      003002          125          114          114
      003005          131          040          055
      003010          105          116          104
      003013          040          117          106
      003016          040          124          105
      003021          123          124          041
      003024          041

```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 27 3  
 DEFAULT MESSAGE DEFINITIONS AND TABLES

107	003025	012	015	056	.ASCIZ <12><15>/....RELOAD PROGRAM..../
	003030	056	056	056	
	003033	122	105	114	
	003036	117	101	104	
	003041	040	120	122	
	003044	117	107	122	
	003047	101	115	056	
	003052	056	056	056	
	003055	000			
108					
109					:::PADDING TO OBTAIN 240 BYTES OF DATA
110	003056	177603	157427	031011	.WORD 177603,157427,031011
111	003064	047321	163715	105221	.WORD 047321,163715,105221
112	003072	143325	142304	040041	.WORD 143325,142304,040041
113	003100	014116	052606	172334	.WORD 014116,052606,172334
114	003106	105025	123754	111337	.WORD 105025,123754,111337
115	003114	111523	030030	145064	.WORD 111523,030030,145064
116					
117					:::CRC VALUE FOR ABOVE 240 BYTES OF DATA
118	003122	152645			.WORD 152645 ;CRC
119					
120	003124	006			.BYTE 6
121	003125	000			.BYTE 0
122	003126	000			.BYTE 0
123	003127	000			.BYTE 0
124	003130				DLLM2E:
125					
126					.EVEN
127					



```

1          .COMMAND LINE BUFFER, DATA LOCATIONS AND MESSAGES FOR ACTION ROUTINES
2
3 003130   CMDBUF: .BLKB   82.          ;BUFFER FOR OPERATOR COMMANDS
4 003252   000000   KEYWD1: .WORD   0          ;THIS LOC WILL =1 IF CLEAR TYPED, 2 FOR SHOW,
5                                     ; A 4 IF RUN WAS TYPED, 5 IF HELP WAS TYPED
6 003254   000000   QUALFG: .WORD   0          ;THIS LOC HOLDS QUALIFIER VALUE (SIZE OR COPY)
7 003256   000000   QUALVL: .WORD   0
8 003260   024123   HLPTAB: .WORD   HLP1
9 003262   024136   .WORD   HLP2
10 003264   024254   .WORD   HLP2B
11 003266   024344   .WORD   HLP2C
12 003270   024417   .WORD   HLP3
13 003272   024504   .WORD   HLP3A   ;REV B EC
14 003274   024531   .WORD   HLP4
15 003276   024610   .WORD   HLP4A
16 003300   024666   .WORD   HLP5
17 003302   024756   .WORD   HLP6
18 003304
19 003304   025377   HLPEND:
20 003306   025421   RHLPTB: .WORD   RHLP1
21 003310   025436   .WORD   RHLP2
22 003312   025470   .WORD   RHLP3
23 003314
24
25
26
27 003314   025647   025656   025663   SHTYTB: .WORD   SHTYP0,SHTYP1,SHTYP2,SHTYP3,SHTYP4,SHTYP5,SHTYP6,SHTYP7
   003322   025670   025675   02570*
   003330   025710   025716
28
29          ; THE LIST OF BYTES BELOW ARE THE FIRST BYTES OF THE PREDEFINED MESSAGES
30          ; USED TO "SHOW" THE TRANSMIT AND COMPARE BUFFER CONTENTS.
31
32 003334   000      377      252   SHTAB: .BYTE   0,377,252,125,203,177,043
   003337   125      203      177
   003342   043
33 003343
34          SHTEND:
35          .EVEN
36 003344   026674   MODES:  .WORD   M00      ;ADDRESSES OF MODE TYPES IN ASCII
37 003346   026704   .WORD   M01
38 003350   026715   .WORD   M02
39 003352   026725   .WORD   M03
40 003354   026734   .WORD   M04
41 003356   026751   .WORD   M05
42 003360   026756   .WORD   M06
43
44 003362   026765   LOOPS:  .WORD   LP0      ;ADDRESSES OF LOOP TYPES IN ASCII
45 003364   026775   .WORD   LP1
46 003366   027006   .WORD   LP2
47 003370   027014   .WORD   LP3
48 003372   027027   .WORD   LP4
49
50          ;COMMAND LINE TRAVERSE LOCATIONS (USED BY "P$TRV")
51
52 003374   000000   P$BUFA: .WORD   0          ;LOC. TO HOLD ADDR. OF CMD LINE BUFFER
53 003376   000000   P$TREE:  WORD   0          ;LOC. TO HOLD ADDR. OF PARSING TREE

```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 28 1  
 DEFAULT MESSAGE DEFINITIONS AND TABLES

54	003400	000000	P\$ACT:	.WORD	0	;LOC. TO HOLD ADDR. OF ACTION ROUTINE
55	003402	000000	P\$CNT:	.WORD	0	;LOC. TO BE A COUNTER LOCATION
56	003404	000000	P\$NUM:	.WORD	0	;LOC. TO HOLD NUMERIC VALUE FROM PARSE
57	003406	000000	P\$RADX:	.WORD	0	;LOC. TO HOLD RADIX USED(LO) AND */ (HI BYTE)
58	003410	000	P\$NUF:	.BYTE	0	;RETURN =0 IF ENOUGH OF COMMAND FOUND
59	003411	000	P\$GDBD:	.BYTE	0	;RETURN CODE 0 IF NO ERROR FOUND
60	003412	000	WRFLG:	.BYTE	0	;WRITE FLAG
61				.EVEN		
62	003414	000000	VALTRB:	.WORD	0	;VALID TRIB FLAG..IF SET 1 THEN VALID REV B EC
63						

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 29  
 MESSAGE BUFFERS AND POINTER TABLES

```

1          .SBTTL          MESSAGE BUFFERS AND POINTER TABLES
2
3 003416    TXBUF:  .BLKB    BUFLIM  ;TRANSMITTER BUFFERS
4 004416    CMPBUF: .BLKB    BUFLIM  ;COMPARISON BUFFERS
5 005416    RXBUF:  .BLKB    RBFLIM  ;RECEIVER BUFFERS
6
7 011416    PTRTAB: .BLKW    MSGLIM*2      ;TABLE FOR MESSAGE ADDRS. & BYTE COUNTS
8 011512    PTR13:  .BLKW    MSGLIM*2
9 011606    PTR23:  .BLKW    MSGLIM*2
10 011702   .BLKW    MSGLIM*2*31.      ;TABLE FOR MULTIPOINT POINTERS
11
12 015406    PTREND:          ; END OF MSG. PTR. TABLE
13
14 015406   .BLKW    2          ;FILLER FOR OVERFLOW OF RX POINTER TABLE
15 015412    CPTRLS: .BLKW    32.      ;TABLE FOR MULTIPPOINT RX POINTERS
16 015512    CPTTLS: .BLKW    32.      ;TABLE FOR MULTIPPOINT TX POINTERS
17 015612    DVRCLS: .BLKB    32.      ;TABLE (BYTES) FOR REC COUNTS
18 015652    DVTCLS: .BLKB    32.      ;TABLE (BYTES) FOR TX COUNTS
19 015712    TRIBLS: .BLKB    32.      ;TABLE (BYTES) OF TRIB ADDRESSES
20 015752    177777 .WORD    177777
21 015754    000000 TRBTOT: .WORD    0          ;TOTAL NUMBER OF TRIBS IN LIST
22 015756    000000 TRIBN:  .WORD    0          ;CURRENT TRIB NUMBER
23 015760    000000 INDW:   .WORD    0          ;WORD INDEX
24 015762    000000 INDEX:  .WORD    0          ;BYTE INDEX FOR TRIBS
25 015764    000000 CTX:    .WORD    0          ;COUNTER FOR TX BUFFER COMPLETE INTERRUPTS
26 015766    000000 CRX:    .WORD    0          ;COUNTER FOR RX BUFFER COMPLETE INTERRUPTS
27 015770    000000 ?SPTRS: .WORD    0          ;STACK POINTER FOR RX INTERPUTS ON STACK
28 015772    000000 RSPTRE: .WORD    0          ;STACK POINTER FOR RX INTERPUTS OFF STACK
29 015774    000000 TSPTR: .WORD    0          ;STACK POINTER FOR TX INTERRUPTS
30 015776
31 016012    RXSTAK: .BLKW    6.
32 016166    RXSKEN: .BLKW    54.      ;TX AND RX INT STACKS

```

1  
2  
3 016166 000000  
4 016170 377  
5 016171 000  
6 016172 000  
7 016173 100  
8 016174 000  
9 016175 020  
10 016176 010  
11 016177 002  
12 016200 010  
13 016201 004  
14  
15 016202 005670  
16 016204 013560  
17  
18  
19  
20 016206 000015  
21 016210 013560  
22 016212 000062  
23 016214 023420  
24 016216 000000  
25 016220  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36 016220  
37  
38  
39 017220  
40

; POLL DEFAULTS FOR TRIBS

POLDEF: .WORD 0 ;TX DELAY TIMER  
 .BYTE 377 ;Q FOR ACTIVE  
 .BYTE 0 ;R FOR ACTIVE  
 .BYTE 0 ;Q FOR INACTIVE  
 .BYTE 100 ;P FOR INACTIVE  
 .BYTE 0 ;Q FOR PDEAD  
 .BYTE 20 ;R FOR PDEAD  
 .BYTE 10 ;QNDM INACTIVE  
 .BYTE 2 ;T/O TO PDEAD  
 .BYTE 10 ;T/O TO DEAD  
 .BYTE 4 ;MAX MESSAGE COUNTER  
 ; DMP DMV  
 DMVDF1: .WORD 5670 ;SELCT TIMER [3 SECS ] 454  
 DMVDF2: .WORD 13560 ;INTERVAL TIMER [6 SECS ] 1130

; GLOBAL DEFAULTS

GLBDEF: .WORD 15 ;NUMSYNC  
 DMVDF3: .WORD 13560 ;CARRIER WAIT TIMING [6 SECS] 1130  
 DMVDF4: .WORD 62 ;DELTA T 24  
 DMVDF5: .WORD 23420 ;DEAD T 1750  
 .WORD 0 ;POLL DELAY  
 GLBENJ:

;\*\*\*\*\*  
 ;  
 ; \* NOTE \* - THE VALUES FOR DMVDF1-DMVDF5 ARE ASSEMBLED FOR DMP IF  
 ; THIS IS A DMV THE INIT CODE CHANGES THESE VALUES TO DEFAULTS  
 ; FOR DMV. THIS IS POSSIBLE BECUASE THIS PROGRAM WILL BE LOADED  
 ; ONE TIME FOR EVERY DEVICE.  
 ;\*\*\*\*\*

; TRIB LIST OF POLL PARAMETERS

POLLIS: .BLKW 8.\*32.

; GLOBAL LIST OF POLL PARAMETERS

GLBPLS: .BLKW 5.

1	017232	000000	MPLY:	.WORD	0	;MULTIPLIER
2	017234	000000	RXPTR:	.WORD	0	;RECEIVER MESSAGE POINTER
3	017236	000000	TXPTR:	.WORD	0	;TRANSMITTER BUFFER POINTER
4	017240	000000	CMPPTR:	.WORD	0	;COMPARISON BUFFER POINTER
5	017242	000000	CMP TOT:	.WORD	0	;CMP MSG TOTAL
6	017244	000000	CTOTCC:	.WORD	0	;COMPARE BUFFER CHAR. COUNT
7	017246	000000	CCURAD:	.WORD	0	;CURRENT ADDR OF CMP BUFF TO ADD AT
8	017250	000000	DVTXA:	.WORD	0	;DEVICE TX ADDR
9	017252	000000	DVTCC:	.WORD	0	;DEVICE TX CHAR COUNT
10	017254	000000	DVTTB:	.WORD	0	;DEVICE TRIBN
11	017256	000000	DVTCT:	.WORD	0	;DEVICE TX MESSAGE COUNT
12	017260	000000	TXMTOT:	.WORD	0	;TX MSG TOTAL
13	017262	000000	TTOTCC:	.WORD	0	;TX BUFFER CHAR. COUNT
14	017264	000000	TCURAD:	.WORD	0	;CURRENT ADDR. OF TX BUFF TO ADD AT
15	017266	000000	DVRTB:	.WORD	0	;RECEIVE TRIBN
16	017270	000000	DVRXA:	.WORD	0	;DEVICE RX ADDR
17	017272	000000	DVRCC:	.WORD	0	;DEVICE RX CHAR COUNT
18	017274	000000	DVRCT:	.WORD	0	;DEVICE RX MESSAGE COUNT
19	017276	000000	RXTOT:	.WORD	0	;RX MSG TOTAL
20						
21	017300	000000	LN CNT:	.WORD	0	;NUMBER OF OPERATOR AWAKE MSGS
22	017302	000000	OPVAR:	.WORD	0	;HOLDER FOR OPTIONAL VARIABLE (1)
23	017304	000000	OPVAR1:	.WORD	0	;HOLDER FOR OPTION VARIABLE (2)
24	017306	000000	PSCNT:	.WORD	0	;PASS COUNTER
25	017310	000000	ERRCNT:	.WORD	0	;ERROR COUNTER
26	017312	000000	STADD:	.WORD	0	;START ADDR.
27	017314	000000	ENADD:	.WORD	0	;END ADDR. FOR DUMP
28	017316	000000	BYTBIT:	.WORD	0	;BYTE BIT FOR DUMP ROUTINE
29	017320	000000	CLNSET:	.WORD	0	;CLEANSET FLAG SET AND CLEARED IN CLEAN UP
30						;INDICATES TO OUTPUT HANDLER THAN NO OUTPUTS SHOULD
31						;BE PRINTED
32	017322	000000	RQIFLG:	.WORD	0	;RQI FLAG
33	017324	000000	FTLFLG:	.WORD	0	;USED AS FATEL ERROR FLAG
34	017326	000000	TSSFLG:	.WORD	0	;USED AS TSS FLAG
35	017330	000000	OVRCNT:	.WORD	0	;USED FOR QUE OVERFLOW FLAG
36						;OTHER MESSAGE RELATED STORAGE LOCATIONS
37						
38	017332	000000	MSGTYP:	.WORD	0	;TYPE OF DATA 0=0'S,1=1'S,2=10'S,3=01'S
39						;4=CCITT,5=QUICK FOX,6=ALPHA/ALM,7=OPER
40	017334	000000	CURCC:	.WORD	0	;TX/RX/CMP CHAR COUNT
41	017336	000000	CPTRR:	.WORD	0	;CURRENT RX POINTER
42	017338	000000	CPTR:	.WORD	0	;CURRENT POINTER
43	017342	000000	CURADD:	.WORD	0	;CURRENT TX/RX/CMP START ADDR
44	017344	000000	TOTCC:	.WORD	0	;TOTAL CHAR COUNT NOT MORE THEN "BUFLIM"
45	017346	000000	OFFSET:	.WORD	0	;OFFSET COUNT
46	017350	000000	TEMP:	.WORD	0	;TEMPORARY LOCATIONS (USED A LOT)
47	017352	000000	TEMP1:	.WORD	0	
48	017354	000000	TEMP2:	.WORD	0	
49	017356	000000	TEMP3:	.WORD	0	
50	017360	000000	TEMP4:	.WORD	0	
51	017362	000000	TEMP5:	.WORD	0	
52	017364	000000	SAVSP:	.WORD	0	;STACK POINTER SAVE AREA
53	017366	000000	CONOTM:	.WORD	0	;CONTROL OUT ERROR MSG. ADDRESS AND TSS AND GSS MSGS.
54	017370	000	GOOD:	.BYTE	0	;BYTE TO HOLD EXPECTED MESSAGE DATA BYTE FOR ERR REPORT
55	017371	000	BAD:	.BYTE	0	;BYTE TO HOLD RECEIVED MESSAGE DATA BYTE FOR ERR REPORT
56						

```

1          ;MORE INDEPENDENT CODE STORAGE LOCATIONS
2
3 017372 000000 LOGUNT: .WORD 0 ;LOC. TO HOLD LOGICAL UNIT NUMBER
4 017374 000000 PCADD:  .WORD 0 ;LOC. HOLD PC OF CALLING ROUTINE
5 017376 000000 DCLFLG: .WORD 0 ;LOC. TO HOLD DO CLEAN FLAG 1 IF DOCLEAN INIT 0 IF NOT.
6 017400 000000 RESFLG: .WORD 0 ;LOC TO HOLD FLAG (-1) THAT A RESTART WAS GIVEN
7 017402 000000 MODTYP: .WORD 0 ;DCLT MODE OF OPERATION TYPE
8          ; (0=REC-ONLY, 1=TX-ONLY, 2=PASSIVE-LOOPBK,
9          ; 3=ACTIVE-LOOPBK, 4=DOWN L.L., 5=TALK, 6=LISTEN)
10 017404 000000 MLTYP:  .WORD 0 ;MAINTENANCE LOOP TYPE (0=NONE, 1=INTERNAL TTL,
11          ; 2=CABLE, 3=MODEM-ANALOG LOOPBK (LOCAL),
12          ; 4=MODEM-DIGITAL LOOPBK (REMOTE), 5=MOP)
13 017406 000000 FMDPLX: .WORD 0 ;FULL OR HALF DUPLEX FLAG (1=FULL FROM P-TABLE)
14 017410 000002 PARAM:  .WORD 2 ;PROGRAM PARAMETERS
15          ; BIT0= STATUS MSGS TO OPR PRINTED (1=YES)
16          ; BIT1= DATA CHECKING DONE ON RCVD MSGS (1=YES)
17          ; BIT2= ECHO (TRANSMIT) RCV'D MSG.(PASSIVE)(1=YES)
18          ; BIT3= MODEM STATUS CHECK (1=YES)
19          ; BIT4= CRC CALC./CHECK DONE (1=YES)
20          ; BIT5= PROTOCOL EMULATION (1=YES)
21          ; BIT6= SPARE
22 017412 000000 RPASS:  .WORD 0 ;PASS NUMBER FROM RUN COMMAND
23 017414 000000 FLAG:   .WORD 0 ;DEVICE FLAG WORD
24 017416 000000 RUNING: .WORD 0 ; 1 = DCLT RUNNING(DEVICES ARE COMMUNICATING)
25
26          ;MODE DISPATCH TABLE
27 017420 060036 MODE:   .WORD RXONLY ;RX ONLY DISPATCH
28 017422 060064          .WORD TXONLY ;TX ONLY DISPATCH
29 017424 060122          .WORD PLCK  ;PASSIVE LOOP BACK DISP
30 017426 060150          .WORD ALCK  ;ACTIVE LOOP BACK DISP
31 017430 061420          .WORD DLL   ;DOWN LINK LOAD DISP
32 017432 062276          .WORD TALCK ;TALK MODE DISPATCH
33 017434 062530          .WORD LISCK ;LISTEN MODE DISPATCH
34
35
36          .SBTTL          CLOCK TABLES, EVENT LOG AND POINTERS
37 017436 000000 CLKCSR: .WORD 0 ;CLOCK CSR ADDRESS
38 017440 000000 LLKBR:  .WORD 0 ;CLOCK INTERRUPT LEVEL
39 017442 000000 CLKVEC: .WORD 0 ;CLOCK INTERRUPT VECTOR
40 017444 000074 CLKHZ:  .WORD 60. ;CLOCK'S HERTZ RATE
41 017446 000000 CLKEN:  .WORD 0 ;CLOCK'S CSR VALUE TO INTRPT. ENABLE IT
42
43 017450 000000 TIMMIN: .WORD 0 ;PLACE TO KEEP TIME SINCE START
44 017452 000000 TIMSEC: .WORD 0
45 017454 000000 TIMTCK: .WORD 0 ;PLACE TO KEEP # OF TICKS/SEC
46
47 017456 000000 TIMER1: .WORD 0 ;EVENT TIMER #1 (TICKS)
48 017460 000000 TIMER2: .WORD 0 ;EVENT TIMER #2 (TICKS)
49 017462 000000 TIMERS: .WORD 0 ;EVENT TIMER #3 (SECONDS)
50

```

CZCLMCO DMP/V 11 DCLT MACRO V05.00  
CLOCK TABLES, EVENT LOG AND POINTERS

Thursday 22-Mar-84 16:24 Page 33

1  
2 017464 017466  
3 017466  
4 020522  
5  
6  
7  
8 020524 000000  
9

.EVENT LOG TABLE AND ITS NEXT ENTRY POINTER  
EVTPTL: .WORD EVTLOG ; POINTER TO NEXT FREE SPACE IN EVENT LOG  
EVTLOG: .BLKW 270. ; EVENT LOG BUFFER  
EVTEND: .BLKW 1. ; APPROXIMATE END OF EVENT TABLE (ALLOWS CIRCULAR QUE)  
  
.SBTTL MODEM DATA SECTION  
MODS: .WORD 0 ; MODEM STATUS

1  
 2  
 3 020526 000004  
 4 020530 000010  
 5 020532 000001  
 6 020534 000040  
 7 020536 000200  
 8 020540 040000  
 9 020542 002000  
 10 020544  
 11  
 12  
 13  
 14 020544 031530  
 15 020546 031534  
 16 020550 031540  
 17 020552 031544  
 18 020554 031550  
 19 020556 031554  
 20 020560 031560  
 21  
 22  
 23  
 24  
 25 020562 030001  
 26 020564 030025  
 27 020566 030054  
 28 020570 030101  
 29 020572 030127  
 30 020574 030174  
 31 020576 030144  
 32 020600 026765  
 33 020602 030222  
 34 020604 030257  
 35 020606 030312  
 36  
 37  
 38  
 39 020610 000000  
 40 020612 000000  
 41 020614 000000  
 42 020616 000000  
 43 020620 000000  
 44 020622 000000  
 45  
 46  
 47  
 48 020624 043444  
 49 020626 043444  
 50 020630 043444  
 51 020632 043444  
 52 020634 043522  
 53 020636 043622  
 54 020640 044056  
 55 020642 044136  
 56 020644 044056  
 57 020646 043776

```

;TABLE OF MODEM SIGNAL BIT DEFINITIONS
MOBITS: .WORD   CTS           ;CLEAR TO SEND (CIRCUIT CB)
         .WORD   DSR           ;DATA SET READY (CIRCUIT CC)
         .WORD   DCD           ;DATA CARRIER DETECT (CIRCUIT CF)
         .WORD   RTS           ;REQUEST TO SEND (CIRCUIT CA)
         .WORD   RI            ;RING INDICATOR (CIRCUIT CE)
         .WORD   SQD          ;SIGNAL QUALITY DETECT (CIRCUIT CG)
         .WORD   TM            ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)
  
```

```

;TABLE OF ADDRESSES OF MODEM SIGNAL MESSAGE POSITIONS
MOMSGS: .WORD   EVMCTS        ;CLEAR TO SEND (CIRCUIT CB)
         .WORD   EVMDSR       ;DATA SET READY (CIRCUIT CC)
         .WORD   EVMDCD       ;DATA CARRIER DETECT (CIRCUIT CF)
         .WORD   EVMRTS       ;REQUEST TO SEND (CIRCUIT CA)
         .WORD   EVMRI        ;RING INDICATOR (CIRCUIT CE)
         .WORD   EVMSQD       ;SIGNAL QUALITY DETECT (CIRCUIT CG)
         .WORD   EVMTM        ;MODEM IN TEST MODE (RS 449 ONLY CIRCUIT TM)
  
```

```

;TABLE OF ADDRESSES OF EVENT DESCRIPTION MESSAGES
; ORDER CORRESPONDS TO MESSAGE TYPE VALUES
EVTLST: .WORD   EDTXQ         ;TRANSMIT MESSAGE QUEUED
         .WORD   EDTXC         ;TRANSMIT OF MESSAGE COMPLETE
         .WORD   EDRXQ         ;RECEIVE MESSAGE SPACE QUEUED
         .WORD   EDRXC         ;MESSAGE RECEIVED - RECEIVE COMPLETE
         .WORD   EDDER         ;DEVICE INFORMATION
         .WORD   EDDVI        ;DEVICE INITIALIZE STARTED
         .WORD   EDDCK        ;DATA COMPARISON DONE
         .WORD   LPO           ;NULL STRING
         .WORD   EDDLE        ;DATA COMPARE LENGTH ERROR
         .WORD   EDDDE        ;DATA COMPARE DATA ERROR
         .WORD   EDEOP        ;END OF PASS
  
```

```

;LOCATIONS USED DURING EVENT REPORTING
EVTSEC: .WORD   0             ;TEMPORARY LOCS TO KEEP EVENT TIME WHILE REPORTING
EVTMIN: .WORD   0
EVTTC: .WORD   0
EVTADD: .WORD   0             ;TEMP. LOC. TO HOLD ADDRESS DURING EVENT REPORTING
EVTBC: .WORD   0             ; " " " BYTE COUNT "
EVTIMP: .WORD   0             ; " " " OTHER DATA "
  
```

```

;REPORT CODING DISPATCH TABLE
RPTDSP: .WORD   RPTTXQ       ;TRANSMIT QUEUED ENTRY DECODING
         .WORD   RPTTXQ       ;TRANSMIT COMPLETE ENTRY DECODING
         .WORD   RPTTXQ       ;RECEIVER QUEUED ENTRY DECODING
         .WORD   RPTTXQ       ;RECEIVER COMPLETE ENTRY DECODING
         .WORD   RPTDER       ;DEVICE ERROR ENTRY DECODING
         .WORD   RPTDVI       ;DEVICE INIT ENTRY DECODING
         .WORD   RPTDCK       ;DATA COMPARISON ENTRY DECODING
         .WORD   RPTMSC       ;PLACE HOLDER
         .WORD   RPTDLE       ;DATA COMPARISON LENGTH ERROR
         .WORD   RPTDDE       ;DATA COMPARISON DATA ERROR
  
```



CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22-Mar-84 16:24 Page 34 1  
MODEM DATA SECTION

```
58 020650 043672          .WORD  RPTEOP  ;END OF PASS
59
60
61 020652 000000          DEV1:  .WORD  0          ;TEMP I OCS TO HOLD DATA FOR EVENT REPORTING
62 020654 000000          DEV2:  .WORD  0          ; AND SHOW MODE,... SUBROUTINE
63 020656 000000          DEV3:  .WORD  0
64 020660 000000          DEV4:  .WORD  0
```

1									
2	020662	032112	.SBTTL		TABLE FOR TSS ASCII AND ROUTINES				
3	020664	032146	TSSLST:	.WORD	TSS0A	; POINTER FOR OFFSET 0	ASCII		
4	020666	032216		.WORD	TSS1A	; POINTER FOR OFFSET 1	ASCII		
5	020670	032252		.WORD	TSS2A	; POINTER FOR OFFSET 2	ASCII		
6	020672	032325		.WORD	TSS3A	; POINTER FOR OFFSET 3	ASCII		
7	020674	032373		.WORD	TSS4A	; POINTER FOR OFFSET 4	ASCII		
8	020676	032455		.WORD	TSS5A	; POINTER FOR OFFSET 5	ASCII		
9	020700	032545		.WORD	TSS6A	; POINTER FOR OFFSET 6	ASCII		
10	020702	032603		.WORD	TSS7A	; POINTER FOR OFFSET 7	ASCII		
11	020704	032637		.WORD	TSS10A	; POINTER FOR OFFSET 10	ASCII		
12	020706	032675		.WORD	TSS11A	; POINTER FOR OFFSET 11	ASCII		
13	020710	032772		.WORD	TSS12A	; POINTER FOR OFFSET 12	ASCII		
14	020712	033066		.WORD	TSS13A	; POINTER FOR OFFSET 13	ASCII		
15	020714	033151		.WORD	TSS14A	; POINTER FOR OFFSET 14	ASCII		
16	020716	033235		.WORD	TSS15A	; POINTER FOR OFFSET 15	ASCII		
17	020720	033320		.WORD	TSS16A	; POINTER FOR OFFSET 16	ASCII		
18	020722	033404		.WORD	TSS17A	; POINTER FOR OFFSET 17	ASCII		
19	020724	033476		.WORD	TSS20A	; POINTER FOR OFFSET 20	ASCII		
20	020726	033565		.WORD	TSS21A	; POINTER FOR OFFSET 21	ASCII		
21	020730	033653		.WORD	TSS22A	; POINTER FOR OFFSET 22	ASCII		
22	020732	033740		.WORD	TSS23A	; POINTER FOR OFFSET 23	ASCII		
23	020734	034027		.WORD	TSS24A	; POINTER FOR OFFSET 24	ASCII		
24	020736	034114		.WORD	TSS25A	; POINTER FOR OFFSET 25	ASCII		
25	020740	034145		.WORD	TSS26A	; POINTER FOR OFFSET 26	ASCII		
26	020742	034230		.WORD	TSS27A	; POINTER FOR OFFSET 27	ASCII		
27	020744	034302		.WORD	TSS30A	; POINTER FOR OFFSET 30	ASCII		
28	020746	034361		.WORD	TSS31A	; POINTER FOR OFFSET 31	ASCII		
29	020750	034444		.WORD	TSS32A	; POINTER FOR OFFSET 32	ASCII		
30	020752	034527		.WORD	TSS33A	; POINTER FOR OFFSET 33	ASCII		
31	020754	034606		.WORD	TSS34A	; POINTER FOR OFFSET 34	ASCII		
32	020756	034665		.WORD	TSS35A	; POINTER FOR OFFSET 35	ASCII		
33	020760	034737		.WORD	TSS36A	; POINTER FOR OFFSET 36	ASCII		
34					TSS37A	; POINTER FOR OFFSET 37	ASCII		
35									
36									
37									
38									
39									
40	020762	000							
41	020763	002							
42	020764	000							
43	020765	002							
44	020766	002							
45	020767	002							
46	020770	002							
47	020771	000							
48	020772	000							
49	020773	000							
50	020774	004							
51	020775	004							
52	020776	004							
53	020777	004							
54	021000	004							
55	021001	002							
56	021002	002							
57	021003	002							

;TABLE FOR TSS ACTION ROUTINES  
;IF BYTE = 0 USE WORD ROUTINE  
;IF BYTE = 2 USE BYTE/BYTE ROUTINE  
;IF BYTE = 4 USE BYTE SPECIAL ROUTINE

TSSIND:	.BYTE	0	; INDEX FOR	TSS 0
	.BYTE	2	; INDEX FOR	TSS 1
	.BYTE	0	; INDEX FOR	TSS 2
	.BYTE	2	; INDEX FOR	TSS 3
	.BYTE	2	; INDEX FOR	TSS 4
	.BYTE	2	; INDEX FOR	TSS 5
	.BYTE	2	; INDEX FOR	TSS 6
	.BYTE	0	; INDEX FOR	TSS 7
	.BYTE	0	; INDEX FOR	TSS 10
	.BYTE	0	; INDEX FOR	TSS 11
	.BYTE	4	; INDEX FOR	TSS 12
	.BYTE	4	; INDEX FOR	TSS 13
	.BYTE	4	; INDEX FOR	TSS 14
	.BYTE	4	; INDEX FOR	TSS 15
	.BYTE	4	; INDEX FOR	TSS 16
	.BYTE	2	; INDEX FOR	TSS 17
	.BYTE	2	; INDEX FOR	TSS 20
	.BYTE	2	; INDEX FOR	TSS 21

58	021004	002	.BYTE	2	;INDEX FOR	TSS 22
59	021005	002	.BYTE	2	;INDEX FOR	TSS 23
60	021006	002	.BYTE	2	;INDEX FOR	TSS 24
61	021007	002	.BYTE	2	;INDEX FOR	TSS 25
62	021010	000	.BYTE	0	;INDEX FOR	TSS 26
63	021011	002	.BYTE	2	;INDEX FOR	TSS 27
64	021012	000	.BYTE	0	;INDEX FOR	TSS 30
65	021013	002	.BYTE	2	;INDEX FOR	TSS 31
66	021014	002	.BYTE	2	;INDEX FOR	TSS 32
67	021015	002	.BYTE	2	;INDEX FOR	TSS 33
68	021016	002	.BYTE	2	;INDEX FOR	TSS 34
69	021017	002	.BYTE	2	;INDEX FOR	TSS 35
70	021020	000	.BYTE	0	;INDEX FOR	TSS 36
71	021021	000	.BYTE	0	;INDEX FOR	TSS 37
72						
73	021022	000000	TSSE:	.WORD	0	;WORD FOR LAST TSS TO BE PRINTED
74	021024	000000	TSSA:	.WORD	0	;WORD FOR ADDRESS
75	021026	000000	TSSKEY:	.WORD	0	;KEY WORD FOR READING TSS
76						

1  
2  
3  
4 021030 035004  
5 021032 035045  
6 021034 035103  
7 021036 035142  
8 021040 035177  
9 021042 035234  
10 021044 035271  
11 021046 035347  
12 021050 035425  
13 021052 035477  
14 021054 035517  
15 021056 035555  
16 021060 035616  
17 021062 035637  
18 021064 035747  
19 021066 036057  
20 021070 036141  
21 021072 036206  
22 021074 036253  
23 021076 036320  
24 021100 036365  
25 021102 036432  
26 021104 036477  
27 021106 036521  
28 021110 036543  
29 021112 036576  
30 021114 036653  
31 021116 036736  
32 021120 036761  
33 021122 037026  
34 021124 037050  
35 021126 037071  
36  
37  
38  
39  
40  
41  
42 021130 002  
43 021131 002  
44 021132 002  
45 021133 002  
46 021134 002  
47 021135 002  
48 021136 002  
49 021137 002  
50 021140 002  
51 021141 000  
52 021142 002  
53 021143 002  
54 021144 000  
55 021145 004  
56 021146 004  
57 021147 002

.SBTTL TABLE FOR GSS ASCII AND ACTION

GSSLST: .WORD GSS0A ; POINTER FOR OFFSET 0 ASCII  
 .WORD GSS1A ; POINTER FOR OFFSET 1 ASCII  
 .WORD GSS2A ; POINTER FOR OFFSET 2 ASCII  
 .WORD GSS3A ; POINTER FOR OFFSET 3 ASCII  
 .WORD GSS4A ; POINTER FOR OFFSET 4 ASCII  
 .WORD GSS5A ; POINTER FOR OFFSET 5 ASCII  
 .WORD GSS6A ; POINTER FOR OFFSET 6 ASCII  
 .WORD GSS7A ; POINTER FOR OFFSET 7 ASCII  
 .WORD GSS10A ; POINTER FOR OFFSET 10 ASCII  
 .WORD GSS11A ; POINTER FOR OFFSET 11 ASCII  
 .WORD GSS12A ; POINTER FOR OFFSET 12 ASCII  
 .WORD GSS13A ; POINTER FOR OFFSET 13 ASCII  
 .WORD GSS14A ; POINTER FOR OFFSET 14 ASCII  
 .WORD GSS15A ; POINTER FOR OFFSET 15 ASCII  
 .WORD GSS16A ; POINTER FOR OFFSET 16 ASCII  
 .WORD GSS17A ; POINTER FOR OFFSET 17 ASCII  
 .WORD GSS20A ; POINTER FOR OFFSET 20 ASCII  
 .WORD GSS21A ; POINTER FOR OFFSET 21 ASCII  
 .WORD GSS22A ; POINTER FOR OFFSET 22 ASCII  
 .WORD GSS23A ; POINTER FOR OFFSET 23 ASCII  
 .WORD GSS24A ; POINTER FOR OFFSET 24 ASCII  
 .WORD GSS25A ; POINTER FOR OFFSET 25 ASCII  
 .WORD GSS26A ; POINTER FOR OFFSET 26 ASCII  
 .WORD GSS27A ; POINTER FOR OFFSET 27 ASCII  
 .WORD GSS30A ; POINTER FOR OFFSET 30 ASCII  
 .WORD GSS31A ; POINTER FOR OFFSET 31 ASCII  
 .WORD GSS32A ; POINTER FOR OFFSET 32 ASCII  
 .WORD GSS33A ; POINTER FOR OFFSET 33 ASCII  
 .WORD GSS34A ; POINTER FOR OFFSET 34 ASCII  
 .WORD GSS35A ; POINTER FOR OFFSET 35 ASCII  
 .WORD GSS36A ; POINTER FOR OFFSET 36 ASCII  
 .WORD GSS37A ; POINTER FOR OFFSET 37 ASCII

; TABLE FOR GSS ACTION ROUTINES  
 ; IF BYTE = 0 USE WORD ROUTINE  
 ; IF BYTE = 2 USE BYTE/BYTE ROUTINE  
 ; IF BYTE = 4 USE BYTE SPECIAL ROUTINE

GSSIND: .BYTE 2 ; INDEX FOR GSS 0  
 .BYTE 2 ; INDEX FOR GSS 1  
 .BYTE 2 ; INDEX FOR GSS 2  
 .BYTE 2 ; INDEX FOR GSS 3  
 .BYTE 2 ; INDEX FOR GSS 4  
 .BYTE 2 ; INDEX FOR GSS 5  
 .BYTE 2 ; INDEX FOR GSS 6  
 .BYTE 2 ; INDEX FOR GSS 7  
 .BYTE 2 ; INDEX FOR GSS 10  
 .BYTE 0 ; INDEX FOR GSS 11  
 .BYTE 2 ; INDEX FOR GSS 12  
 .BYTE 2 ; INDEX FOR GSS 13  
 .BYTE 0 ; INDEX FOR GSS 14  
 .BYTE 4 ; INDEX FOR GSS 15  
 .BYTE 4 ; INDEX FOR GSS 16  
 .BYTE 2 ; INDEX FOR GSS 17

58	021150	002	.BYTE	2	; INDEX FOR	GSS 20
59	021151	002	.BYTE	2	; INDEX FOR	GSS 21
60	021152	002	.BYTE	2	; INDEX FOR	GSS 22
61	021153	002	.BYTE	2	; INDEX FOR	GSS 23
62	021154	002	.BYTE	2	; INDEX FOR	GSS 24
63	021155	002	.BYTE	2	; INDEX FOR	GSS 25
64	021156	000	.BYTE	0	; INDEX FOR	GSS 26
65	021157	000	.BYTE	0	; INDEX FOR	GSS 27
66	021160	000	.BYTE	0	; INDEX FOR	GSS 30
67	021161	002	.BYTE	2	; INDEX FOR	GSS 31
68	021162	002	.BYTE	2	; INDEX FOR	GSS 32
69	021163	000	.BYTE	0	; INDEX FOR	GSS 33
70	021164	000	.BYTE	0	; INDEX FOR	GSS 34
71	021165	000	.BYTE	0	; INDEX FOR	GSS 35
72	021166	000	.BYTE	0	; INDEX FOR	GSS 36
73	021167	000	.BYTE	0	; INDEX FOR	GSS 37
74						

```

1      .SBTTL          COMMAND LINE ACTION TREE
2
3      ;SAMPLE CLI TREE NODE  (ALWAYS AT LEAST 1 WORD)
4
5      ; ACTION  ! CHAR CODE !
6
7      ; MISS DISPLACEMENT !          ONLY IF "MISS" ARGUMENT DEFINED
8
9      ; NEXT NODE DISPLMNT !          ONLY IF "ASCII" ARGUMENT DEFINED
10
11     ; ASCIZ MATCH STRING !          ONLY IF "ASCII" ARGUMENT DEFINED
12     ; ( .EVEN ) !
13     ; -----
14
15
16 021170 CLITRE:
17
18 ;FIRST KEYWORD
19 021170 CLI CLISPA,0,N10$ ;SKIP ANY LEADING SPACES
20 021174 N10$: CLI <'?'>,HLP,N42$ ;IS THE FIRST NON SP CHAR A ?
21 021200 CLI CLIEXI,0 ; IF YES DO "HLP" AND EXIT
22 021202 N42$: CLI CLISTR,HLP,N43$,<'HELP'> ;ELSE, IS FIRST WORD A "HELP"
23 021216 CLI CLIEXI,0 ; IF YES DO "HLP" AND EXIT
24 021220 N43$: CLI CLISTR,PRNT,N44$,<'PRINT'> ;ELSE, IS FIRST WORD A "PRINT"
25 021234 CLI CLIEXI,0 ; IF YES DO "PRINT" AND EXIT
26 021236 N44$: CLI CLISTR,EXIT,N45$,<'EXIT'> ;ELSE, IS FIRST WORD A "EXIT"
27 021252 CLI CLIEXI,0 ; IF YES DO "EXIT" AND EXIT
28 021254 N45$: CLI CLISTR,RUN,N46$,<'RUN'> ;ELSE, IS FIRST WORD A "RUN"
29 021266 CLI CLIBR,0,N80$ ; IF YES DO "RUN" & GOTO N80$
30 021272 N46$: CLI CLISTR,NOTNUF,N40$,<'DUMP'> ;ELSE, IS FIRST WORD A "DUMP"
31 021306 CLI CLIBR,0,N50$ ; IF YES GOTO N80$
32 021312 N40$: CLI CLISTR,CLEAR,N47$,<'CLEAR'> ;ELSE, IS FIRST WORD A "CLEAR"
33 021326 CLI CLIBR,NOTNUF,N100$ ; IF YES DO "CLR" & GOTO N100$
34 021332 N47$: CLI CLISTR,CTPP,N20$,<'TRIB'> ;ELSE IS FIRST WORD TRIB
35 021346 CLI CLIBR,NOTNUF,N105$
36 021352 N20$: CLI <'S'>,NOTNUF,N30$ ;ELSE, IS FIRST CHAR. A 'S'
37 021356 CLI CLISTR,SHOW,N25$,<'HOW'> ; IF YES IS REST OF WORD "HOW"
38 021370 CLI CLIBR,0,N100$ ; IF YES, DO "SHOW",BR N100$
39 021374 N25$: CLI CLISTR,0,N30$,<'ET'> ; ELSE, IS REST OF WORD ET'
40 021406 CLI CLIBR,0,N110$ ; IF YES, DO 'SET', BR N110$
41 021412 N30$: CLI CLIERR,0 ; OTHERWISE "ILL CMD" EXIT
42
43 ;SECOND KEYWORD (MODE=) FOR RUN COMMAND
44
45 021414 N80$: CLI CLISPA,0,N30$ ;SKIP LEADING SPS, IF NONE ERR
46 021420 N81$: CLI CLISTR,NOTNUF,N30$,<'MODE'> ;IS NEXT WORD "MODE="
47 021434 CLI <'='>,0,N30$ ; IF NO, IT S WRONG ERR EXIT
48 021440 CLI CLISTR,ATVMOD,N82$,<'ACTIVE'> ;IS NEXT WORD "ACTIVE"
49 021456 CLI CLIBR,0,N115$ ; IF YES, DO "ACTIVE",BR N115$
50 021462 N82$: CLI CLISTR,PASMOD,N83$,<'PASSIVE'> ;IS NEXT WORD "PASSIVE"
51 021500 CLI CLIBR,0,N115$ ; IF YES, DO "PASSVE",BR N115$
52 021504 N83$: CLI CLISTR,RECMOD,N84$,<'RECEIVE'> ;IS NEXT WORD "RECEIVE"
53 021522 CLI CLIBR,0,N115$ ; IF YES, DO "RECVE",BR N115$
54 021526 N84$: CLI CLISTR,LISMOD,N85$,<'LISTEN'> ;IS NEXT WORD "LISTEN"
55 021544 CLI CLIBR,0,N115$ ; IF YES, DO "LISTEN",BR N115$
56 021550 N85$: CLI CLISTR,DLLMOD,N86$,<'DOWNLINELOAD'> ;IS NEXT WORD 'DOW...'
57 021574 CLI CLIBR,0,N115$ ; IF YES, DO 'DWNLL',BR N115$

```

```

58 021600      N86$:  CLI      <'T>,0,N30$      ;IS NEXT CHAR A 'T
59 021604      CLI      CLISTR,TRAMOD,N87$,<'RANSMIT' > ; IS REST OF WORD "RANSMIT"
60 021622      CLI      CLIBR,0,N115$      ; IF YES, DO "TRANSM",BR N115$
61 021626      N87$:  CLI      CLISTR,TALMOD,N30$,<'ALK' > ; IS REST OF WORD "ALK"
62 021640      CLI      CLIBR,0,N115$      ; IF YES, DO "TALK",BR N115$
63             ; IF NO, ERROR - EXIT
64
65             ;SECOND KEYWORD (FOR CLEAR OR SHOW)
66 021644      N100$: CLI      CLISPA,0,N30$      ;SKIP LEADING SPACES, NONE=ERR
67 021650      N102$: CLI      CLISTR,CSHEXP,N104$,<'EXPECTBUFF' > ;IS NEXT WORD "EXPE..."
68 021672      CLI      CLIEXI,0      ; IF YES, DO CLR EXP,EXIT
69 021674      N104$: CLI      CLISTR,CSHTRN,N30$,<'TRANSMITBUFF' > ;IS NEXT WORD "TRANS.."
70 021720      CLI      CLIEXI,0      ; IF YES, DO CLR TRN,EXIT
71             ;IF NO ERROR EXIT
72
73
74             ;SECOND KEYWORD (FOR SET)
75 021722      N110$: CLI      CLISPA,0,N30$
76 021726      N111$: CLI      CLISTR,SETEXP,N112$,<'EXPECTMSG' >
77 021746      CLI      CLIBR,0,N120$
78 021752      N112$: CLI      CLISTR,SETTRN,N30$,<'TRANSMITMSG' >
79 021774      CLI      CLIBR,0,N120$
80
81             ;GET ADDRESSES FOR DUMP COMMAND
82 022000      N50$:  CLI      CLIALP,0,N51$
83 022004      N51$:  CLI      CLISPA,0,N52$
84 022010      N52$:  CLI      CLIOCT,DMPS,N30$
85 022014      CLI      <' ->,NOTNUF,N125$
86 022020      CLI      CLIOCT,DMPE,N30$
87 022024      CLI      <' />,NOTNUF,N125$
88 022030      CLI      <'B>,DMPQ,N30$
89 022034      CLI      CLIBR,0,N125$
90
91             ;QUALIFIERS FOR THE RUN COMMAND
92 022040      N115$: CLI      CLIALP,0,N114$
93 022044      N114$: CLI      <' />,NOTNUF,N125$
94 022050      CLI      CLISTR,NO,N116$,<'NO' >
95 022062      N116$: CLI      <'C>,0,N117$
96 022066      CLI      CLISTR,CHECK,N117$,<'HECK' >
97 022102      CLI      CLIBR,0,N115$
98
106
107 022105      N117$: CLI      CLISTR,STATUS,N118$,<'STATUS' >
108 022124      CLI      CLIBR,0,N115$
109 022130      N118$: CLI      CLISTR,ECHO,N130$,<'ECHO' >
110 022144      CLI      CLIBR,0,N115$
111
124 022150      N130$: CLI      CLISTR,0,N132$,<'PASS' >
125 022164      CLI      CLIBR,0,N150$
126
127 022170      N132$: CLI      CLISTR,MOSC,N131$,<'MODEM' >
128 022204      CLI      CLIBR,0,N115$
129
130 022210      N131$: CLI      CLISTR,0,N30$,<'LOOP' >
131 022224      CLI      CLIBR,0,N140$
132
133             ;GET MESSAGE TYPE FOR SET MESSAGE COMMANDS

```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 37 2  
 COMMAND LINE ACTION TREE

```

134 022230 N120$: CLI <'>,0,N30$
135
136 ; LOOK FOR DEFAULT MESSAGE NAME
137 022234 N60$: CLI CLISTR,CMSG1,N61$, <' ONFS' >
138 022250 CLI CLIBR,0,N121$
139 022254 N61$: CLI CLISTR,CMSG0,N62$, <' ZEROES' >
140 022272 CLI CLIBR,0,N121$
141 022276 N62$: CLI CLISTR,CMSG2,N63$, <' 1ALT' >
142 022312 CLI CLIBR,0,N121$
143 022316 N63$: CLI CLISTR,CMSG3,N64$, <' 0ALT' >
144 022332 CLI CLIBR,0,N121$
145 022336 N64$: CLI CLISTR,CMSG5,N65$, <' ITP' >
146 022352 CLI CLIBR,0,N121$
147 022356 N65$: CLI CLISTR,CMSG4,N66$, <' CCITT' >
148 022372 CLI CLIBR,0,N121$
149 022376 N66$: CLI CLISTR,CMSG6,N67$, < ALPHA' >
150 022412 CLI CLIBR,0,N121$
151 022416 N67$: CLI CLISTR,SETET,N68$, <' TRANSMIT >
152 022436 CLI CLIBR,0,N125$
153
154 ; LOOK FOR QUOTED MESSAGE
155 022442 N68$: CLI <'>,OPRMSG,N30$
156 022446 N70$: CLI <'>,ENDQO,N71$
157 022452 CLI CLIBR,0,N121$
158 022456 N71$: CLI CLISPA,0,N72$
159 022462 N72$: CLI CLIALN,0,N73$ ; ONLY A Z,SP,TAB, OR 0 9 BETWEEN ''S
160 022466 CLI CLIBR,0,N70$
161 022472 N73$: CLI CLIERR,BADCHR ; PRINT ERROR IF NONE LEGAL CHAR FOR ''S
162
163 ;GET QUALIFIERS (SIZE OR COPY) FOR SET MESSAGE COMMANDS
164 022474 N121$: CLI CLIALP,0,N123$
165 022500 N123$: CLI <'>,NOTNUF,N125$
166 022504 CLI CLISTR,SIZE,N122$, <' SIZE' >
167 022520 CLI CLIBR,0,N126$
168 022524 N122$: CLI CLISTR,QCOPY,N30$, <' COPY' >
169 022540 CLI CLIBR,0,N126$
170
171 ;NUMER FOR SIZE OR COPY
172 022544 N126$: CLI <'>,0,N30$
173 022550 CLI CLIDEC,NUM,N30$
174 022554 CLI CLIBR,0,N121$
175
176 ;GET MAINTENANCE LOOP TYPE FOR RUN "LOOP" QUALIFIER
177 022560 N140$: CLI <'>,0,N30$
178
187
188 022564 N141$: CLI CLISTR,TTLLOP,N142$, <' INTERNAL TTL' >
189 022606 CLI CLIBR,0,N115$
190 022612 N142$: CLI CLISTR,CBLLOP,N143$, <' CABLE' >
191 022626 CLI CLIBR,0,N115$
192 022632 N143$: CLI CLISTR,LMDLOP,N144$, <' LOCAL MODEM' >
193 022654 CLI CLIBR,0,N115$
194 022660 N144$: CLI CLISTR,RMDLOP,N30$, < REMOTE MODEM' >
195 022702 CLI CLIBR,0,N115$
196
197 ;GET LINE NUMBER FOR "PASS' RUN QUALIFIER
198 022706 N150$: CLI <'>,0,N30$

```



CZCLMCO DMP/V 11 DCLT MACPO V05.00 Thursday 22 Mar 84 16:24 Page 37 3  
COMMAND LINE ACTION TREE

```

199 022712          CLI      CLIDEC,PASC,N308
200 022716          CLI      CLIBR,O,N1158
201                ;GET TRIB SHOW OR ADDR FOR KILL OR ESTABLISH
202 022722          N1058: CLI      CLISPA,NOTNUF,N1068
203 022726          N1068: CLI      CLISTR,SLST,N1078,<'SHOW'>
204 022742          CLI      CLIEXI,O
205 022744          N1078: CLI      CLISTR,ETRB,N1088,<'ESTABLISH'>
206 022764          CLI      CLIBR,O,N1608
207 022770          N1088: CLI      CLISTR,KTRB,N308,<'KILL'>
208 023004          N1608: CLI      <'>,O,N308
209 023010          N1618: CLI      CLISTR,KALL,N1628,<'ALL'>
210 023022          N1628: CLI      CLIDEC,EKTB,N308
211 023026          CLI      CLISTR,ETWS,N1638,<' /W'>
212 023040          N1638: CLI      54,NOTNUF,N1258
213 023044          CLI      CLIBR,O,N1618
214
215                ;END OF -LINE
216 023050          N1258: CLI      CLIEXI,O
217

```

;LOOKING FOR .

C7

```

14
15
16 ;DEVICE DEPENDENT STORAGE LOCATIONS FOR
17 ; CURRENT DEVICE PARAMETERS
18
19 023052 SEL0:
20 023052 000000 BSEL0: .WORD 0 ;ADDRESSES OF REGISTERS SEL0 THRU BSEL7
21 023054 000000 BSEL1: .WORD 0
22 023056 SEL2:
23 023056 000000 BSEL2: .WORD 0
24 023060 000000 BSEL3: .WORD 0
25 023062 SEL4:
26 023062 000000 BSEL4: .WORD 0
27 023064 000000 BSEL5: .WORD 0
28 023066 SEL6:
29 023066 000000 BSEL6: .WORD 0
30 023070 000000 BSEL7: .WORD 0
31
32
33 023072 000000 INVEC: .WORD 0 ;INPUT INTERRUPT VECTOR ADDRESS
34 023074 000000 OUTVEC: .WORD 0 ;OUTPUT INTERRUPT VECTOR ADDRESS
35 023076 000000 INTPRI: .WORD 0 ;INTERRUPT PRIORITY
36 023100 000000 OPTYP: .WORD 0 ;OPTION TYPE
37 023102 000000 DEVPAR: .WORD 0 ;DEVICE PARAM. BIT 0 BIT1
    ; 1 MTP CONT
    ; 0 PTP TRIB
38
39
40 023104 000000 STATYP: .WORD 0 ;STATION TYPE
41 ; DEVICE ERROR MSG TABLES
42 023106 000000 CONCLS: .WORD 0 ;TABLE HOLDER
43 023110 041075 .WORD RXTHEM ;RX THRESHOLD ERROR MESSAGE ADDR.
44 023112 041075 .WORD TXTHEM ;TX THRESHOLD ERROR MESSAGE ADDR
45 023114 040076 .WORD SLTHEM ;SELECT THRESHOLD MESSAGE
46 023116 040117 .WORD STRCM ;DDCMP START REC MESSAGE ADDR.
47 023120 040140 .WORD MARM ;DDCMP MAINT REC IN RUN
48 023122 040161 .WORD MARHM ;MAINT RECEIVED IN HALD
49 023124 040203 .WORD STRHM ;START REC. IN MAINT MESSAGE.
50 023126 041075 .WORD PE142M ;SPARE
51 023130 040245 .WORD DEATHM ;DEAD TRIB MESSAGE
52 023132 040257 .WORD RUSH ;RUN STATE SET IN ERROR
53 023134 040275 .WORD BABTM ;BABLING TRIB MESSAGE
54 023136 040312 .WORD STREAM ;STREAMING TRIB MESSAGE
55 023140 040226 .WORD RIM ;RING DETECTED
56 ;PROCEDURE ERRORS
57
58 023142 040331 CONCLS: .WORD PE100M ;NO MODE DEF
59 023144 040345 .WORD PE102M ;ILLEGAL TYPE
60 023146 040367 .WORD PE104M ;ILLEGAL MODE CHANGE
61 023150 040413 .WORD PE106M ;CONTROL IN TO UNESTABLISHED TRIB
62 023152 040444 .WORD PE110M ;NON-GLOBAL TO TRIB 0
63 023154 040466 .WORD PE112M ;ILLEGAL REQUEST
64 023156 040517 .WORD PE114M ;ATTEMPT TO ESTABLISH MORE THAN MAX TRIBS
65 023160 040542 .WORD PE116M ;ESTABLISH TO ALREADY ESTABLISHED
66 023162 040576 .WORD PE120M ;ILLEGAL CONTROL IN
67 023164 040622 .WORD PE122M ;ASSIGN BUFFER FOR UNESTABLISHED TRIB
68 023166 040652 .WORD PE124M ;ASSIGN BUFFER FOR HALTED TRIB
69 023170 040701 .WORD PE126M ;ASSIGN BUFFER WITH BYTE COUNT =0
70 023172 040730 .WORD PE130M ;ASSIGN TX BUFFER TO TRIB 0
  
```

71	023174	040756	.WORD	PE132M	;ATTEMPT TO R/W RESERVED ISS/GSS	
72	023176	041002	.WORD	PE134M	;USING RESERVED BITS IN BSEL7	
73	023200	041034	.WORD	PE136M	;COMMON POOL ERROR	
74	023202	041056	.WORD	PE140M	;COMMON POOL QUOTA ERROR	
75	023204	041075	.WORD	PE142M	;SPARE	
76	023206	041103	.WORD	PE144M	;SPARE	
77						
78	023210	041111	CONO3S:	.WORD	BUFTSM	;BUFFER TOO SMALL
79	023212	041132		.WORD	NOEXM	;NONESTANT MEM
80	023214	041150		.WORD	DISCON	;DISCON MESSAGE
81	023216	041162		.WORD	QUEOM	;QUEOVER M.
82	023220	041175		.WORD	CAPLOS	;CARRIER LOSS
83						
84						
85						
86						
87						
88	023222	032006	DLLIND:	.WORD	DPM	
89	023224	032011		.WORD	DUM	
90	023226	032014		.WORD	DLM	
91	023230	032017		.WORD	DGM	
92	023232	032022		.WORD	DAM	
93	023234	032025		.WORD	DUPM	
94	023236	032031		.WORD	DMCM	
95	023240	032035		.WORD	DNM	
96	023242	032040		.WORD	DLVM	
97	023244	032044		.WORD	DMPM	
98	023246	032050		.WORD	DTM	
99	023250	032054		.WORD	DVM	
100	023252	032057		.WORD	DZM	
101	023254	032062		.WORD	UNKM	
102	023256	032072		.WORD	KDPM	
103	023260	032076		.WORD	KDZM	
104	023262	032102		.WORD	KLM	
105	023264	032105		.WORD	DMVM	
106						
107						
120						
131						

;:;FOLLOWING TABLE USED IN DOWNLINE LOAD ROUTINE.  
 ;:;CONTAINS POINTERS TO ASCII DEVICE DESCRIPTIONS  
 ;REV B EC

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
24  
25  
26  
32  
33  
34  
35  
36  
37  
38  
39  
46  
47  
48

.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 PPROGRAM  
;

DEVTYP <DMP OR DMV 11>

L\$DVTYP::  
.ASCIZ /DMP OR DMV 11/

023266			
023266	104	115	120
023271	040	117	122
023274	040	104	115
023277	126	040	061
023302	061	000	

.EVEN

.SBTTL PROGRAM IDENTIFICATION  
; TEST DESCRIPTION  
;

DESCRIPT <CZCLMCO DMP DMV 11 DATA COMM. LINK TEST>

L\$DESC::  
.ASCIZ /CZCLMCO DMP DMV-11

DATA COMM. LINK TEST/  
ST/

023304	103	132	103
023304			
023304			
023307	114	115	103
023312	060	040	104
023315	115	120	040
023320	104	115	126
023323	055	061	061
023326	040	104	101
023331	124	101	040
023334	103	117	115
023337	115	056	040
023342	114	111	116
023345	113	040	124
023350	105	123	124
023353	000		

.EVEN

.EVEN

Line	Address	Offset	Symbol	Format	Description
1			.SBTTL		GLOBAL FORMAT STATEMENTS, MESSAGES, AND ASCII INFO
2			.NLIST	BEX	
3					
4	023354	104	114	CLI#PM: .ASCIZ	/DCLT>/
5	023362	045	116	CLIERM: .ASCIZ	/#N#A?ILL CMD-BAD SYNTAX?/
6	023412	045	116	CLINUF: .ASCIZ	/#N#A?INCMPLTE CMD?/
7	023435	045	116	CLINBG: .ASCIZ	/#N#A?NUM TOO BIG?/
8	023457	045	116	CLIBRX: .ASCIZ	/#N#A?BAD RADIX?/
9	023477	045	116	CLIBDL: .ASCIZ	/#N#A?"LOOP" VALID ONLY IN ACTIVE?/
10	023541	045	116	CLINPS: .ASCIZ	/#N#A?"ECHO" VALID ONLY IN PASSIVE?/
11	023604	045	116	CLIBCR: .ASCIZ	/#N#A?ILL CHR- "A-Z,0-9,SP,TAB ONLY?/
12	023651	045	116	CLISEO: .ASCIZ	/#N#A?"SIZE=0" NOT VALID?/
13	023702	045	116	CLIPPE: .ASCIZ	/#N#A?TRIB CMDS ILLEGAL IN PT PT MODE?/
14	023750	045	116	CLIPW: .ASCIZ	/#N#A?TRANSMIT & EXPECT LIST MUST BE IDENTICAL FOR LOOP?/
15	024040	045	116	HLP0: .ASCIZ	/#N#ATHIS IS DCLT. TYPE "H" OR "?" FOR DETAILS/
16	024116	045	116	HLPF: .ASCIZ	/#N#T/
17	024123	104	114	HLP1: .ASCIZ	/DCLT CMDS:/
18	024136	040	114	HLP2: .ASCII	/ CLEAR OR SHOW EXPECTLIST OR TRANSMITLIST/<15><12>
19	024212	040	120		/ PRINT OR EXIT/<15><12>
20	024232	040	104		? DUMP START-END/B?
21	024254	040	124	HLP2B: .ASCIZ	? TRIB SHOW, TRIB ESTABLISH=, /W,N(D)..OR TRIB KILL=N,ALL?
22	024344	015	012	HLP2C: .ASCIZ	<15><12>/ WHERE W=INDICATES WRITE POLL PARAMS/
23	024417	040	123	HLP3: .ASCIZ	? SET EXPECTMSG OR TRANSMITMSG=TYPE/SIZE=N OR /COPY=N?
24	024504	040	123	HLP3A: .ASCIZ	/ SET EXPECT=TRANSMIT/
25	024531	040	040	HLP4: .ASCIZ	? TYPE=ONES,ZEROES,1ALT,0ALT,ITEP,CCITT,ALPHA?
26	024610	040	040	HLP4A: .ASCIZ	/ OR "OPR SPCD=A-Z,SP,TAB,0-9 IN QUOTES"/
27	024666	040	122	HLP5: .ASCIZ	? RUN MODE=HTYP/LOOP=LTYP/CHECK,STATUS,ECHO,MODEM PASS=N?
28	024756	040	040	HLP6: .ASCII	/ HTYP=TRAN,REC,ACT,PAS,TAL,LIS,DOWN/<15><12>
29	025025	040	040		/ LTYP=INT,CAB,LOC,REM/
30	025055	116	105	EQUQ: .ASCIZ	/NEW POLL PARAMETERS (WORD)=/
31	025111	116	105	EQUQ1: .ASCIZ	/NEW POLL PARAMETERS (BYTE LOW)=/
32	025151	116	105	EQUQ2: .ASCIZ	/NEW POLL PARAMETERS (BYTE HI)=/
33	025210	123	101	DLQ1: .ASCIZ	/SATELLITE PASSWORD=
34	025234	045	116	POLPM: .ASCIZ	/#N#S#APOLL PARAMETERS FOR TRIB #D5/
35	025301	045	116	POLPM3: .ASCIZ	/#N#AGLOBAL POLL PARAMETERS/
36	025334	122	120	CLI#RP: .ASCIZ	/RPT>/
37	025341	045	116	RHLP0: .ASCIZ	/#N#ATYPE "H" OR "?" FOR HELP!/
38	025377	104	103	RHLP1: .ASCIZ	/DCLT REPORT CMDS:/
39	025421	040	105	RHLP2: .ASCIZ	/ EXIT OR LOG/
40	025436	040	124	RHLP3: .ASCIZ	? TSS NNN(D)/SW OR GSS/SW?
41	025470	040	040	RHLP4: .ASCIZ	? WHERE /SW= /FULL,/ERROR,/OFFSET=NN(O)?
42	025544	045	116	RPTIV: .ASCIZ	/#N#A?TRIB STATUS OFFSET=#02#A#TOO BIG?/
43	025613	045	116	SHMSG: .ASCIZ	?#N#AMSG: TYPE=#T#A/SIZE=#D3?
44	025647	132	105	SHTYP0: .ASCIZ	/ZEROES/
45	025656	117	116	SHTYP1: .ASCIZ	/ONES/
46	025663	061	101	SHTYP2: .ASCIZ	/1ALT/
47	025670	060	101	SHTYP3: .ASCIZ	/0ALT/
48	025675	103	103	SHTYP4: .ASCIZ	/CCITT/
49	025703	111	124	SHTYP5: .ASCIZ	/ITEP/
50	025710	101	114	SHTYP6: .ASCIZ	/ALPHA/
51	025716	117	120	SHTYP7: .ASCIZ	/OPR SPEC/
52	025727	045	110	SHTRE: .ASCIZ	\#N#ATRIB ADDRESS LIST IS EMPTY\
53	025766	045	116	SHTRH: .ASCIZ	\#N#ATRIB ADDRESS LIST:#N\
54	026017	045	104	SHTAP: .ASCIZ	\#D3#A, \
55	026027	045	116	SHTFL: .ASCIZ	\#N#ATRIB ADDRESS LIST FULL ADDRESS= #23#A NOT ADDED\
56	026115	045	116	SHTUN: .ASCIZ	\#N#A?TRIB ADDRESS= #23#A IS NOT UNIQUE?\
57	026165	045	116	SHTNF: .ASCIZ	\#N#A?TRIB ADDRESS= #23#A NOT FOUND?\

```

58 026231      045      116      045 SHTLP:  .ASCIZ  \#N#A?CABLE,LOC,REM LOOP NOT VALID IN "MULTIPT MODE"?
59 026316      045      116      045 SHTLPA: .ASCIZ  /#N#A?TRIBS MUST BE ESTABLISHED TO EXECUTE?/
60 026371      045      116      045 SHTLPB: .ASCIZ  /#N#A?TRIB STATION CANNOT DO LOOP?/
61 026433      045      116      045 SHTLPC: .ASCIZ  /#N#A?ONLY ONE TRIB (TRIB ADDR 1) ALLOWED/
62 026504      045      116      045 SHTLPD: .ASCIZ  /#N#S5#AFOR LOOP IN MULTIPOINT?/
63 026543      045      116      045 SHTIV:  .ASCIZ  \#N#A?TRIB ADDRESS= #Z3#A INVALID?\
64 026605      045      116      045 SHTBR:  .ASCIZ  /#N#ARX BUFFER NOT BIG ENOUGH#N#A?TOO MANY TRIBS OR MSGS/
65 026674      122      105      103 MOO:    .ASCIZ  /RECEIVE/
66 026704      124      122      101 HO1:    .ASCIZ  /TRANSMIT/
67 026715      120      101      123 MO2:    .ASCIZ  /PASSIVE/
68 026725      101      103      124 MO3:    .ASCIZ  /ACTIVE/
69 026734      104      117      127 MO4:    .ASCIZ  /DOWNLINELOAD/
70 026751      124      101      114 MO5:    .ASCIZ  /TALK/
71 026756      114      111      123 MO6:    .ASCIZ  /LISTEN/
72 026765      000
73 026766      057      114      117 LP00:   .ASCIZ  ?/LOOP=?
74 026775      111      116      124 LP1:    .ASCIZ  ?INTERNAL?
75 027006      103      101      102 LP2:    .ASCIZ  ?CABLE?
76 027014      114      117      103 LP3:    .ASCIZ  ?LOCALMODEM?
77 027027      122      105      115 LP4:    .ASCIZ  ?REMOTEMODEM?
78 027043      116      117
79 027045      123      124      101 PST:    .ASCIZ  /STATUS/
80 027054      116      117
81 027056      103      110      105 PCK:    .ASCIZ  /CHECK/
82 027064      116      117
83 027066      105      103      110 PEC:    .ASCIZ  /ECHO/
84 027073      116      117
85 027075      115      117      104 PMS:    .ASCIZ  /MODEM/
96
97 027103      045      116      045 LISP:   .ASCIZ  /#N#ALIS>/
98 027114      124      114      113 OPRM:   .ASCIZ  /TLK>/
99 027121      124      110      111 L5G60: .ASCIZ  /THIS A 50. OR 60. HZ. LSI-11:/
100
101
102
103
104
105
106
107 027160      045      116      045 DLLCM:  .ASCIZ  /#N#ADOWN LINE LOAD COMPLETED SUCCESSFULLY/
108
109 027232      045      116      045 BDCLK:  .ASCIZ  /#N#ACLOCK NOT FOUND/
110 027256      045      116      045 NOCLK:  .ASCIZ  /#N#ABAD CLOCK - PROGRAM WILL HANG ON 'TIMEOUT'!!/
111 027337      115      101      130 TAJEX:  .ASCIZ  /MAX. CHAR. MSG COUNT EXCEEDED /
112 027377      102      125      106 BUFEX:  .ASCIZ  /BUFFER FULL -/
113 027415      045      116      045 MSGTRN: .ASCIZ  /#N#T#A MSG. NOT BUILT !!/
114 027446      045      116      045 MSGTRU: .ASCIZ  /#N#ACHAR. COUNT EXCEEDS BUFF LIMIT   MSG TRUNCATED/
115 027531      045      116      045 SHFO:   .ASCIZ  ?#N#S5#AMODE=#T#T#A/PASS=#Z5?
116
122
123 027567      045      116      045 SHF1:   .ASCIZ  ?#N#S5#S5#S5#A/#T#A/#T#A/#T#A/#T?
124 027627      045      123      065 EFM2:   .ASCIZ  /#S5#ATOTAL MISMATCHES IN MSG = #D5/
125 027672      045      116      045 PCPM:   .ASCIZ  /#N#S3#ACALLED FROM PC=#D6/
126 027724      045      123      065 EFM11:  .ASCIZ  /#S5#ACOMPARE COUNT=#D5#S3#ARECEIVE COUNT=#D5/
127
128
129

```

; FORMAT STATEMENTS USED IN PRINT CALLS

; EVENT DESCRIPTION MESSAGES







Line	Address	Mode	Slot	Label	Format	Description
1				.SBTTL		ASCII FORMATS FOR TSS AND GSS SLOTS
2						
3	032112	045	116	045 TSS0A:	.ASCIZ	/#N#06#S2#ATRIB STATUS FLAGS/
4	032146	045	116	045 TSS1A:	.ASCIZ	/#N#03#S5#ANAK REASON#N#03#S5#ATRIB ADDR/
5	032216	045	116	045 TSS2A:	.ASCIZ	/#N#06#S2#APOLL STATUS FLAGS/
6	032252	045	116	045 TSS3A:	.ASCIZ	/#N#03#S5#APOLL RATE#N#03#S5#APOLL PRIORITY/
7	032325	045	116	045 TSS4A:	.ASCIZ	/#N#03#S5#ANAN#N#03#S5#AMAX MSG COUNTER/
8	032373	045	116	045 TSS5A:	.ASCIZ	/#N#03#S5#ACOMM POOL QUOTA#N#03#S5#ARX THRESH ERRS/
9	032455	045	116	045 TSS6A:	.ASCIZ	/#N#03#S5#ATX THRESH. ERRS #N#03#S5#ASELECT THRESH. ERRS/
10	032545	045	116	045 TSS7A:	.ASCIZ	/#N#06#S2#ADATA MSGS. TX'MITTD/
11	032603	045	116	045 TSS10A:	.ASCIZ	/#N#06#S2#ADATA MSGS. RX'CVD/
12	032637	045	116	045 TSS11A:	.ASCIZ	/#N#06#S2#ASELECTION INTERVALS/
13	032675	045	116	045 TSS12A:	.ASCIZ	/#N#03#S5#ADATA ERRORS OUT#N#S8#ABCC #01#A BCC #01#A REP #01/
14	032772	045	116	045 TSS13A:	.ASCIZ	/#N#03#S5#ADATA ERRORS IN#N#S8#ABCC #01#A BCC #01#A REP #01/
15	033066	045	116	045 TSS14A:	.ASCIZ	/#N#03#S5#ALOCAL BUFFER ERRS#N#S8#A TU #01#A TS #01/
16	033151	045	116	045 TSS15A:	.ASCIZ	/#N#03#S5#AREMOTE BUFFER ERRS#N#S8#A TU #01#A TS #01/
17	033235	045	116	045 TSS16A:	.ASCIZ	/#N#03#S5#ASELECTION T-0#N#S8#A NRTS #01#A IRTS #01/
18	033320	045	116	045 TSS17A:	.ASCIZ	/#N#03#S5#ALOCAL REPLY T-0#N#03#S5#AREMOTE PEPLY T 0/
19	033404	045	116	045 TSS20A:	.ASCIZ	/#N#03#S5#AHIGHEST MSG # TX'D#N#03#S5#AHIGHEST MSG # ACK D/
20	033476	045	116	045 TSS21A:	.ASCIZ	/#N#03#S5#ANEXT MSG # TO TX#N#03#S5#ATPTR ADDR OF LNKBK/
21	033565	045	116	045 TSS22A:	.ASCIZ	/#N#03#S5#ALAST MSG # TX'D#N#03#S5#AXPTR ADDR OF LNKBK/
22	033653	045	116	045 TSS23A:	.ASCIZ	/#N#03#S5#ACTL X REPLY T-0#N#03#S5#ASTRT OF TX BUFF Q/
23	033740	045	116	045 TSS24A:	.ASCIZ	/#N#03#S5#AEND OF TX BUFF Q#N#03#S5#AHIGHEST MSG # RX'D/
24	034027	045	116	045 TSS25A:	.ASCIZ	/#N#03#S5#ASTRT OF RX BUFF Q#N#03#S5#AEND OF RX BUFF Q/
25	034114	045	116	045 TSS26A:	.ASCIZ	/#N#06#S2#ATX DELAY TIMER/
26	034145	045	116	045 TSS27A:	.ASCIZ	/#N#03#S5#AND DATA MSG COUNTER#N#03#S5#AT 0 COUNTER/
27	034230	045	116	045 TSS30A:	.ASCII	/#N#06#S2#A/
28	034242	120	122	105 TSS30AA:	.ASCIZ	/PRESET VALUE FOR TX DELAY TIMER/
29	034302	045	116	045 TSS31A:	.ASCIZ	/#N#03#S5#AQ VAL FOR ACT#N#03#S5#AR VAL FOR ACT/
30	034361	045	116	045 TSS32A:	.ASCIZ	/#N#03#S5#AQ VAL FOR INACT#N#03#S5#AR VAL FOR INACT/
31	034444	045	116	045 TSS33A:	.ASCIZ	/#N#03#S5#AQ VAL FOR UNRSP#N#03#S5#AR VAL FOR UNRSP/
32	034527	045	116	045 TSS34A:	.ASCIZ	/#N#03#S5#ADM TO INACT#N#03#S5#A# T 0 TO UNRSP/
33	034606	045	116	045 TSS35A:	.ASCIZ	/#N#03#S5#A# T-0 TO DEAD#N#03#S5#AMAX MSG COUNT/
34	034665	045	116	045 TSS36A:	.ASCIZ	/#N#06#S2#ASELECTION INTERVAL TIMING COUNT/
35	034737	045	116	045 TSS37A:	.ASCIZ	/#N#06#S2#ABABBLING TRIB TIMING COUNT/
36	035004	045	116	045 GSS0A:	.ASCIZ	/#N#03#S5#APOLPTR#N#03#S5#ARCVPTR/
37	035045	045	116	045 GSS1A:	.ASCIZ	/#N#03#S5#AXMTPTR#N#03#S5#ATSP/
38	035103	045	116	045 GSS2A:	.ASCIZ	/#N#03#S5#ANASP#N#03#S5#ABUFPTR/
39	035142	045	116	045 GSS3A:	.ASCIZ	/#N#03#S5#AS-OF#N#03#S5#AE-OF/
40	035177	045	116	045 GSS4A:	.ASCIZ	/#N#03#S5#AS-OQ#N#03#S5#AE-OQ/
41	035234	045	116	045 GSS5A:	.ASCIZ	/#N#03#S5#AS-OC#N#03#S5#AE-OC/
42	035271	045	116	045 GSS6A:	.ASCIZ	/#N#03#S5#ATIMER STATUS#N#03#S5#AS R TIMER [L]/
43	035347	045	116	045 GSS7A:	.ASCIZ	/#N#03#S5#AS-R TIME [H]#N#03#S5#AB CW TIME [L]/
44	035425	045	116	045 GSS10A:	.ASCIZ	/#N#03#S5#AB-CW TIME [H]#N#03#S5#ARPM CNTR/
45	035477	045	116	045 GSS11A:	.ASCIZ	/#N#06#S2#AACTIM/
46	035517	045	116	045 GSS12A:	.ASCIZ	/#N#03#S5#AMODE#N#03#S5#AMODE/
47	035555	045	116	045 GSS13A:	.ASCIZ	/#N#03#S5#AALT SW#N#03#S5#AXMTQRT/
48	035616	045	116	045 GSS14A:	.ASCIZ	/#N#06#S2#ARTNADD/
49	035637	045	116	045 GSS15A:	.ASCIZ	/#N#03#S5#AREMOTE STA ERRS#N#S8#AOVRN #01#A MFE #01#A SEL #01#A STR #01/
50	035747	045	116	045 GSS16A:	.ASCIZ	/#N#03#S5#ALOCAL STA ERRS#N#S8#AOVRN #01#A MFE #01#A UNDR #01#A OVR #01/
51						
52	036057	045	116	045 GSS17A:	.ASCIZ	/#N#03#S5#AGBL HDR BCC#N#03#S5#AMAIN DATA BCC ERR/
53	036141	045	116	045 GSS20A:	.ASCIZ	/#N#03#S5#ATX HDR 1#N#03#S5#ATX HDR 2/
54	036206	045	116	045 GSS21A:	.ASCIZ	/#N#03#S5#ATX HDR 3#N#03#S5#ATX HDR 4/
55	036253	045	116	045 GSS22A:	.ASCIZ	/#N#03#S5#ATX HDR 5#N#03#S5#ATX HDR 6/
56	036320	045	116	045 GSS23A:	.ASCIZ	/#N#03#S5#ARX HDR 1#N#03#S5#ARX HDR 2/
57	036365	045	116	045 GSS24A:	.ASCIZ	/#N#03#S5#ARX HDR 3#N#03#S5#ARX HDR 4/

58	036432	045	116	045	GSS25A:	.ASCIZ	/#N#03#S5#ARX HDR 5#N#03#S5#ARX HDR 6/
59	036477	045	116	045	GSS26A:	.ASCIZ	/#N#06#S2#AR TIMER/
60	036521	045	116	045	GSS27A:	.ASCIZ	/#N#06#S2#AD TIMER/
61	036543	045	116	045	GSS30A:	.ASCIZ	/#N#06#S2#APOLL DELAY TIMER/
62	036576	045	116	045	GSS31A:	.ASCIZ	/#N#03#S5#APOLL UPDATE PTR#N#03#S5#ADEAD SCAN/
63	036553	045	116	045	GSS32A:	.ASCIZ	/#N#03#S5#ACARRIER LOSS TIM#N#03#S5#AUSART HANG CTR/
64	036736	045	116	045	GSS33A:	.ASCIZ	/#N#06#S2#ANUM SYNC/
65	036761	045	116	045	GSS34A:	.ASCIZ	/#N#06#S2#ACARRIER WAIT TIMER COUNTER/
66	037026	045	116	045	GSS35A:	.ASCIZ	/#N#06#S2#ADELTA T/
67	037050	045	116	045	GSS36A:	.ASCIZ	/#N#06#S2#ADEAD T/
68	037071	045	116	045	GSS37A:	.ASCIZ	/#N#06#S2#APOLL DELAY/
69							
88							
89							;DEVICE ERROR MESSAGES
90	037116	104	105	126	DVEM0:	.ASCII	/DEVICE DID NOT RETURN RUN BIT/
91	037153	015	012	040		.ASCIZ	<15><12>/ SEL0 SEL2/
92							
93	037177	106	101	111	DVEM1:	.ASCII	/FAILURE IN MICRO DIAGNOSTICS/
94	037233	015	012	040		.ASCIZ	<15><12>/ SEL0 SEL6/
95							
96	037257	124	111	115	DVEM2:	.ASCII	/TIME OUT WAITING FOR TX OR RX TO COMPLETE/
97	037330	015	012	040		.ASCIZ	<15><12>/ SEL0 SEL2/
98							
99	037354	124	111	115	DVEM3:	.ASCII	/TIME OUT WAITING FOR RDI/
100	037404	015	012	040		.ASCIZ	<15><12>/ SEL0 SEL2/
101							
102	037430	103	117	116	DVEM4:	.ASCII	/CONTROL OR INFORMATION OUT ERROR/
103	037470	015	012	040		.ASCIZ	<15><12>/ SEL2 SEL6/
104							
105	037514	111	114	114	DVEM5:	.ASCII	/ILLEGAL TRANSMIT COMPLETE/
106	037545	015	012	040		.ASCIZ	<15><12>/ SEL4 SEL6/
107							
108	037571	111	114	114	DVEM6:	.ASCII	/ILLEGAL RECEIVE COMPLETE/
109	037621	015	012	040		.ASCIZ	<15><12>/ SEL4 SEL6/
110							
111	037645	121	125	105	DVEM7:	.ASCII	/QUE OVERFLOW BUFFER COMPLETE/
112	037701	015	012	040		.ASCIZ	<15><12>/ SEL4 SEL6/
113							
114	037725	122	114	104	DVEM8:	.ASCII	/RLD OR MODE ENABLE OF PASSWORD SW NOT SET/
115	037776	015	012	040		.ASCIZ	<15><12>/ SEL0 SEL2/
116							
117	040022	040	104	117	DLLAB:	.ASCII	/DOWN LINE LOAD ABORTED/
118	040051	015	012	040		.ASCIZ	<15><12>/ RXBUF TXBUF /
119							
120							
121	040076	123	105	114	SLTHEM:	.ASCIZ	/SELECT THRESHOLD/
122	040117	123	124	101	STRCM:	.ASCIZ	/START RXD IN RUN/
123	040140	115	101	111	MARM:	.ASCIZ	/MAINT RXD IN RUN/
124	040161	115	101	111	MARM:	.ASCIZ	/MAINT RXD IN HALT/
125	040203	123	124	101	STRMM:	.ASCIZ	/START RXD IN MAINT/
126	040226	122	111	116	RIM:	.ASCIZ	/RING DETECTED/
127	040245	104	105	101	DEADTM:	.ASCIZ	/DEAD TRIB/
128	040257	122	125	116	RUSM:	.ASCIZ	/RUN STATE ERR/
129	040275	102	101	102	BABTM:	.ASCIZ	/BABLING TRIB/
130	040312	123	124	122	STREAM:	.ASCIZ	/STREAMING TRIB/
131	040331	116	117	040	PE100M:	.ASCIZ	/NO MODE DEF/
132	040345	111	114	114	PE102M:	.ASCIZ	/ILLEGAL TYPE CODE/

CZCLMCO SMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 41 2  
 ASCII FORMATS FOR TSS AND GSS SLOTS

133	040367	111	114	114	PE104M:	.ASCIZ	/ILLEGAL MODE CHANGE/
134	040413	103	117	116	PE106M:	.ASCIZ	/CONTROL IN TO UNES. TRIB/
135	040444	103	117	115	PE110M:	.ASCIZ	/COMMAND TO TRIB 0/
136	040466	103	117	115	PE112M:	.ASCIZ	/COMMAND TO UNHALTED TRIB/
137	040517	115	101	130	PE114M:	.ASCIZ	/MAX TRIBS EXCEEDED/
138	040542	105	123	124	PE116M:	.ASCIZ	/ESTB TO ALREADY ESTABLISHED/
139	040576	111	114	114	PE120M:	.ASCIZ	/ILLEGAL REQUEST KEY/
140	040622	101	123	123	PE122M:	.ASCIZ	/ASSIGN BUFF UNEST. TRIB/
141	040652	101	123	123	PE124M:	.ASCIZ	/ASSIGN BUFF HALTD TRIB/
142	040701	101	123	123	PE126M:	.ASCIZ	/ASSIGN BUFF BYTE CNT 0/
143	040730	101	123	123	PE130M:	.ASCIZ	/ASSIGN TX BUFF TRIB 0/
144	040756	122	040	117	PE132M:	.ASCIZ	/R OR W RESERVED TSS/
145	041002	125	123	105	PE134M:	.ASCIZ	/USE RESERVED BIT IN BSEL7/
146	041034	103	117	115	PE136M:	.ASCIZ	/COMMON POOL ERROR/
147	041056	121	125	117	PE140M:	.ASCIZ	/QUOTA OVERFLOW/
148	041075				TXTHEM:		
149	041075				RXTHEM:		
150	041075	123	120	101	PE142M:	.ASCIZ	/SPARE/
151	041103	123	120	101	PE144M:	.ASCIZ	/SPARE/
152							

153	041111	102	125	106	BUFTSM:	.ASCIZ	/BUFFER TOO SMALL/
154	041132	116	117	116	NOEXM:	.ASCIZ	/NON EXIST MEM/
155	041150	104	111	123	DISCON:	.ASCIZ	/DISCONNECT/
156	041162	121	125	105	QUEOM:	.ASCIZ	/QUEUE OVER/
157	041175	103	101	122	CARLOS:	.ASCIZ	/CARRIER LOSS/
158	041212	111	116	106	INFOM:	.ASCIZ	/INFORMATION OUT/
159	041232	124	130	040	TXNC:	.ASCIZ	/TX NOT COMPLETE/
160	041252	122	130	040	RXNC:	.ASCIZ	/RX NOT COMPLETE/
161	041272	123	105	103	RXM1:	.ASCIZ	/SEC REQ ERR WORD 1/
162	041315	123	105	103	RXM2:	.ASCIZ	/SEC REQ ERR WORD 2/

163  
164  
165                    .EVEN  
                      .LIST BEX

1  
15  
26  
27  
35  
36  
37  
38

.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.  
;

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
49

041340  
041340  
041340 005046  
041342 153716 017371  
041346 005046  
041350 153716 017370  
041354 013746 017346  
041360 012746 031324  
041364 012746 000004  
041370 010600  
041372 104414  
041374 062706 000012  
041400  
041400 104423  
041402  
041402  
041402 013746 017360  
041406 012746 027627  
041412 012746 000002  
041416 010600  
041420 104414  
041422 062706 000006  
041426  
041426 104423  
041430  
041430  
041430 013746 017356  
041434 010446  
041436 012746 027724  
041442 012746 000003  
041446 010600  
041450 104414  
041452 062706 000010  
041456  
041456 104423

BGNMSG ERR1

PRINTB @EVTF5A,OFFSET,<B,GOOD>,<B,BAD>

ERR1::

;INDIVIDUAL DATA COMPARE ERROR

CLR (SP)  
BISB BAD,(SP)  
CLR (SP)  
BISB GOOD,(SP)  
MOV OFFSET,(SP)  
MOV @EVTF5A,(SP)  
MOV @4,(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD @12,SP

ENDMSG

L10001:

TRAP C\$MSG

BGNMSG ERR2

PRINTB @EFM2,TEMP4

ERR2::

;TOTAL DATA COMPARE FAILS ERROR

MOV TEMP4,-(SP)  
MOV @EFM2,(SP)  
MOV @2,(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD @6,SP

ENDMSG

L10002:

TRAP C\$MSG

BGNMSG ERR10

PRINTB @EFM11,R4,TEMP3

ERR10::

;LENGTH COMPARISON ERROR

MOV TEMP3,-(SP)  
MOV R4,(SP)  
MOV @EFM11,-(SP)  
MOV @3,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD @10,SP

ENDMSG

L10003:

TRAP C\$MSG

50							
51							
52							
53							
54	041460						
	041460						
55	041460						
	041460	013746	017360				
	041464	013746	017356				
	041470	012746	030660				
	041474	012746	000003				
	041500	010600					
	041502	104414					
	041504	062706	000010				
56	041510						
	041510						
	041510	104423					
57							
58							
59							
60							
61							
62							
63	041512						
	041512						
64	041512						
	041512	013746	017366				
	041516	013746	017360				
	041522	013746	017356				
	041526	012746	030675				
	041532	012746	000004				
	041536	010600					
	041540	104414					
	041542	062706	000012				
65	041546						
	041546						
	041546	104423					
66							
67							
68	041550						
	041550						
69	041550						
	041550	005046					
	041552	153716	066654				
	041556	013746	066652				
	041562	013746	066650				
	041566	012746	030717				
	041572	012746	000004				
	041576	010600					
	041600	104414					
	041602	062706	000012				
70	041606						
	041606						
	041606	104423					
71							
72	041610						
	041610						

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

BGNMSG ERR13

PRINTB @EVT3C,TEMP3,TEMP4

ENDMSG

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

BGNMSG ERR14

PRINTB @EVT3D,TEMP3,TEMP4,CONOTM

ENDMSG

BGNMSG ERR15

PRINTB @EVT3F,RSEL4,RSEL6,<B,RSEL3>

ENDMSG

BGNMSG ERR16

ERR13::

MOV TEMP4,(SP)  
MOV TEMP3,(SP)  
MOV @EVT3C,(SP)  
MOV @3,(SP)  
MOV SP,RO  
TRAP C@PNTB  
ADD @10,SP

L10004:

TRAP C@MSG

ERR14::

MOV CONOTM,(SP)  
MOV TEMP4,(SP)  
MOV TEMP3,(SP)  
MOV @EVT3D,(SP)  
MOV @4,(SP)  
MOV SP,RO  
TRAP C@PNTB  
ADD @12,SP

L10005:

TRAP C@MSG

ERR15::

CLR (SP)  
BISB RSEL3,(SP)  
MOV RSEL6,(SP)  
MOV RSEL4,(SP)  
MOV @EVT3F,(SP)  
MOV @4,(SP)  
MOV SP,RO  
TRAP C@PNTB  
ADD @12,SP

L10006:

TRAP C@MSG

ERR16::

```

73 041610
   041610 005046
   041612 153716 066646
   041616 013746 066642
   041622 013746 066644
   041626 012746 030717
   041632 012746 000004
   041636 010600
   041640 104414
   041642 062706 000012
74 041646
   041646
   041646 104423
75
76 041650
   041650 000167
   041652 177772
77

```

PRINTB @EVTFSF,TSEL4,TSEL6,<B,TSEL3>

ENDMSG

EXIT MSG

```

CIR      (SP)
BISB    TSEL3,(SP)
MOV     TSEL6,(SP)
MOV     TSEL4,(SP)
MOV     @EVTFSF,(SP)
MOV     @4,(SP)
MOV     SP,R0
TRAP   C@PNTB
ADD     @12,SP

```

```

L10007: TRAP   C@MSG

```

```

.WORD   JSJMP
.WORD   L10007 2

```

CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 27 Mar 84 16:24 Page 44  
GLOBAL SUBROUTINES SECTION

```

1      .SBTTI  GLOBAL SUBROUTINES SECTION
2
3
4      ;**
5      ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
6      ; THAT ARE USED IN MORE THAN ONE TEST.
7
84     .SBTTL   CLOCK SETUP SUBROUTINE
85
86     ;**
87     ; FUNCTIONAL DESCRIPTION:
88     ; THIS SUBROUTINE SETS UP THE CLOCK INFORMATION TABLE FOLLOWING A "CLOCK"
89     ; CALL EXECUTED IN THE INITIALIZATION CODE. BUT SINCE THE "CLOCK" CALL
90     ; SAYS NOTHING ABOUT AN LSI 11'S CLOCK, THIS ROUTINE IS ONLY USED IF A
91     ; LINE OR P-CLOCK IS FOUND.
92
93     ;
94     ; INPUTS:
95     ; R1= POINTS TO SUPERVISOR SPACE WHERE CLOCK INFO WAS RETURNED
96     ; R2= POINTS TO "CLK" TABLE WHERE CLOCK INFO WILL BE KEPT
97
98     ; IMPLICIT INPUTS:
99     ; THE SUPERVISOR SPACE WHERE CLOCK INFO WAS RETURNED BY THE "CLOCK" CALL
100
101     ; OUTPUTS:
102     ; "CLKCSR" GETS LOADED WITH THE CLOCK'S CSR ADDRESS
103     ; "CLKBR" GETS LOADED WITH THE CLOCK'S INTERRUPT LEVEL
104     ; "CLKVEC" GETS LOADED WITH THE CLOCK'S INTERRUPT VECTOR
105     ; "CLKHZ" GETS LOADED WITH THE LINE FREQ. (HERTZ RATE) WHICH DETERMINES
106     ; THE NUMBER OF TICKS IN A SECOND
107
108     ; CALLING SEQUENCE:
109     ; JSR      PC,CLKSET          ;CALL CLOCK SETUP WITH R1 & R2 SETUP
110     ;--
111
112     CLKSET:
113     MOV      (R1),.(R2),        ;LOAD CLOCK'S CSR ADDR. INTO "CLKCSR"
114     MOV      (R1),.(R2),        ;LOAD CLOCK'S INT. LEVEL INTO "CLKBR"
115     ASL     (R2),                ;ADJUST THE INT. LEVEL FOR LOADING INTO
116     ASL     (R2),                ; THE PSW WITH A "SETVEC" CALL
117     ASL     (R2),
118     ASL     (R2),
119     ASL     (R2),
120     MOV      (R1),.(R2),        ;LOAD CLOCK'S INT. VECTOR INTO CLKVEC
121     MOV      (R1),.(R2),        ;LOAD CLOCK'S HERTZ RATE INTO 'CLKHZ'
122     RTS     PC
123

```



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56

```
.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, & 3: 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 "3" DECREMENTED BY 1 TICK 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.
:
```

```

33 041700          BGNSRV  CLKINT
34 041700          CLKINT::
35 041700 005077 155532  CLH   @CLKCSR      ;DISABLE THE CLOCK FROM INTERRUPTING
36 041704 005337 017454  DEC   TIMTCK      ;DECREMENT THE # OF TICKS/SEC.
37 041710 001015          BNE   1$          ;GO CHECK TIMERS (1&2-TICKS, 3-SECONDS)
38 041712 013737 017444 017454  MOV   CLKHZ,TIMTCK  ;RESET THE # OF TICKS/SEC.
39 041720 005237 017452          INC   TIMSEC      ;INC # OF SECS-SINCE-START
40 041724 022737 000074 017452  CMP   #60.,TIMSEC  ;SEE IF WE'VE COUNTED 60 SECS. YET
41 041732 001004          BNE   1$          ;IF NOT, GO CHECK TIMERS
42 041734 005237 017450          INC   TIMMIN      ; ELSE INC MINUTES-SINCE-START
43 041740 005037 017452          CLR   TIMSEC      ; AND RESTART SECOND COUNTER
44
45 041744 005737 017456          1$: TST   TIMER1      ;SEE IF TIMER #1, TIMING ANYTHING
46 041750 001402          BEQ   2$          ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
47 041752 005337 017456          DEC   TIMER1      ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
48 041756 005737 017460          2$: TST   TIMER2      ;SEE IF TIMER #2, TIMING ANYTHING
49 041762 001402          BEQ   3$          ; IF=0, NOTHING BEING TIMED CHECK NEXT TIMER
50 041764 005337 017460          DEC   TIMER2      ; ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
51 041770 005737 017462          3$: TST   TIMERS      ;SEE IF TIMER #3, TIMING ANYTHING
52 041774 001406          BEQ   4$          ; IF=0, NOTHING BEING TIMED, LEAVE
53 041776 023737 017444 017454  CMP   CLKHZ,TIMTCK  ;SEE IF A SECOND HAS BFEN COUNTED OFF
54 042004 001002          BNE   4$          ; RR IF NO
55 042006 005337 017462          DEC   TIMERS      ; ELSE DECREMENT THE TIMER VALUE (BY 1 SEC.)
56 042012 013777 017446 155416 4$: MOV   CLKEN,@CLKCSR ;REENABLE THE CLOCK TO INTERRUPT

```

F8

SEQ 96

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22-Mar 84 16:24 Page 45 1  
CLOCK INTERRUPT SERVICE ROUTINE

5 \* 042020  
042020  
042020 000002

ENDSRV

L10010: RTI

CZCLMCO DMP/V-11 DCL T  
EVENT LOG SUBROUTINES

MACRO V05.00 Thursday 22-Mar 84 16:24 Page 46

```

1          .SBITL          EVENT LOG SUBROUTINES
2
3          ;**
4          ; FUNCTIONAL DESCRIPTION:
5          ; THIS SUBROUTINE HAS A DIFFERENT ENTRY POINT
6          ; FOR EACH EVENT TO BE LOGGED AND ALWAYS PRINTS
7          ; THE SHORT "OPERATOR AWAKE" MESSAGE TO CONSOLE THEN LOGS THE
8          ; EVENT TYPE, TIME, AND THE OTHER 3 WORDS OF INFO PASSED TO THE
9          ; SUBROUTINE AT CALLING TIME
10
11         ;
12         ; INPUTS:
13         ; TIMMIN & TIMSEC:          CURRENT VALUE OF "TIME SINCE START"
14         ; TEMP2: WORD #1 OF EVENT LOG INFORMATION (FOR MOST EVENT TYPES)
15         ; TEMP3: WORD #2 OF EVENT LOG INFORMATION
16         ; TEMP4: WORD #3 OF EVENT LOG INFORMATION
17         ; MODS: CURRENT VALUE OF THE MODEM SIGNALS AVAILABLE FROM THE DEVICE
18
19         ;
20         ; OUTPUTS:
21         ; "OPERATOR AWAKE" MESSAGE SENT TO THE CONSOLE
22         ; NEW EVENT LOGGED IN "EVTLOG" (EVENT LOG)
23         ; UPDATED "EVTPTN" (EVENT LOG ENTRY POINTER)
24
25         ;
26         ; SUBORDINATE ROUTINES USED:
27         ; "DVMODS" THE DEVICE SUBROUTINE THAT RETURNS MODEM STATUS IN "MODS"
28         ; (FOR SOME EVENT TYPES)
29
30         ;
31         ; FUNCTIONAL SIDE EFFECTS:
32         ; TEMP: USED TO STORE ADDRESS OF "OPERATOR AWAKE" MESSAGE
33         ; TEMP1: USED TO SETUP THE VALUE OF THE "EVENT TYPE" BYTE FOR LOGGING
34
35         ;
36         ; CALLING SEQUENCE:
37         ; JSR PC,LOGTXQ          ;CALL THE LOG EVENT SUBROUTINE WITH TEMP,TEMP1,
38         ; " " " "          ; TEMP2, TEMP3, AND TEMP4 SETUP
39         ; " " " "
40         ; JSR PC,LOGCMP
41         ;--
42
43         LOGTXQ:
44         042022      012737  031571  017352      MOV     #S1XQ,TEMP1      ;SET UP MSG. TO PRINT
45         042030      012737  000000  017350      MOV     #TXQ,TEMP        ;SET UP EVENT TYPE
46         042036      000522                                BR     LOGS1            ;GO LOG EVENT AND TIME
47
48         LOGTXC:
49         042040      012737  031602  017352      MOV     #STXC,TEMP1     ;SET UP MSG. TO PRINT
50         042046      012737  000002  017350      MOV     #TXC,TEMP       ;SET UP EVENT TYPE
51         042054      000513                                BR     LOGS1            ;GO LOG EVENT AND TIME
52
53         LOGRXQ:
54         042056      012737  031613  017352      MOV     #SRXQ,TEMP1     ;SET UP MSG. TO PRINT
55         042064      012737  000004  017350      MOV     #RXQ,TEMP       ;SET UP EVENT TYPE
56         042072      000504                                BR     LOGS1            ;GO LOG EVENT AND TIME
57
58         LOGRXC:
59         042074      012737  000006  017350      MOV     #RXC,TEMP       ;SET UP EVENT TYPE
60         042102      C00500                                BR     LOGS1            ;GO LOG EVENT AND TIME
61
62         LGDVE:
63         042104      012737  031624  017352      MOV     #SDVE,TEMP1     ;SET UP MSG. TO PRINT

```

CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 22-Mar-84 16.24 Page 46 1  
EVENT LOG SUBROUTINES

```

58 042112 012737 000010 017350      MOV      @DFR,TEMP      ;SET UP EVENT TYPE
59 042120 000511                BR          LOGS3      ;GO LOG EVENT AND TIME
60
61 042122                LOGDVI:
62 042122 012737 031646 017352      MOV      @SDVI,TEMP1   ;SET UP MSG. TO PRINT
63 042130 012737 000012 017350      MOV      @DVI,TEMP     ;SET UP EVENT TYPE
64 042136 113737 017402 017354      MOV      MODTYP,TEMP2
65 042144 113737 017404 017355      MOV      MLTYP,TEMP2+1
66 042152 013737 017412 017356      MOV      RPASS,TEMP3
67 042160 013737 017410 017360      MOV      PARAM,TEMP4
68 042166 000466                BR          LOGS3      ;SET UP EVNT ENTRIES
69                                ;GO LOG EVENT AND TIME
70 042170                LOGCMP:
71 042170 012737 031635 017352      MOV      @SCM,TEMP1   ;SET UP MSG. TO PRINT
72 042176 012737 000014 017350      MOV      @DCK,TEMP     ;SET UP EVENT TYPE
73 042204 000415                BR          LOGS3A
74 042206                LOGCML:
75 042206 012737 031657 017352      MOV      @SCML,TEMP1
76 042214 012737 000020 017350      MOV      @DLE,TEMP
77 042222 000406                BR          LOGS3A    ;SET UP MSG. AND TYPE
78                                ;GO LOG EVENT AND TIME
79 042224 012737 031670 017352      MOV      @SCMD,TEMP1
80 042232 012737 000022 017350      MOV      @DDE,TEMP
81 042240 013737 015756 017362      LOGS3A: MOV      TRIBN,TEMP5
82 042246 000436                BR          LOGS3      ;GO LOG MSG TYPE AND TIME
83
84 042250                LOGEOP:
85 042250 012737 031701 017352      MOV      @SEOP,TEMP1
86 042256 012737 000024 017350      MOV      @EOP,TEMP
87 042264 000427                BR          LOGS3      ;GO LOG MSG TYPE AND TIME
88
89 042266                LOGMSC:
90 042266 012737 031712 017352      MOV      @SMSC,TEMP1
91 042274 012737 000016 017350      MOV      @MSC,TEMP
92 042302 000420                BR          LOGS3
93
94 042304 013746 017310                LOGS1: MOV      ERRCNT, -(SP) ;SAVE CURRENT ERROR COUNT
95 042310 013737 015756 017362      MOV      TRIBN,TEMP5 ;SAVE TRIBN
96 042316 004737 063756                JSR      PC,DVMODS    ;GO GET MODEM STATUS
97 042322 012604                MOV      (SP)+,R4     ;GET SAVED ERRCNT VALUE
98 042324 020437 017310                CMP      R4,ERRCNT    ;WERE ANY ERRORS FOUND
99 042330 001402                BEQ      1$           ;BR IF NONE
100 042332 000137 042552                JMP      LOGEX        ;ELSE, LEAVE WITHOUT LOGGING ANYTHING
101                                ; BUT THE DEVICE ERROR FROM DVMODS
102 042336 013737 020524 017360      1$:  MOV      MODS,TEMP4 ;AND PUT IT IN TEMP4
103
104 042344                LOGS3:
105 042344 022737 000006 017350      CMP      @RXC,TEMP
106 042352 001434                BEQ      LOGS5        ;IF RXC DON'T PRINT
107 042354 032737 000001 017410      BIT      @STATB,PARAM
108 042362 001430                BEQ      LOGS5        ;IF NO STATUS SELECTED
109                                ;GO TO 5
110
111 042364 022737 000010 017300      CMP      @10,LNCNT
112 042372 001012                BNE      LOGS4
113 042374 005037 017300                CLR      LNCNT
114                                ;HAVE WE DONE 10?
115                                ;IF NOT GO TO 4
116                                ;ELSE CLEAR IT

```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 46 2  
 EVENT LOG SUBROUTINES

```

115 042400          PRINTF  #CR          ;ELSE PRINT CR
      042400 012746 031566
      042404 012746 000001
      042410 010600
      042412 104417
      042414 062706 000004
116 042420          LOGS4:
117 042420 005237 017300          INCL  LNCNT          ;INC COUNTER OF # OF AWAKE MSGS
118 042424          PRINTF  TEMP1        ;PRINT OPERATOR AWAKE MSG.
      042424 013746 017352
      042430 012746 000001
      042434 010600
      042436 104417
      042440 062706 000004
119 042444 010346          LOGS5: MOV  R3,-(SP)          ;SAVE R3 ON THE STACK
120 042446 013703 017464          MOV  EVTPTR,R3
121 042452 113723 017350          MOV  TEMP,(R3)+          ;LOG EVENT
122 042456 013737 017444 017350          MOV  CLKHZ,TEMP
123 042464 163737 017454 017350          SUB  TIMTCK,TEMP
124 042472 113723 017350          MOV  TEMP,(R3)+          ;LOG TIME SINCE START
125 042476 113723 017452          MOV  TIMSEC,(R3)+
126 042502 113723 017450          MOV  TIMMIN,(R3)+          ;TICKS,SECS AND MINS.
127 042506 013723 017354          MOV  TEMP2,(R3)+          ;LOG EVNT ENTRY 3
128 042512 013723 017356          MOV  TEMP3,(R3)+          ;LOG EVNT ENTRY 4
129 042516 013723 017360          MOV  TEMP4,(R3)+          ;LOG EVNT ENTRY 5
130 042522 013723 017362          MOV  TEMP5,(R3)+          ;LOG EVNT ENTRY 6
131 042526 020327 020522          CMP  R3,#EVTEND
132 042532 103404          BLO  LOGS2
133
134 042534 012713 177777          MOV  #1,(R3)
135 042540 012703 017466          MOV  #EVTLOG,R3          ;LOG A TABLE END
136 042544 010337 017464          LOGS2: MOV  R3,EVTPTR          ;PUT R3 TO START OF TABLE
137 042550 012603          MOV  (SP)+,R3          ;RESTORE POINTER
138 042552 000207          LOGEX: RTS  PC          ;RESTORE R3
139
140

```

Line	Address	Offset	Value	Instruction	Comment	Register	Value
1				.SBTTL	DUMP EVENT LOG AND BASE TABLE		
2							
3							
4	042554	010246		REPORT: MOV	R2, -(SP)		;SAVE R2,R3,R4 ON THE STACK
5	042556	010346		MOV	R3, (SP)		
6	042560	010446		MOV	R4, (SP)		
7							
8							
9							
10	042562			PRINTF	@RHLPO		
	042562	012746	025341				MOV @RHLPO, (SP)
	042566	012746	000001				MOV @1, (SP)
	042570	010600					MOV SP, R0
	042574	104417					TRAP C\$PNTF
	042576	062706	000004				ADD @4, SP
11	042602	105037	003411	GETRCL: CLR	B P\$GDBD		;CLEAR GOOD BAD FLAG
12	042606	105037	003410	CLR	B P\$NNUF		
13							
14							
15							
16	042612			GMANID	CLI\$RP, CMDBUF, A, 0, 1, 72, ., NO		
	042612	104443					TRAP C\$GMAN
	042614	000406					BR 10000\$
	042616	003130					.WORD CMDBUF
	042620	000142					.WORD T\$CODE
	042622	025334					.WORD CLI\$RP
	042624	000000					.WORD 0
	042626	000001					.WORD T\$LOLIM
	042630	000110					.WORD T\$HILIM
	042632						10000\$:
17	042632	012737	003130	MOV	@CMDBUF, P\$BUFA		
18	042640	012737	044774	MOV	@CLIRT, P\$TREE		
19	042646	012737	044360	MOV	@CLIRAC, P\$ACT		
20	042654	005037	003254	CLR	QUALFG		;CLEAR QUALIFIER FLAG LOCATION
21	042660	004737	047646	JSR	PC, P\$TRV		;GO PARSE COMMAND LINE
22	042664	105737	003411	TSTB	P\$GDBD		;SEE IF PARSED OK OR AN ERROR
23	042670	001412		BEQ	1\$		
24	042672			PRINTF	@CLIERM		
	042672	012746	023362				MOV @CLIERM, (SP)
	042676	012746	000001				MOV @1, (SP)
	042702	010600					MOV SP, R0
	042704	104417					TRAP C\$PNTF
	042706	062706	000004				ADD @4, SP
25	042712	000137	042602	JMP	GETRCL		
26	042716	105737	003410	1\$: TSTB	P\$NNUF		;SEE IF INCOMPLETE COMMAND TYPED
27	042722	001412		BEQ	10\$		
28	042724			PRINTF	@CLINUF		
	042724	012746	023412				MOV @CLINUF, (SP)
	042730	012746	000001				MOV @1, (SP)
	042734	010600					MOV SP, R0
	042736	104417					TRAP C\$PNTF
	042740	062706	000004				ADD @4, SP
29	042744	000137	042602	JMP	GETRCL		
30							
31	042750	023727	003252	10\$: CMP	KEYWD1, @RPTSS		
32	042756	00' 003		BNE	20\$		
33	042760	004737	043006	JSR	PC, RPTSS		;JUMP TO REPORT TSS

CZCLMCC DMP/V-11 DCLT MACRC V05.00 Thursday 22-Mar R4 16:24 Page 47 1  
 DUMP EVENT LOG AND BASE TABLE

```

34 042764 000706          BR      GETRCL      ;IF EQUAL JUMP BACK
35 042766 023727 003252 000002 20$:  CMP      KEYWD1, @RPEXT ;SEE IF EXIT REPORT SECTION
36 042774 001302          BNE      GETRCL
37 042776 012604          ENDALL: MOV      (SP)+, R4 ;RESTORE R4,R3,R2
38 043000 012603          MOV      (SP)+, R3
39 043002 012602          MOV      (SP)+, R2
40 043004 000207          RTS      PC ;RETURN TO CALLING ROUTINE
41
42
43 043006 012737 000046 021024 RPTTSS: MOV      @46, TSSA ;SET KEY UP TO FIRST ERROR
44 043014 005737 015756          TST      TRIBN
45 043020 001003          BNE      RDTSS2 ;BRANCH IF TSS
46 043022 012737 000054 021024          MOV      @54, TSSA ;IF GSS USE 55
47 043030 012737 000057 021022 RDTSS2: MOV      @57, TSSE ;SET UP 57 AS END
48 043036 122737 000105 021026          CMPB    @105, TSSKEY ;IS THIS AN E
49 043044 001422          BEQ      RDTSS ;AND GO READ THEM
50 043046 012737 000037 021024          MOV      @37, TSSA
51 043054 012737 000077 021022          MOV      @77, TSSE ;SET UP LIMITS
52 043062 122737 000106 021026          CMPB    @106, TSSKEY ;IS THIS FULL
53 043070 001410          BEQ      RDTSS ;IF SO READ FULL
54 043072 013737 021026 021024          MOV      TSSKEY, TSSA
55 043100 005337 021024          DEC      TSSA
56 043104 013737 021026 021022          MOV      TSSKEY, TSSE
57
58 043112 005237 021024          RDTSS: INC      TSSA
59 043116 152777 000200 157726          BISB    @RQI, @BSELO ;MAKE RQEST
60 043124 004737 065114          JSR      PC, TOORIO
61 043130 012737 177777 017326          MOV      @-1, TSSFLG ;SET FLAG
62 043136 113777 015756 157714          MOVB    TRIBN, @BSEL3
63 043144 013777 021024 157714          MOV      TSSA, @SEL6
64 043152 112777 000001 157676          MOVB    @01, @BSEL2 ;DO CONTROL IN READ TSS
65 043160 023737 021024 021022          CMP      TSSA, TSSE ;ARE WE DONE
66 043166 001351          BNE      RDTSS ;IF NOT GO BACK FOR MORE
67 043170 152777 000200 157654          BISB    @RQI, @BSELO ;MAKE RQEST
68 043176 004737 065114          JSR      PC, TOORIO
69 043202 113777 015756 157650          MOVB    TRIBN, @BSEL3
70 043210 105077 157652          CLRB    @BSEL6
71 043214 112777 000001 157634          MOVB    @01, @BSEL2 ;DO CONTROL IN [NO REQUEST]
72                                     ;THIS GETS LAST OUTPUT
73 043222 000207          RTS      PC ;RETURN WHEN DONE
74
75
76
77
78 043224 010246          REPLOG: MOV      R2, (SP) ;SAVE R2,R3,R4 ON THE STACK
79 043226 010346          MOV      R3, (SP)
80 043230 010446          MOV      R4, (SP)
81
82 043232 013702 017464          MOV      EVTPT, R2 ;MAKE R2 A POINTER TO EVENT TABLE
83 043236 023727 017466 177777          CMP      EVTLOG, @ 1 ;SEE IF EVENT TABLE IS EMPTY
84 043244 001034          BNE      RPTO ;BR IF NO
85 043246          PRINTS @NULEVT ;IF EMPTY TELL OPERATOR.
      MOV      @NULEVT, (SP)
      MOV      @1, (SP)
      MOV      SP, R0
      TRAP    C$PNTS
      ADD     @4, SP
    
```

```

86 043266 000137 044204          JMP      ENDEVT          ;AND END
87
88 043272 162702 000014          RPT:    SUB      #14,R2  ;NOW POINT BACK TO TOP OF ENTRY U
89                                ;JUST PRINTED
90
91 043276 020227 017466          CMP      R2,#EVTLOG     ;POINTING TO TOP OF EVNT LOG QUEUE?
92 043302 001010          BNE     RPT1           ; BR IF NO
93 043304 012702 020522          MOV      #EVTEND,R2    ;SET R2 TO POINT TO BOTTOM OF LOG
94 043310 026227 177776          CMP      -2(R2),#-1    ;
95 043316 001007          BNE     RPT0           ;IF END OF LOG IS NOT EMPTY
96 043320 000137 044204          JMP      ENDEVT          ;CONTINUE...ELSE EXIT
97
98 043324 020237 017464          RPT1:   CMP      R2,EVTPTR ;ARE WE BACK TO POINTER?
99 043330 001002          BNE     RPT0           ;IF NOT CONTINUE
100 043332 000137 044204         JMP      ENDEVT          ;IF SO EXIT....
101
102 043336 162702 000014          RPT0:   SUB      #14,R2  ;POINT R2 TO START OF ENTRY
103 043342          RPTAA: PRINTS  #EVTFO   ;PRINT EVENT ENTRY HEADER
                                MOV      #EVTFO,(SP)
                                MOV      #1,-(SP)
                                MOV      SP,R0
                                TRAP    C$PNTS
                                ADD     #4,SP
104 043342 012746 030447          MOV      (R2)+,R3      ;PUT EVENT TYPE INTO R3
105 043346 012746 000001          MOV      (R2)+,EVTICK  ;
106 043352 010600          MOV      (R2)+,EVTSEC  ;PUT EVENT TIME (TICKS,SECS,MINS IN TEMP LOC.S)
107 043354 104416          MOV      (R2)+,EVTMIN  ;
108 043356 062706 000004          PRINTS  #EVTFO,EVTMIN,EVTSEC,EVTICK,EVTLST(R3) ;PRINT EVENT TIME AND DESCRIPT.
                                MOV      EVTLST(R3),(SP)
                                MOV      EVTICK,(SP)
                                MOV      EVTSEC,(SP)
                                MOV      EVTMIN,(SP)
                                MOV      #EVTFO,-(SP)
                                MOV      #5,(SP)
                                MOV      SP,R0
                                TRAP    C$PNTS
                                ADD     #14,SP
109 043400 016346 020562          JMP      @RPTDSP(R3)   ;DISPATCH TO DECODING SECTION FOR SPECIFIC TYPE
110
111 043404 013746 020614          RPTTXQ: MOV      (R2)+,EVTADD ;STORE MESSAGE ADDRESS FOR PRINTING
112 043410 013746 020610          MOV      (R2)+,EVTBCT  ;STORE BYTE COUNT FOR PRINTING
113 043414 013746 020612          MOV      (R2)+,R3      ;STORE MODEM STATUS FOR PRINTING
114 043420 012746 030545          JSR      PC,PNTTRB    ;PRINT TRIB NO.
115 043424 012746 000005          PRINTS  #EVTFO,EVTADD,EVTBCT ;PRINT ADDR,BYTE CNT
                                MOV      EVTBCT,(SP)
                                MOV      EVTADD,-(SP)
                                MOV      #EVTFO,(SP)
                                MOV      #3,(SP)
                                MOV      SP,R0
                                TRAP    C$PNTS
                                ADD     #10,SP
116 043430 010600          JSR      PC,RPTMSB    ;GO PRINT MODEM STATUS
117 043432 104416          JMP      RPT          ;GO BACK FOR NEXT EVENT ENTRY
118
119 043434 062706 000014          RPTDER: MOV      (R2)+,EVTTMP ;GET ADDRESS OF DEVICE INFO MESSAGE
120 043436 013746 020620          MOV      (R2)+,DEV1   ;STORE DEVICE REG CONTENTS FOR PRINTING
121 043438 013746 020616          MOV      (R2)+,DEV2
    
```



CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 47 3  
 DUMP EVEN LOG AND BASE TABLE

122	043536	012237	017362	MOV	(R2)+,TEMP5				
123	043542			PRINTS	#EVTF3,EVTMP	;PRINT DEVICE REG CONTENTS.			
	043542	013746	020622				MOV	EVTMP,-(SP)	
	043546	012746	030646				MOV	#EVTF3,-(SP)	
	043552	012746	000002				MOV	#2,-(SP)	
	043556	010600					MOV	SP,R0	
	043560	104416					TRAP	C#PNTS	
	043562	062706	000006				ADD	#6,SP	
124	043566			PRINTS	#EVTF3C,DEV1,DEV2				
	043566	013746	020654				MOV	DEV2,(SP)	
	043572	013746	020652				MOV	DEV1,(SP)	
	043576	012746	030660				MOV	#EVTF3C,-(SP)	
	043602	012746	000003				MOV	#3,-(SP)	
	043606	010600					MOV	SP,R0	
	043610	104416					TRAP	C#PNTS	
	043612	062706	000010				ADD	#10,SP	
125	043616	000137	043272	JMP	RPT	;GO BACK FOR NEXT EVENT ENTRY			
126									
127	043622	005037	020652	RPTDVI:	CLR	DEV1			
128	043626	005037	020654		CLR	DEV2			
129	043632	112237	020652		MOVB	(R2)+,DEV1			
130	043636	112237	020654		MOVB	(R2)+,DEV2			
131	043642	012237	020656		MOV	(R2)+,DEV3			
132	043646	012237	020660		MOV	(R2)+,DEV4			
133	043652	010246			MOV	R2,-(SP)			
134	043654	004737	047344		JSR	PC,SHWOP			
135	043660	012602			MOV	(SP)+,R2			
136	043662	012237	017362		MOV	(R2)+,TEMP5			
137	043666	000137	043272		JMP	RPT			
138	043672	012237	020616	RPTTEOP:	MOV	(R2)+,EVTADD			
139	043676	012237	020620		MOV	(R2)+,EVTBCT			
140	043702	012237	020622		MOV	(R2)+,EVTMP			
141	043706	012237	017362		MOV	(R2)+,TEMP5			
142									
143									
144									
145	043712								
	043712	013746	020620		PRINTS	#EVTF4B,EVTADD,EVTBCT			
	043716	013746	020616				MOV	EVTBCT,(SP)	
	043722	012746	031016				MOV	EVTADD,-(SP)	
	043726	012746	000003				MOV	#EVTF4B,(SP)	
	043732	010600					MOV	#3,(SP)	
	043734	104416					MOV	SP,R0	
	043736	062706	000010				TRAP	C#PNTS	
	043742						ADD	#10,SP	
146	043742			PRINTS	#EVTF44,EVTMP,TEMP5				
	043742	013746	017362				MOV	TEMP5,-(SP)	
	043746	013746	020622				MOV	EVTMP,(SP)	
	043752	012746	031055				MOV	#EVTF44,(SP)	
	043756	012746	000003				MOV	#3,(SP)	
	043762	010600					MOV	SP,R0	
	043764	104416					TRAP	C#PNTS	
	043766	062706	000010				ADD	#10,SP	
147	043772	000137	043272	JMP	RPT	;THEN GO GET NEXT EVENT ENTRY			
148									
149									
150	043776	012237	020616	RPTDDE:	MOV	(R2)+,EVTADD			
151	044002	012237	020620		MOV	(R2)+,EVTBCT			

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 47 4  
 DUMP EVENT LOG AND BASE TABLE

```

152 044006 012237 020622      MOV      (R2)+,EVTTMP      ;STORE TOTAL # OF CMP ERRORS
153 044012 004737 044326      JSR      PC,PNTTRB        ;PRINT TRIB NO.
154 044016      PRINTS   #EVT4,EVTADD,EVTBCT,EVTMP      ;PRINT ADDR, BYTE CNT, # CMP ERRS
                                MOV      EVTMP,-(SP)
                                MOV      EVTBCT,-(SP)
                                MOV      EVTADD,-(SP)
                                MOV      #EVT4,-(SP)
                                MOV      #4,(SP)
                                MOV      SP,R0
                                TRAP    C$PNTS
                                ADD     #12,SP
155 044052 000137 043272      JMP      RPT              ;THEN GO GET NEXT EVENT ENTRY
156
157 044056      RPTDLE:
158 044056 012237 020616      RPTDCK: MOV      (R2)+,EVTADC      ;STORE MSG ADDR FOR PRINT
159 044062 012237 020620      MOV      (R2)+,EVTBCT      ;STORE BYTE COUNT
160 044066 012237 020622      MOV      (R2)+,EVTMP        ;STORE BYTE COUNT COMP
161 044072 004737 044326      JSR      PC,PNTTRB        ;PRINT TRIB NO.
162 044076      PRINTS   #EVT4A,EVTADD,EVTBCT,EVTMP      ;PRINT ADDR,RXBYTES,CMPBYTES.
                                MOV      EVTMP,(SP)
                                MOV      EVTBCT,-(SP)
                                MOV      EVTADD,(SP)
                                MOV      #EVT4A,-(SP)
                                MOV      #4,(SP)
                                MOV      SP,R0
                                TRAP    C$PNTS
                                ADD     #12,SP
163
164 044132 000137 043272      JMP      RPT              ;THEN GO GET NEXT EVENT ENTRY
165
166
167 044136      RPTMSC:
168
169 044136 012203      MOV      (R2)+,R3          ;PUT OLD MODEM STATUS IN R3 FOR PRINTING
170 044140 004737 044214      JSR      PC,RPTMSB        ;GO PRINT OLD MODEM STATUS
171 044144      PRINTS   #EVMOCG          ;GO PRINT "CHANGED TO:"
                                MOV      #EVMOCG,-(SP)
                                MOV      #1,(SP)
                                MOV      SP,R0
                                TRAP    C$PNTS
                                ADD     #4,SP
172 044164 012203      MOV      (R2)+,R3          ;PUT NEW MODEM STATUS IN R3 FOR PRINTING
173 044166 004737 044214      JSR      PC,RPTMSB        ;GO PRINT NEW MODEM STATUS
174 044172 012203      MOV      (R2)+,R3          ;POP NULL WORD FROM ENTRY OUT OF LOG
175 044174 012237 017362      MOV      (R2)+,TEMP5      ;DUMMY MOVE
176 044200 000137 043272      JMP      RPT              ;THEN GO GET NEXT EVENT
177
178 044204      ENDEVT:
179 044204 012604      MOV      (SP)+,R4          ;RETURN TO CALLER AFTER REG RESTORE
180 044206 012603      MOV      (SP)+,R3          ;RESTORE R4,R3,R2
181 044210 012602      MOV      (SP)+,R2
182 044212 000207      RTS      PC              ;RETURN TO CALLING ROUTINE
183
184
185      ;REPORT MODEM STATUS SUBROUTINE
186      ;
187      PART OF STATISTICAL REPORTING (DUMPING EVENT LOG)
    
```

188	044214			RPTMSB: PRINTS	#EVMOH0		;PRINT MODEM STATUS HEADER	
	044214	012746	031433				MOV	#EVMOH0, (SP)
	044220	012746	000001				MOV	#1, (SP)
	044224	010600					MOV	SP,RO
	044226	104416					TRAP	C#PNTS
	044230	062706	000004				ADD	#4,SP
189	044234	012704	020526		MOV	#MOBITS,R4	;MAKE R4 A POINTER TO MODEM SIG. BIT DEF. TABLE	
190	044240	012705	020544		MOV	#MOMSGS,R5	;MAKE R5 A POINTER TO MODEM MSG. POSITION TABLE	
191	044244	005714		6:	TST	(R4)	;SEE IF BIT AVAILABLE FROM DEVICE	
192	044246	001004			BNE	7:	;BR IF THAT MODEM SIG. AVAILABLE	
193	044250	112735	000130		MOVB	#'X,B(R5).	;ELSE PUT "X" IN REPORT IF SIGNAL NOT AVAILABLE	
194	044254	005724			TST	(R4).	;BUMP R4 TO POINT TO NEXT BIT DEFINITION	
195	044256	000407			BR	9:	;GO SEE IF CHECKED ALL MODEM SIGNALS	
196	044260	032403		7:	BIT	(R4).,R3	;IF THERE, SEE IF THAT BIT IN DEVICE S ENTRY-1	
197	044262	001403			BEG	8:	;BR IF BIT (SIGNAL) VALUE =0	
198	044264	112735	000061		MOVB	#'1,B(R5).	;IF=1, PUT "1" IN REPORT MESSAGE	
199	044270	000402			BR	9:	;GO SEE IF ALL MODEM SIGNALS CHECKED	
200	044272	112735	000060	8:	MOVB	#'0,B(R5).	;IF BIT(SIGNAL)=0, PUT "0" IN REPORT MESSAGE	
201	044276	020427	020544	9:	CMP	R4,#MOBITE	;SEE IF ALL BITS(SIGNALS) CHECKED	
202	044302	002760			BLT	6:	;LOOP UNTIL ALL SIGNALS(BITS) CHECKED	
203	044304				PRINTS	#EVMOST	;THEN PRINT MODEM SIGNAL VALUE MESSAGE	
	044304	012746	031513				MOV	#EVMOST, (SP)
	044310	012746	000001				MOV	#1, (SP)
	044314	010600					MOV	SP,RO
	044316	104416					TRAP	C#PNTS
	044320	062706	000004				ADD	#4,SP
204	044324	000207			RTS	PC	;RETURN TO EVENT DECODING	
205								
206								
207								
208	044326	012237	017362					
209	044332			PNTTRB: MOV	(R2).,TEMP5			
	044332	013746	017362		PRINTS	#EVI6,TEMP5	;PRINT TRIB NUMBER.	
	044336	012746	030756				MOV	TEMP5, (SP)
	044342	012746	000002				MOV	#EVI6, (SP)
	044346	010600					MOV	#2, (SP)
	044350	104416					MOV	SP,RO
	044352	062706	000006				TRAP	C#PNTS
210	044356	000207			RTS	PC	;RETURN TO EVENT	

```

1
2 044360
3 044360 006302
4 044362 016202 044376
5 044366 062702 044376
6 044372 004712
7 044374 000207
8
9 044376 000034
10 044400 000036
11 044402 000102
12 044404 000112
13 044406 000126
14 044410 000156
15 044412 000202
16 044414 000150
17 044416 000274
18 044420 000310
19 044422 000026

```

.SBTTL  
CLIRAC:

```

ASL
MOV
ADD
JSR
RTS

```

CLI FOR RREPORT CODING SECTION

```

R2
10$(R2),R2
#10$,R2
PC,(R2)
PC

```

;FORM ADDRESS OF ACTION ROUTINE

10\$:

```

.WORD ACTRML 10$
.WORD ACTRML 10$ ;RPHLP
.WORD ACTREX-10$ ;RPEXT
.WORD ACTRLG 10$ ;RPLOG
.WORD ACTRGS-10$ ;RPGSS
.WORD ACTRTS-10$ ;RPTSS
.WORD ACTRTN 10$ ;RPTSN
.WORD ACTRSE-10$ ;RPSWE
.WORD ACTRSF 10$ ;RPSWF
.WORD ACTRSO 10$ ;RPSWO
.WORD ACTRNF 10$ ;RNOTNF

```

1				.SBTTL	REPORT COMMAND ACTION ROUTINES		
2	044424	112737	177777	003410	ACTRNF: MOV	# 1,P#NNUF	;SET FLAG TO SAY MORE NEEDED
3	044432	000207			ACTRNL: RTS	PC	
4	044434	012702	003304		ACTRML: MOV	#RHLPTB,R2	;SETUP R2 AS A POINTER TO HELP MSG TABLE
5	044440				1\$: PRINTF	#HLPF,(R2).	;PRINT HELP INFORMATION MESSAGES
	044440	012246					MOV (R2), (SP)
	044442	012746	024116				MOV #HLPF, (SP)
	044446	012746	000002				MOV #2, (SP)
	044452	010600					MOV SP,R0
	044454	104417					TRAP C\$PNTF
	044456	062706	000006				ADD #6,SP
6	044462	020227	003314		CMP	R2,#RHLPEN	;SEE IF ALL INFO PRINTED YET
7	044466	001364			BNE	1\$	;IF NO KEEP PRINTING
8	044470	012737	000001	003252	MOV	#RPHLP,KEYWD1	
9	044476	000207			RTS	PC	
10	044500	012737	000002	003252	ACTREX: MOV	#RPEXT,KEYWD1	;SET UP EXIT WORD
11	044506	000207			RTS	PC	
12	044510	004737	043224		ACTRLG: JSR	PC,R#PLOG	;GO REPORT DCLT EVENT LOG
13	044514	012737	000003	003252	MOV	#R#PLOG,KEYWD1	
14	044522	000207			RTS	PC	
15	044524	105037	015756		ACTRGS: CLRB	TRIBN	;FOR GLOBAL STATUS MAKE TRIN =0
16	044530	012737	000105	021026	MOV	#105,TSSKEY	
17	044536	012737	000005	003252	MOV	#RPTSS,KEYWD1	;SET UP KEY WORD
18	044544	000207			RTS	PC	;AND RETURN
19	044546	105037	003410		ACTRSE: CLRB	P#NNUF	;CLEAR NOT NUF FLAG
20	044552	000207			RTS	PC	
21	044554	012737	000105	021026	ACTRTS: MOV	#105,TSSKEY	
22	044562	012737	000005	003252	MOV	#RPTSS,KEYWD1	;SET UP KEY WORD
23	044570	112737	177777	003410	MOV	# 1,P#NNUF	
24	044576	000207			RTS	PC	;AND RETURN
25	044600	105037	003410		ACTRTN: CLRB	P#NNUF	;CLEAR NOT NUF
26	044604	012705	000040		MOV	#32,R5	
27	044610	012702	015712		MOV	#TRIBLS,R2	
28	044614	122237	003404		3\$: CMPB	(R2),P#NUM	
29	044620	001420			BEQ	4\$	
30	044622	005305			DEC	R5	
31	044624	001373			BNE	3\$	
32	044626				PRINTF	#SHTNF,P#NUM	
	044626	013746	003404				MOV P#NUM, (SP)
	044632	012746	026165				MOV #SHTNF, (SP)
	044636	012746	000002				MOV #2, (SP)
	044642	010600					MOV SP,R0
	044644	104417					TRAP C\$PNTF
	044646	062706	000006				ADD #6,SP
33	044652	112737	177777	003411	MOV	# 1,P#GD8D	
34	044660	000403			BR	5\$	
35	044662	113737	003404	015756	4\$: MOV	P#NUM,TRIBN	
36	044670	000207			5\$: RTS	PC	
37	044672	105037	003410		ACTRSF: CLRB	P#NNUF	
38	044676	012737	000106	021026	MOV	#106,TSSKEY	
39	044704	000207			RTS	PC	
40	044706	105037	003410		ACTRSO: CLRB	P#NNUF	
41	044712	023727	003404	000037	CMP	P#NUM,#37	
42	044720	003416			BLF	2\$	
43	044722				PRINTF	#RPTIV,P#NUM	
	044722	013746	003404				MOV P#NUM, (SP)
	044726	012746	025544				MOV #RPTIV, (SP)

044732	012746	000002							
044736	010600							MOV	@2, (SP)
044740	104417							MOV	SP, R0
044742	062706	000006						TRAP	C\$PNTF
44 044746	112737	177777	003411			MOVB	@ 1, P\$GDBD	ADD	@6, SP
45 044754	000406					BR	3\$		
46 044756	013737	003404	021026	2\$:		MOV	P\$NUM, TSSKEY		
47 044764	052737	000040	021026			BIS	@BIT5, TSSKEY		
48 044772	000207			3\$:		RTS	PC		

```

1          .SBTTL          REPORT CODE COMMAND LINE PARSING TREE
2
3 044774   CLIRT:  CLI      CLISPA,0,R10$    ;SKIP ANY SPACES
4 045000   R10$:  CLI      <'?'>,RPHLP,R11$ ;IS FIRST NON-SP CHAR A "?"?
5 045004   CLI      CLIEXI,0    ;EXIT
6 045006   R11$:  CLI      CLISTR,RPHLP,R12$,<'HELP'>
7 045022   CLI      CLIEXI,0
8 045024   R12$:  CLI      CLISTR,RPEXT,R13$,<'EXIT'>
9 045040   CLI      CLIEXI,0
10 045042  R13$:  CLI      CLISTR,;PGSS,R14$,<'GSS'>
11 045054  CLI      CLIBR,0,R20$
12 045060  R14$:  CLI      CLISTR,RPLOG,R15$,<'LOG'>
13 045072  CLI      CLIEXI,0
14 045074  R15$:  CLI      CLISTR,RPTSS,R30$,<'TSS'>
15 045106  CLI      CLISPA,RNOTNF,R30$
16 045112  CLI      CLIDEC,RPTSN,R30$
17 045116  R20$:  CLI      <'/'>,RNOTNF,R125$
18 045122  CLI      CLISTR,RPSWE,R21$,<'ERROR'>
19 045136  CLI      CLIEXI,0
20 045140  R21$:  CLI      CLISTR,RPSWF,R22$,<'FULL'>
21 045154  CLI      CLIEXI,0
22 045156  R22$:  CLI      CLISTR,RNOTNF,R30$,<'OFFSET'>
23 045174  CLI      <'=>,0,R30$
24 045200  CLI      CLIOCT,RPSWO,;~?>
25 045204  CLI      CLIEXI,0
26 045206  R30$:  CLI      CLIERR,0
27 045210  R125$: CLI      CLIEXI,0
28
29

```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 51  
DUMP BYTES OR WORDS

```

1      .SBTTL          DUMP BYTES OR WORDS
2
3
4
5      ;**
6      ; FUNCTIONAL DESCRIPTION:
7      ;   DUMPSR   DUMP BYTES OR WORDS SUBROUTINE
8
9      ;   THIS SUBROUTINE PRINTS THE CONTENTS OF THE LOCATIONS BETWEEN
10     ;   A STARTING AND END ADDRESS IN LOCS. "STADD" AND "ENADD".
11     ;   THE WORD OR BYTE CONTENTS ARE PRINTED 8 TO A LINE WITH THE
12     ;   ADDRESS OF THE FIRST BYTE AS THE FIRST 6 OCTAL CHARS. FOLLOWED
13     ;   BY A SEMICOLON.
14
15     ; INPUTS:
16     ;   STADD=  STARTING ADDRESS (FIRST LOC. TO PRINT)
17     ;   ENADD=  END ADDRESS (LAST LOCATION TO DUMP)
18     ;   BYTBIT= 1 IF SUPPOSED TO PRINT "BYTES"
19     ;             0 IF SUPPOSED TO PRINT "WORDS"
20
21     ; OUTPUTS:
22     ;   CONTENTS OF A RANGE OF LOC.S PRINTED ON THE OPERATORS CONSOLE.
23
24     ; CALLING SEQUENCE:
25     ;   JSR PC,DUMPSR          ;CALL DUMP BYTES SUBROUTINE
26
27     ;--
28 045212 013702 017312  DUMPSR: MOV     STADD,R2          ;SET R2 UP TO STARTING ADDR.
29 045216 005003          DUM4:  CLR     R3              ;CLEAR R3
30 045220          PRINTF  @BASM1,R2          ;PRINT ADDRESS
31          045220 010246          MOV     R2,(SP)
32          045222 012746 030401    MOV     @BASM1,(SP)
33          045226 012746 000002    MOV     @2,(SP)
34          045232 010600          MOV     SP,R0
35          045234 104417          TRAP   C$PNTF
36          045236 062706 000006    ADD     @6,SP
37 045242 005737 017316  DUM3:  TST     BYTBIT          ;IS THIS BYTE OR WORD
38 045246 001416          BEQ     DUM1          ;BR IF WORD
39 045250 112237 017350    MOVB   (R2)+,TEMP     ;MOV BYTE TO TEMP
40 045254          PRINTF  @BASM3,<B,TEMP> ;PRINT BYTE
41          045254 005046          CLR     -(SP)
42          045256 153716 017350    BISB   TEMP,(SP)
43          045262 012746 030363    MOV     @BASM3,(SP)
44          045266 012746 000002    MOV     @2,-(SP)
45          045272 010600          MOV     SP,R0
46          045274 104417          TRAP   C$PNTF
47          045276 062706 000006    ADD     @6,SP
48 045302 000411          BR     DUM2
49 045304          DUM1:  PRINTF  @BASM2,(R2)+ ;PRINT WORD
50          045304 012246          MOV     (R2)+,-(SP)
51          045306 012746 030372    MOV     @BASM2,(SP)
52          045312 012746 000002    MOV     @2,(SP)
53          045316 010600          MOV     SP,R0
54          045320 104417          TRAP   C$PNTF
55          045322 062706 000006    ADD     @6,SP
56 045326 020237 017314  DUM2:  CMP     R2,ENADD      ;COMPARE FOR LAST ADD
57 045332 003005          BGT   DUMEX          ;IF DONE EXIT

```



CZCLMCO DMP V.11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 51 1  
DUMP BYTES OR WORDS

39	045334	005203		INC	R3		;ELSE BUMP R3
40	045336	022703	000010	CMP	#8.,R3		;HAVE WE PRINTED 8 ACROSS
41	045342	001725		BEG	DUM4		;IF SO GO BACK TO 4
42	045344	000736		BR	DUM3		;ELSE GO BACK AND PRINT ANOTHER
43							;BYTE OR WORD
44	045346	000207		DUMEX:	RTS	PC	;RETURN TO CALLER
45							

```

1      .SBTTL          UPDATE TOTAL CHAR. COUNT SUBROUTINE
2
3      ;**
4      ; FUNCTIONAL DESCRIPTION:
5      ;     UPDATES TOTAL CHAR. COUNT TOTCC BASED ON CURCC.
6      ;     LAST MESSAGE IS TRUNCATED TO FIT INTO THE
7      ;     BUFFER IF TOTAL CHAR. COUNT EXCEEDS 'BUFLIM' A MESSAGE
8      ;     IS PRINTED TELLING THE OPERATOR THE TRUNCATION OCCURRED.
9
10     ; INPUTS:
11     ;     CURCC= CHAR. COUNT OF MESSAGE BEING ADDED
12     ;     TOTCC= TOTAL CHAR COUNT OF BUFFER ITS BEING ADDED TO
13
14     ; OUTPUTS:
15     ;     MESSAGE TO OPERATOR IF MESSAGE TRUNCATED TO FIT
16
17     ; FUNCTIONAL SIDE EFFECTS:
18     ;     LOCATION "TEMP" USED FOR CALCULATIONS
19
20     ; CALLING SEQUENCE:
21     ;     JSR      PC,ADDC      ;UPDATED TOTAL CHAR. COUNT
22
23
24     045350 063737 017334 017344  ADDCC:  ADD      CURCC,TOTCC      ;ADD CURRENT TO TOTAL
25     045356 022737 001000 017344  CMP      #BUFLIM,TOTCC  ; COMPARE TO "BUFLIM"
26     045364 103027                      BHS      ADDC1          ;IF NOT MORE THEN "BUFLIM" EXIT
27
28     ; PRINT MESSAGE AND TRUNCATE COUNT
29
30     045366                      PRINTF  #MSGTRU
31     045366 012746 027446                      MOV      #MSGTRU, (SP)
32     045372 012746 000001                      MOV      #1, (SP)
33     045376 010600                      MOV      SP,R0
34     045400 104417                      TRAP    C$PNTF
35     045402 062706 000004                      ADD      #4,SP
36     045406 163737 017334 017344  SUB      CURCC,TOTCC    ;SUB CURRENT FROM TOTAL
37     045414 012737 001000 017350  MOV      #BUFLIM,TEMP   ;MOV "BUFLIM" TO TEMP
38     045422 163737 017344 017350  SUB      TOTCC,TEMP     ;SUB TOTAL FROM "BUFLIM"
39     045430 013737 017350 017334  MOV      TEMP,CURCC     ;AND ESTABLISH NEW CURRENT
40     045436 063737 017334 017344  ADD      CURCC,TOTCC    ;ADD "ADJUSTED CURRENT" TO TOTAL CHAR. CNT.
41     045444 000207                      ADDC1:  RTS      PC     ;RETURN TO CALLER
    
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52

```

.SBTTL          BUILD MESSAGE BUFFERS SUBROUTINE

***
: FUNCTIONAL DESCRIPTION:
:   BLDBUF      BUILD POINTER TABLE AND BUFFERS
:
:   THIS SUBROUTINE ADDS A MESSAGE TO THE TRANSMIT OR EXPECT LIST
:   USING THE POINTER, BYTE COUNT, AND ADDRESS PASSED TO IT.
:
: INPUTS:
:   CURCC= CHAR. COUNT OF MESSAGE TO BE ADDED
:   CURADD= ADDRESS OF MESSAGE TO BE ADDED
:   CPTR=  ADDRESS OF POINTER TABLE WORD WHERE MESSAGE POINTERS ARE
:           TO BE BUILT
:   MSGTYP= VALUE TO USE AS AN INDEX TO FIND SOURCE OF MESSAGE DATA
:           INDEX INTO DMSGCT() AND DMSGAD().
:
: OUTPUTS:
:   A MESSAGE ADDED TO EITHER TXBUF OR CMPBUF
:   APPROPRIATE POINTERS IN PTRTAB POINTER TABLE
:
: CALLING SEQUENCE:
:   JSR PC,BLDBUF          ;BUILD MESSAGE IN BUFFER AND ADD PTRS.
:
:
BLDBUF:
:   MOV      R2, (SP)      ;SAVE R2 AND R3 ON THE STACK
:   MOV      R3, (SP)
:   MOV      CPTR,R2
:
BLDB1:  MOV      CURADD,(R2)+ ;PUT CURRENT ADD ON POINTER TAB
:   MOV      CURCC,(R2)+   ;PUT CURRENT CC ON POINTER TAB
:   MOV      R2,CPTR      ;PUT UPDATED R2 BACK TO CURRENT PCINT
:   MOV      MSGTYP,R2    ;GET MESSAGE TYPE TO USE AS INDEX
:   ASL      R2           ;DOUBLE FOR WORD INDEX
:   MOV      CURADD,TEMP   ;MOVE CURRENT ADD TO TEMP
:   ADD      CURCC,TEMP    ;ADD CHAR COUNT TO IT TO GET END
:   MOV      CURADD,R3    ;SET R3 TO CURRENT START ADD
:   MOV      DMSGCT(R2),TEMP2 ;GET BYTE COUNT
:   MOV      DMSGAD(R2),R4 ;PUT STARTING FROM ADD IN R4
:   ADD      R4,TEMP2     ;ADD IT TO TEMP2 TO GET END OF FROM
:
BLDB3:  MOVB   (R4)+,(R3)+  ;MOV BYTE FROM PATTERN TO BUFFER
:   CMP      R3,TEMP       ;ALL DONE?
:   BEQ     BLDBEX        ;IF SO EXIT
:   CMP     R4,TEMP2      ;IS PATTERN COUNT EXPIRED
:   BEQ     BLDB2        ;IF SO GO START AGAIN
:   BR      BLDB3        ;IF NOT GET ANOTHER BYTE
:
BLDBEX: ADD     CURCC,CURADD ;BUMP CURADD
:   MOV     (SP)+,R3      ;RESTORE R3 AND R2
:   MOV     (SP)+,R2
:   RTS     PC           ;RETURN TO CALLER
    
```

```

25 045446
26 045446 010246
27 045450 010346
28 045452 013702 017340
29
30 045456 013722 017342
31 045462 013722 017334
32 045466 010237 017340
33 045472 013702 017332
34 045476 006302
35 045500 013737 017342 017350
36 045506 063737 017334 017350
37 045514 013703 017342
38 045520 016237 002150 017354
39 045526 016204 002176
40 045532 060437 017354
41 045536 112423
42 045540 020337 017350
43 045544 001404
44 045546 020437 017354
45 045552 001762
46 045554 000770
47 045556 063737 017334 017342
48 045564 012603
49 045566 012602
50 045570 000207
    
```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22-Mar-84 16:24 Page 54  
 CREATE FACSIMILE OF TX BUFFER AND MESSAGE LIST

```

1      .SBTTL  CREATE FACSIMILE OF TX BUFFER AND MESSAGE LIST
2
3      : **
4      : FUNCTIONAL DESCRIPTION:
5      :   FACSIMILE: THIS ROUTINE IS USED TO CREATE A FACSIMILE OF THE
6      :   OF THE TRANSMIT LIST AND TRANSMIT BUFFER IN THE
7      :   EXPECT LIST AND EXPECT BUFFER. THE ROUTINE IS
8      :   NORMALLY CALLED WHEN USER COMMAND "SET E [XPECT]=
9      :   T [RANSMIT] IS ENTERED.
10
11      :   CALLING SEQUENCE: JSR  PC,FACSIMILE
12
13      :
14      : DEFINITIONS  CMPBUF = EXPECTED DATA BUFFER  HOLDS MAX 512 BYTES
15      :                TXBUF  = TRANSMIT DATA BUFFER  HOLDS MAX 512 BYTES
16      :                TTOTCC = NUMBER OF BYTES IN TXBUF
17      :                PTRTAB = TOP OF MESSAGE LIST POINTER TABLE
18      :                CTOTCC = NUMBER OF BYTES IN EXPECT MESSAGE
19      :                CMPTOT = NUMBER OF EXPECTED MESSAGES
20      :                CMPPTR = EXPECTED MESSAGE LIST POINTER
21      :                TXPTR  = TRANSMIT MESSAGE LIST POINTER
22      :                TXMTOT = NUMBER OF TRANSMIT MESSAGES
23      :                CCURAD = STORAGE ADDRESS OF MESSAGE IN CMPBUF
24      :                MSGLIN = MAXIMUM NUMBER OF MESSAGES THAT CAN BE STORED
25      :                BUFLIM = NUMBER OF BYTES IN BUFFER
26
27      : BEGIN FACSIMILE ROUTINE
28      : (*COPY TXBUF ==> CMPBUF*)
29      : ..SAVE R1
30      : ..INIT R1
31      : ..REPEAT
32      : ....[CMPBUF]R1=[TXBUF]R1
33      : ....R1=R1+1
34      : ..UNTIL R1 = BUFLIM
35
36      : (*NOW CALCULATE EXPECT LIST MESSAGE POINTER*)
37      : ..CMPPTR = PTRTAB + (2 * MSGLIN)
38
39      : (*NOW PRIME THE WHILE - DO LOOP*)
40      : ..TXPTR = PTRTAB
41      : ..CCURAD = CMPBUF
42      : ..TXPTR = TXPTR + 2
43      : ..CTOTCC = [TXPTR]
44      : ..CMPTOT = 0
45      : ..WHILE TXMTOT <> CMPTOT DO
46      : ....[CMPPTR] = CCURAD
47      : ....CMPPTR = CMPPTR + 2
48      : ....[CMPPTR] = CTOTCC
49      : ....TXPTR = TXPTR + 4
50      : ....CCURAD = CCURAD + CTOTCC
51      : ....CTOTCC = [TXPTR]
52      : ....CMPPTR = CMPPTR + 2
53      : ....CMPTOT = CMPTOT + 1
54      : ..END WHILE DO
55      : ..CTOTCC = TTOTCC
56      : END FACSIMILE ROUTINE
57

```

CZCLMCO CMP/V-11 DCLT MACRO /05.00 Thursday 22-Mar-84 16:24 Page 54-1  
 CPEATE FACSIMILE OF TX BUFFER AND MESSAGE LIST

```

58 045572                                FACSIMILE:
59 045572 010146                          MOV    R1, (SP)                ;SAVE R1
60 045574 005001                          CLR    R1                    ;INIT R1
61 045576 116161 003416 004416 10$:      MOVB   TXBUF(R1),CMPBUF(R1)   ;COPY TX BUFFER TO EXPECTED BUFFER
62 045604 005201                          INC    R1                    ;BUMP INDEX
63 045606 020127 001000                  CMP    R1,#BUFLIM           ;ALL DATA COPIED ?
64 045612 001371                          BNE   10$                   ;NO,BRANCH
65
66 045614 012701 000017                    20$:  MOV    #MSGLIM,R1        ;MESSAGE LIMIT
67 045620 006301                          ASL   R1                    ;MULTIPLY BY 2
68 045622 006301                          ASL   R1                    ;MULTIPLY BY 2
69 045624 012737 011416 017240          MOV    #PTRTAB,CMPPTR      ;TOP OF POINTER TABLE
70 045632 060137 017240                  ADI   R1,CMPPTR            ;START OF EXPECTED POINTER TABLE
71 045636 005001                          CLR   R1                    ;INIT R1
72
73                                          ;SET UP WHILE - DO LOOP
74 045640 012737 011416 017236          MOV    #PTRTAB, TXPTR      ;TX POINTER NOW AT TOP OF TABLE
75 045646 012737 004416 017246          MOV    #CMPBUF,CCURAD     ;TRANSFER ADDRESS OF 1ST MESSAGE
76 045654 062737 000002 017236          ADD    #2, TXPTR           ;BUMP POINTER
77 045662 017737 151350 017244          MOV    #TXPTR,CTOTCC      ;BYTE COUNTER 1ST MESSAGE
78 045670 005037 017242                  CLR   CMPTOT              ;INIT EXPECTED MESSAGE COUNT
79
80                                          ;WHILE TX MESSAGE TOTAL <> EXPECTED MESSAGE TOTAL DO
81 045674 023737 017260 017242 30$:     CMP    TXMTOT,CMPTOT      ;ALL MESSAGES COPIED ?
82 045702 001430                          BEQ   40$                   ;YES,BRANCH
83 045704 013777 017246 151326          MOV    CCURAD,#CMPPTR     ;TRANSFER ADDRESS OF MESSAGE
84 045712 062737 000002 017240          ADI   #2,CMPPTR           ;BUMP POINTER
85 045720 013777 017244 151312          MOV    CTOTCC,#CMPPTR    ;BYTE COUNT OF MESSAGE
86 045726 062737 000004 017236          ADD    #4, TXPTR          ;BUMP TX MESSAGE POINTER
87 045734 063737 017244 017246          ADD    CTOTCC,CCURAD     ;CALC. TRANSFER ADDRESS
88 045742 017737 151270 017244          MOV    #TXPTR,CTOTCC     ;BYTE COUNT NEXT MESSAGE
89 045750 062737 000002 017240          ADD    #2,CMPPTR         ;BUMP POINTER
90 045756 005237 017242                  INC   CMPTOT              ;INCREMENT MESSAGE COUNT
91 045762 000744                          BR    30$                   ;DO IT AGAIN
92                                          ;END WHILE - DO
93 045764 013737 017262 017244 40$:     MOV    TTOTCC,CTOTCC      ;COPY TOTAL CHARACTER COUNT
94
95                                          ;END ROUTINE
96 045772 012601                          MOV    (SP)+,R1            ;RESTORE R1
97 045774 000207                          RTS    PC                  ;RETURN
98
99
100

```

```

1          .SBTTL          DO ALL G OBAL PARMAS
2
3          ;**
4          ; FUNCTIONAL DESCRIPTION:          DOGLOB - ASK QUESTIONS ABOUT ALL GLOBALS
5          ;
6          ;          THIS ROUTINE ASKS QUESTIONS TO ALL GLOBAL POLL PARMAS
7          ;          IF NESCESSARY THEN CLEARS THE WRITE GLOBAL FLAG
8          ;
9          ; CALLING SEQUENCE:
10         ;          JSR          PC,DOGLOB
11         ; --
12
13 045776 010246          DOGLOB: MOV          R2,-(SP)          ;SAVE R2,R3,R4 ON THE STACK
14 046000 010346          MOV          R3,-(SP)
15 046002 010446          MOV          R4,-(SP)
16 046004 105037 003412          CLRB          WRFLG          ;CLEAR WRITE GLOBAL FLAG
17 046010          PRINTF          #POLPM3          ;PRINT GLOBAL PARAMS ARE
18          046010 012746 025301          MOV          #POLPM3,-(SP)
19          046014 012746 000001          MOV          #1,(SP)
20          046020 010600          MOV          SP,R0
21          046022 104417          TRAP          C$PNTF
22          046024 062706 000004          ADD          #4,SP
23          18 046030          CLR          R3
24          19 046032 012702 000032          DOGL1: MOV          #32,R2
25          20 046036 005202          INC          R2
26          21 046040 010204          MOV          R2,R4
27          22 046042 006304          ASL          R4
28          23 046044 016337 017220 017350          MOV          GLBPLS(R3),TEMP ;GET DEFAULT
29          24 046052 016437 021030 017366          MOV          GSSLST(R4),CONOTM
30          25 046060          PRINTF          CONOTM,TEMP
31          26          046060 013746 017350          MOV          TEMP,(SP)
32          27          046064 013746 017366          MOV          CONOTM,-(SP)
33          28          046070 012746 000002          MOV          #2,(SP)
34          29          046074 010600          MOV          SP,R0
35          30          046076 104417          TRAP          C$PNTF
36          31          046100 062706 000006          ADD          #6,SP
37          32 046104          GMANID EQUQ.TEMP,0, 1,0,-1,YES          ;GET INPUT
38          33          046104 104443          TRAP          C$GMAN
39          34          046106 000406          BR          10001$
40          35          046110 017350          .WORD          TEMP
41          36          046112 000032          .WORD          T$CODE
42          37          046114 025055          .WORD          EQUQ
43          38          046116 177777          .WORD          -1
44          39          046120 000000          .WORD          T$LOLIM
45          40          046122 177777          .WORD          T$HILIM
46          41          046124          10001$:
47          42 27 046124 013763 017350 017220          MOV          TEMP,GLBPLS(R3)          ;PUT ANSWER BACK
48          43 28 046132 062703 000002          ADD          #2,R3          ;BUMP R3
49          44 29 046136 032737 000002 023102          BIT          #TRBB,DEVPAR          ;IS THIS TRIB
50          45 30 046144 001403          BEQ          DOGL4          ;BRANCH IF TRIB
51          46 31
52          47 32 046146 022702 000037          DOGL2: CMP          #37,R2          ;ALL DONE
53          48 33 046152 001331          BNE          DOGL1
54          49 34
55          50 35 046154 012604          DOGL4: MOV          (SP)+,R4          ;RESTORE R4,R3,R2
56          51 36 046156 012603          MOV          (SP)+,R3
57          52 37 046160 012602          MOV          (SP)+,R2

```

N9

LZCLMCO DMP/V 11 DCLT MACRO '05.00 Thursday 22 Mar 84 16:24 Page 55 1  
DO ALL GLOBAL PARMAS

SEQ 117

38 046162 000207

RTS PC

;RETURN TO CALLING ROUTINE

```

1      .SBTTL          QUEUE UP ALL REC BUFFERS FOR MULTIPOINT
2
3
4      ;**
5      ; FUNCTIONAL DESCRIPTION:          QURXAL  QUEUE ALL REC BUFFERS
6      ;
7      ; THIS ROUTINE QUEUES ALL REC BUFFERS FOR VALID TRIBS
8      ; IF MODE IS POINT TO POINT TRIB LIST WILL STILL BE ONE.
9      ;
10     ; SUBORDINATE ROUTINES USED:
11     ;          GTVIND          LOADS INDEX WITH OFFSET TO NEXT
12     ;                          VALID TRIB AND LOADS TRIBN WITH
13     ;                          ADDRESS OF NEXT VALID TRIB
14     ;          ULRPLS          MOVES RXPTR FOR THIS TRIB TO
15     ;                          CPTRR FROM CPTRLS.
16     ;          LOGAQR          QUES REC BUFFER POINTED TO BY
17     ;                          CPTRR AND LOGS THIS IN EVENT LOG
18     ;          LDRPLS          MOVES VALUE OF CPTRR TO SLOT IN
19     ;                          CPTRLS FOR THIS TRIB
20     ;
21     ; CALLING SEQUENCE:
22     ;          JSR          PC,RXQUAL
23     ;**
24
25
26 046164 012737 177777 015762 RXQUAL: MOV      #-1,INDEX          ;SET INDEX TO -1
27 046172 004737 046462          RXQU1: JSR      PC,GTVIND          ;GET NEXT VALID INDEX
28 046176 022737 000040 015762          CMP      #32.,INDEX          ;IS ALL DONE
29 046204 001412          BEQ      RXQUEX          ;IF SO EXIT
30 046206 004737 046526          JSR      PC,ULRPLS          ;LOAD CPTRR FOR THIS TRIB
31 046212 052737 000004 017414          BIS      #QRX,FLAG          ;SET THE QRX,FLAG
32 046220 004737 047302          JSR      PC,LOGAQR
33 046224 004737 046506          JSR      PC,LDRPLS          ;RELOAD RX PTR LIST
34 046230 000760          BR       RXQU1          ;AND THEN GO BACK FOR MORE
35 046232 000207          RXQUEX: RTS     PC          ;RETURN TO CALLER
36
    
```



```

1      .SBTTL          LOAD CPTRLS LIST INITIALLY
2
3      ;**
4      ; FUNCTIONAL DESCRIPTION:      LCPRLS -LOAD CPTR LIST INITIALLY
5      ;
6      ;          THIS ROUTINE LOADS UP THE CPTRLS LIST FOR ALL
7      ;          VALID TRIB ADDRESS IN THE TRIBLS IT ALSO LOADS
8      ;          THE DVRCLS LIST FOR MSG COUNTS.
9      ;
10     ; INPUTS:      RXMTOT   TOTAL NUMBER OF RX MSGS PER TRIB
11     ;
12     ; OUTPUTS:     CPTRLS   LOADED WITH POINTERS TO THE RXPTR TABLE
13     ;               FOR EACH TRIB
14     ;               DVRCLS   LOADED WITH RXTOT COUNT FOR EACH TRIB
15     ;
16     ; SUBORDINATE ROUTINES USED:
17     ;               GTVIND   GETS NEXT VALID INDEX BY
18     ;               LCPRL1   CHECKING TRIBLS FOR NON ZERO ENTRY
19     ;                       LOADS POINTER TABLE FOR TRIB AT THIS
20     ;                       INDEX VALUE AND RXMTOT TO DVRCLS FOR
21     ;                       THIS TRIB.
22     ;
23     ; CALLING SEQUENCE:
24     ;               JSR      PC,LCPRLS
25     ;
26
27 046234 012737 177777 015762 LCPRLS: MOV     # 1,INDEX      ;SET UP INDEX VALUE TO 1
28 046242 004737 046462          LCPRL1: JSR     PC,GTVIND    ;GET VALID INDEX
29 046246 022737 000040 015762          CMP     #32.,INDEX   ;IS IT 32?
30 046254 001403          BEQ     LCPREX    ;BRANCH IF 32.
31 046256 004737 046266          JSR     PC,LCPRL1   ;IF NOT LOAD CPTRLS FOR THIS TRIB.
32 046262 000767          BR      LCPRL1    ;GO BACK FOR NEXT
33 046264 000207          LCPREX: RTS    PC        ;RETURN TO CALLER WHEN DONE WITH ALL.
34
35
36

```

```

1      .SBTTL          LOAD CPTRLS AND DVRCLS FROM INDEX
2
3      ;**
4      ; FUNCTIONAL DESCRIPTION:          LCPRL1 - LOAD CPTRLS AND DVRCLS FROM INDEX
5      ;
6      ;          THIS ROUTINE LOADS UP THE CPTRLS LIST FOR THE
7      ;          INDEX VALUE AND THE DVRCLS IS LOADED WITH RXMTOT.
8      ;
9      ; INPUTS:          RXMTOT - TOTAL NUMBER OF RX MSGS PER TRIB
10     ;          PTR23 - START OF RX POINTER TABLE
11     ;
12     ; OUTPUTS:          CPTRLS   LOADED WITH POINTERS TO THE RXPTR LIST
13     ;          DVRCLS   LOADED WITH RXMTOT COUNT
14     ;
15     ; SUBORDINATE ROUTINES USED:
16     ;          MTPLY          MULTIPLIES VALUE IN INDEX BY VALUE IN
17     ;          TEMP AND THEN ADDS THAT RESULT TO VALUE
18     ;          IN TEMP2 AND PUTS FINAL RESULT IN TEMP2
19     ;
20     ; CALLING SEQUENCE:
21     ;          JSR          PC,LCPRL1
22     ;--
23
24 046266 012737 011606 017354 LCPRL1: MOV     #PTR23,TEMP2      ;SET UP TEMP 2 AS BASE
25 046274 013737 015762 017232      MOV     INDEX,MPLY      ;SET UP MULTIPLIER
26 046302 012737 000074 017350      MOV     #60, .MP        ;SET UP MULTIPLICAN
27 046310 004737 046436              JSR     PC,MTPLY        ;GO MULTIPY
28 046314 013703 015762              LCPRL2: MOV     INDEX,R3
29 046320 113763 017276 015612      MOVVB  RXMTOT,DVRCLS(R3) ;LOAD UP COUNT LIST
30 046326 006303              ASL     R3              ;MAKE R3 WORD INDEX
31 046330 013763 017354 015412      MOV     TEMP2,CPTRLS(R3) ;SET UP POINTER TABLE
32 046336 000207              RTS     PC              ;RETURN TO CALLER
33
34
    
```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 59  
 CLEAR RECEIVE POINTER LIST

```

1          .SBTTI          CLEAR RECIVE POINTER LIST
2
3          ;**
4          ; FUNCTIONAL DESCRIPTION:      CLRPLS - CLEAR RX POINTER LIST
5          ;                             THIS ROUTINE CLEARS ALL 32 SLOTS OF THE CTRPLS
6          ;
7          ; OUTPUTS:          CTRPLS - IS ZEROED IN ALL SLOTS
8          ;
9          ; CALLING SEQUENCE:
10         ;                   JSR      PC,CLRPLS
11         ;--
12
13 046340 012737 000040 017350 CLRPLS: MOV      #32,TEMP
14 046346 012703 015412          MOV      @CTRPLS,R3      ;LOAD START OF LIST TO R3
15 046352 005023          CLRPL1: CLR      (R3)+          ;CLEAR THIS SLOT
16 046354 005337 017350          DEC      TEMP
17 046360 001374          BNE      CLRPL1          ;IF NOT DONE GO BACK
18 046362 000207          CLRPEX: RTS     PC          ;RETURN TO CALLER WHEN DONE
19
20
21

```

```

1      .SBTTL          LOAD TX POINTER LIST INITIALLY
2
3      ;**
4      ; FUNCTIONAL DESCRIPTION:      LCPTLS - LOAD TRANSMIT POINTER LIST
5      ;                             THIS ROUTINE LOADS CPTTLS WITH TX POINTERS
6      ;                             FOR EACH VALID TRIB.
7      ;
8      ; INPUTS:
9      ;     TXMTOT          TOTAL NUMBER OF TX MSGS
10     ;     PTRTAB         POINTER TO TOP OF TX POINTER TABLE
11
12     ; OUTPUTS:
13     ;     CCTLS         LOADED WITH POINTERS TO TX POINTER TABLE
14     ;                   FOR ALL VALID TRIBS
15     ;     DVTCLS        TX MSG COUNT LIST LOADED WITH MSG COUNTS
16     ;                   FOR ALL VALID TRIBS
17     ; SUBORDINATE ROUTINES USED:
18     ;     GTVIND         GETS NEXT VALID INDEX BY
19     ;                   CHECKING TRIBLS FOR NON ZERO ENTRY
20
21     ;     LDTPLS         LOADS VALUE FROM CPTR TO CPTTLS INDEXED
22     ;                   BY TRIBN
23
24     ;     LDTCLS         LOADS DVTCT TO DVTCLS INDEXED BY TRIBN
25
26     ; CALLING SEQUENCE:
27     ;     JSR           PC,LCPTLS
28     ; --
29 046364 013737 017260 017256 LCPTLS: MOV     TXMTOT,DVTCT      ;LOAD UP COUNT
30 046372 012737 011416 017340      MOV     #PTRTAB,CPTR
31 046400 012737 177777 015762      MOV     #-1,INDEX      ;LOAD INDEX WITH 1
32 046406 004737 046462      LCPT1: JSR     PC,GTVIND  ;GET VALID INDEX
33 046412 022737 000040 015762      CMP     #32.,INDEX     ;IS THIS THE END
34 046420 001405      BEQ     LCPTEX        ;EXIT IF SO
35 046422 004737 046644      JSR     PC,LDTPLS     ;LOAD TX POINTER LIST
36 046426 004737 046704      JSR     PC,LDTCLS     ;LOAD TX COUNT LIST
37 046432 000765      BR      LCPT1        ;GO BACK
38 046434 000207      LCPTEX: RTS      PC      ;RETURN TO CALLER
39

```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22-Mar 84 16:24 Page 61  
MULTIPLY

```

1          .SBTTL          MULTIPLY
2
3          ;**
4          ; FUNCTIONAL DESCRIPTION:          MTPLY MULTIPLY
5          ; THIS ROUTINE MULTIPLIES THE VALUE IN MPLY BY
6          ; THE VALUE IN TEMP AND THEN ADDS IN THE VALUE OF TEMP2
7          ; WITH THE RESULT GOING TO TEMP2
8          ;
9          ; INPUTS:          TEMP2 - INITIALLY VALUE
10         ;                  TEMP  - VALUE TO MULTIPLY BY
11         ;                  MPLY  - NUMBER OF TIMES TO MULTIPLY
12         ;
13         ; OUTPUTS:          TEMP2 - RESULT OF [MPLY * TEMP].TEMP2
14         ;
15         ; CALLING SEQUENCE:
16         ;                  JSR    PC,MTPLY
17         ;--
18
19 046436 005737 C17232 MTPLY: TST    MPLY
20 046442 001406          BEQ    MTPLEX          ;IF MULTIPLIER IS ZERO QUIT
21 046444 063737 017350 017354 ADD    TEMP,TEMP2      ;ADD THE FACTOR TO BASE
22 046452 005337 017232          DEC    MPLY          ;COUNT DOWN THE MULTIPLIER
23 046456 000767          BR     MTPLY          ;GO BACK FOR MORE
24 046460 000207          MTPLEX: RTS    PC          ;RETURN TO CALLEP
25

```

```

1      .SBTTL          GET NEXT VALID INDEX
2
3      ;**
4      ; FUNCTIONAL DESCRIPTION:          GTVIND          GET NEXT VALID INDEX
5      ;
6      ;          THIS LOADS INDEX WITH INDEX VALUE OF NEXT VALID TRIB. THIS ALSO
7      ;          LOADS TRIBN WITH THE ADDRESS.
8      ;          TRIB BEING THE LOCATION IN THE TRIBLS THAT HAS A NON ZERO
9      ;          ENTRY.
10     ;
11     ; INPUTS:          INDEX          SET TO VALUE OF LAST INDEX
12     ; OUTPUTS:        INDEX          SET TO VALUE OF THIS TRIB
13     ;          TRIBN          ADDRESS OF THIS TRIB
14     ; CALLING SEQUENCE:
15     ;          JSR          PC,GTVIND
16     ;
17
18 046462 013703 015762      GTVIND: MOV          INDEX,R3
19 046466 005203      GTVI1:  INC          R3
20 046470 116337 015712 015756      MOVB         TRIBLS(R3) TRIBN          ;LOAD TRIBN
21 046476 001773      BEQ          GTVI1          ;IF ZERO GO GET ANOTHER
22 046500 010337 015762      MOV          R3,INDEX          ;LOAD INDEX VALUE IF NOT ZERO
23 046504 000207      RTS          PC          ;RETURN TO CALLER WHEN DONE
24
    
```

CZCLMCO GMP V 11 DCL T  
LOAD REC POINTER LIST

MACRO V05.00 Thursday 22 Mar 84 16:24 Page 63

```

1      .SBTTL          LOAD REC POINTER LIST
2      ;**
3      ; FUNCTIONAL DESCRIPTION:      LDRPLS - LOAD RX POINTER LIST FROM CPTRR
4      ; THIS ROUTINE MOVES DATA FROM CPTRR TO THE SLOT IN THE
5      ; CPTRLS INDEXED BY INDW.
6      ; INPUTS:                      INDW  WORD INDEX INTO LIST
7      ; OUTPUTS:                     CPTRLS  CORRECT SLOT LOADED WITH DATA FROM CPTRR
8      ; SUBORDINATE ROUTINES USED:
9      ; GETIND                      GETS INDW FOR THIS TRIBN
10     ; CALLING SEQUENCE
11     ; JSR      PC,LDRPLS
12     ;
13
14     046506  004737  047154      LDRPLS: JSR      PC,GETIND          ;GET INDW FOR THIS TRIBN
15     046512  013703  015760          MOV      INDW,R3          ;MOVE WORD INDEX TO R3
16     046516  013763  017336  015412  MOV      CPTRR,CPTRLS(R3) ;LOAD CPTRLS LIST
17     046524  000207          RTS      PC          ;RETURN TO CALLER
18
19     .SBTTL          UNLOAD CPTRR LIST
20     ;**
21     ; FUNCTIONAL DESCRIPTION:      ULRPLS  UNLOAD RX POINTER LIST
22     ; THIS ROUTINE MOVES DATA FROM CPTRLS SLOT INDEXED
23     ; BY INDW TO CPTRR.
24     ; IMPLICIT INPUTS:
25     ; TRIBN                      ADDRESS OF CURRENT TRIB
26     ; CPTRR                      VALUE FROM CPTRLS
27     ; SUBORDINATE ROUTINES USED:
28     ; GETIND                      GET INDW FOR THIS TRIBN
29     ; CALLING SEQUENCE:
30     ; JSR      PC,ULRPLS
31     ;
32
33     046526  004737  047154      ULRPLS: JSR      PC,GETIND          ;GET INDEX
34     046532  013703  015760          MOV      INDW,R3          ;MOVE WORD INDEX TO R3
35     046536  016337  015412  017336  MOV      CPTRLS(R3),CPTRR ;LOAD CPTRR FROM LIST INDEX
36     046544  000207          RTS      PC          ;RETURN TO CALLER

```

```

1      .SBTTL          GET REC POINTER TO CPTR
2
3      ;**
4      ; FUNCTIONAL DESCRIPTION:          GRPTCP  GET RX POINTER TO CPTR
5      ; THIS ROUTINE GETS THE RX POINTER TO CPTR FOR USE IN BUILD
6      ; BUFFER
7
8      ; INPUTS:          INDEX  INDEX VALUE FOR TRIB
9
10     ; OUTPUTS:         CPTR  - LOADED WITH ADDRESS OF RX BUFFER FOR THIS TRIB
11     ; SUBORDINATE ROUTINES USED:
12     ; MTPLY  MULTIPLIES INDEX BY TEMP AND ADDS TEMP2 TO RESULT
13     ; CALLING SEQUENCE:
14     ; JSR      PC,GRPTCP
15     ;
16
17
18 046546 013737 015762 017232 GRPTCP: MOV     INDEX,MPLY      ;SET UP MULPILIER
19 046554 012737 000074 017350      MOV     #60.,TEMP
20 046562 013737 017234 017354      MOV     RXPTR,TEMP2
21 046570 004737 046436              JSR     PC,MTPLY      ;[INDEX VALUE X 60.] + RXPTR  POINTER ADDRESS
22 046574 013737 017354 017340      MOV     TEMP2,CPTR   ;SET UP POINTER ADDR.
23 046602 000207
    
```



```

1      .SBTTL          LOAD DVRCT LIST
2      ;**
3      ; FUNCTIONAL DESCRIPTION:      LDRCLS - LOAD RX COUNT LIST
4      ; THIS ROUTINE LOADS THE VALUE FROM DVRCT TO
5      ; THE SLOT IN DVRCLS INDEXED BY TRIBN
6      ; INPUTS:                      TRIBN - ADDRESS OF TRIB IN USE
7      ;                               DRVCT - COUNT VALUE TO GO TO LIST
8      ; OUTPUTS:                     DVRCLS VALUE OF DRVCT
9      ; SUBORDINATE ROUTINES USED:
10     ; GETIND                      GET INDEX FROM TRIBLS
11     ; CALLING SEQUENCE:
12     ; JSR          PC,LDRCLS
13     ; --
14
15 046604 004737 047154      LDRCLS: JSR          PC,GETIND          ;GET INDEX
16 046610 013703 015762      MOV          INDEX,R3          ;LOAD R3 WITH BYTE INDEX
17 046614 113763 017274 015612  MOVB         DVRCT,DVRCLS(R3)      ;LOAD LIST WITH COUNT
18 046622 000207              RTS          PC          ;RETURN TO CALLER
19
20     .SBTTL          UNLOAD DVRCT LIST
21     ;**
22     ; FUNCTIONAL DESCRIPTION:      ULRCLS UNLOAD RX COUNT LIST
23     ; THIS ROUTINE UNLOADS THE VALUE TO DVRCT FROM
24     ; THE SLOT IN DVRCLS INDEXED BY TRIBN
25     ; INPUTS:                      TRIBN - ADDRESS OF TRIB IN USE
26     ;                               DVRCLS VALUE OF DRVCT
27     ; OUTPUTS:                     DRVCT - COUNT VALUE FROM LIST
28     ; SUBORDINATE ROUTINES USED:
29     ; GETIND                      GET INDEX FROM TRIBLS
30     ; CALLING SEQUENCE:
31     ; JSR          PC,ULRCLS
32     ;
33
34
35 046624 004737 047154      ULRCLS: JSR          PC,GETIND          ;GET INDEX
36 046630 013703 015762      MOV          INDEX,R3          ;MOVE INDEX TO R3
37 046634 116337 015612 017274  MOVB         DVRCLS(R3),DVRCT      ;UNLOAD LIST
38 046642 000207              RTS          PC          ;RETURN TO CALLER

```

CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 22-Mar-84 16:24 Page 66  
 LOAD CPTR LIST (TRANSMIT POINTER)

```

1      .SBTTL          LOAD CPTR LIST (TRANSMIT POINTER)
2
3      ;**
4      ; FUNCTIONAL DESCRIPTION:      LDTPLS - LOAD TX POINTER LIST
5      ; THIS ROUTINE LOADS THE VALUE FROM CPTR TO
6      ; THE TX POINTER LIST INDEXED BY TRIBN INDEX.
7      ; INPUTS:                      TRIBN - ADDRESS OF TRIB IN USE
8      ; OUTPUTS:                     CPTTLS - SLOT LOADED WITH CPTR DATA
9      ; SUBORDINATE ROUTINES USED:
10     ; GETIND - GET INDEX VALUE FROM TRIBLS
11     ; CALLING SEQUENCE:
12     ; JSR      PC,LDTPLS
13     ;
14
15     046644 004737 047154      LDTPLS: JSR      PC,GETIND      ;GET INDEX
16     046650 013703 015760      MOV      INDW,R3      ;MOVE INDEX TO R3
17     046654 013763 017340 015512  MOV      CPTR,CPTTLS(R3) ;LOAD LIST
18     046662 000207              RTS      PC              ;RETURN TO CALLER
19
20     .SBTTL          UNLOAD CPTR LIST (TRANSMIT POINTER)
21
22     ;**
23     ; FUNCTIONAL DESCRIPTION:      ULTPLS - UNLOAD TX POINTER LIST
24     ; THIS ROUTINE MOVES DATA FROM TX POINTER LIST
25     ; TO CPTR.
26     ; INPUTS:                      TRIBN - ADDRESS OF TRIB IN USE
27     ; OUTPUTS:                     CPTR - VALUE FROM THE TX POINTER LIST
28     ; SUBORDINATE ROUTINES USED:
29     ; GETIND - GET INDEX FROM TRIBLS
30     ; CALLING SEQUENCE:
31     ; JSR      PC,ULTPLS
32     ; -
33
34     046664 004737 047154      ULTPLS: JSR      PC,GETIND      ;GET INDEX
35     046670 013703 015760      MOV      INDW,R3      ;MOVE WORD INDEX TO R3
36     046674 016337 015512 017340  MOV      CPTTLS(R3),CPTR ;GET PTR FROM LIST
37     046702 000207              RTS      PC              ;RETURN TO CALLER
38

```

CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 67  
 LOAD DVTCT LIST (TRANSMIT COUNT)

```

1      .SBTTL          LOAD DVTCT LIST (TRANSMIT COUNT)
2
3      ;**
4      ; FUNCTIONAL DESCRIPTION:          LDTCLS - LOAD TX COUNT LIST
5      ;                               THIS ROUTINE LOADS A VALUE FROM DVTCT TO
6      ;                               THE TX COUNT LIST (DVTCLS). INDEXED BY TRIBN.
7
8      ; INPUTS:          TRIBN          - ADDRESS OF TRIB IN USE
9      ;                 DVTCT          - CURRENT TX COUNT FOR TRIB
10     ; OUTPUTS:         DVTCLS         - SLOT LOADED WITH DVTCT
11     ; SUBORDINATE ROUTINES USED:
12     ;                 GETIND         - GET INDEX FROM TRIBLS
13     ; CALLING SEQUENCE:
14     ;                 JSR           PC,LDTCLS
15     ; -
16
17     046704 004737 047154      LDTCLS: JSR           PC,GETIND          ;GET INDEX
18     046710 013703 015762      MOV           INDEX,R3              ;MOVE BYTE INDEX TO R3
19     046714 113763 017256 015652  MOVB          DVTCT,DVTCLS(R3);LOAD LIST
20     046722 000207              RTS            PC                  ;RETURN TO CALLER
21
22     .SBTTL          UNLOAD DVTCT LIST (TX COUNT)
23     ;**
24     ; FUNCTIONAL DESCRIPTION:          ULTCLS - UNLOAD TX COUNT LIST
25     ;                               THIS ROUTINE TAKES DATA FROM DVTCLS AND MOVES
26     ;                               IT TO DVTCT
27     ; INPUTS:          TRIBN          - ADDRESS OF TRIBN IN USE
28     ;                 DVTCT          - VLAUE
29     ; SUBORDINATE ROUTINES USED:
30     ;                 GETIND         - GET INDEX VALUE FROM TRIBLS
31     ; CALLING SEQUENCE:
32     ;                 JSR           PC,ULTCLS
33     ; -
34
35     046724 004737 047154      ULTCLS: JSR           PC,GETIND          ;GET INDEX
36     046730 013703 015762      MOV           INDEX,R3              ;MOVE BYTE INDEX TO R3
37     046734 116337 015652 017256  MOVB          DVTCLS(R3),DVTCT
38     046742 000207              RTS            PC                  ;RETURN TO CALLER
39
40

```

```

1      .SBTTL          GET ALL RX POINTERS FROM LIST TO CPTRR
2
3      ;**
4      ; FUNCTIONAL DESCRIPTION:      GARPFL - GET ALL RX POINTERS FROM LIST
5      ;                             THIS ROUTINE CHECKS ALL RX POINTERS FOR VALID TRIBS
6      ;                             IN CPTRLS AND MAKES SURE THEY ARE ALL ZERO.
7      ; OUTPUTS:                    CPTRR - ZERO IF ALL CPTRLS IS ZERO
8      ;                             NON ZERO IF NOT.
9
10     ; SUBORDINATE ROUTINES USED:
11     ;                             GTVIND - GET VALID INDEX
12     ;                             ULRPLS  UNLOAD CPTRR LIST TO CPTRR
13     ; CALLING SEQUENCE:
14     ;                             JSR      PC,GARPFL
15     ;--
16
17 046744 013737 015756 017362 GARPFL: MOV      TRIBN,TEMP5
18 046752 012737 177777 015762          MOV      # -1,INDEX
19 046760 004737 046462          GARP1:  JSR      PC,GTVIND      ;GET VALID INDEX
20 046764 022737 030040 015762          CMP      #32.,INDEX    ;COMPARE INDEX
21 046772 001405          BEQ      GARP1          ;EXIT IF DONE
22 046774 004737 046526          JSR      PC,ULRPLS     ;LOAD CPTRR WITH VALUE
23 047000 005737 017336          TST      CPTRR        ;TEST THE VALUE
24 047004 001765          BEQ      GARP1          ;IF ZERO CHECK NEXT
25 047006 013737 017362 015756 GARPEX: MOV      TEMP5,TRIBN
26 047014 000207          RTS      PC              ;RETURN TO CALLER WHEN DONE
27
    
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27

013737 015756 017362  
012737 177777 015762  
005037 017256  
004737 046462  
022737 000040 015762  
001410  
013703 015762  
105763 015652  
001765  
012737 000001 017256  
013737 017362 015756  
000207

```
.SBTTL          GET ALL TX COUNTS FROM LIST TO DVTCT
;RETURN WITH DVTCT=1 IF ANY COUNT HAS SOME IN IT
;IF ALL COUNTS ARE ZERO EXIT

;+
; FUNCTIONAL DESCRIPTION:      GATCFL - GET ALL TX COUNTS FROM LIST
; THIS ROUTINE GETS AND CHECKS ALL TX COUNTS TO BE ZERO
; OUTPUTS:      DVTCT      ZERO IF LIST IS ZERO
;                NON ZERO IF NOT
; SUBORDINATE ROUTINES USED:
;                GTVIND      GET NEXT VALID INDEX
; CALLING SEQUENCE:
;                JSR        PC,GATCFL

GATCFL: MOV      TRIBN,TEMP5
        MOV      @ 1,INDEX
        CLR      DVTCT
        JSR      PC,GTVIND      ;CLEAR COUNT
        CMP      @32,INDEX      ;GET VALID INDEX
        BEQ      GATCEX        ;IS INDEX =32 ALL DONE
        MOV      INDEX,R3      ;IF SO EXIT
        TSTB    DVTCLS(R3)     ;IS THIS COUNT 0
        BEQ      GATC1        ;IF THIS ONE IS ZERO
        MOV      @01,DVTCT    ;LOAD COUNT WITH A 1
GATCEX: MOV      TEMPS,TRIBN
        RTS                    ;RETURN TO CALLER
```

```

1      .SBTTL          GET NEXT TX POINTER FROM LIST
2
3      ;**
4      ; FUNCTIONAL DESCRIPTION:      GNTXPR - GET NEXT TX POINTER
5      ; THIS ROUTINE GETS THE NEXT TX POINTER TO CPTR
6      ; OUTPUTS:          CPTR - POINTER FOR NEXT TRANSMIT MESSG
7      ; SUBORDINATE ROUTINES USED:
8      ;          GTVIND -          GET VALID INDEX
9      ; CALLING SEQUENCE:
10     ;          JSR          PC,GNTXPR
11     ;
12     GNTXPR:  CMP          #32.,INDEX          ;IS INDEX = DONE
13             BNE          GNTX1
14     GNTX2:  MOV          #-1,INDEX
15     GNTX1:  JSR          PC,GTVIND
16             CMP          #2.,INDEX
17             BEQ          GNTX2
18             JSR          PC,ULTCLS          ;GET COUNT FROM LIST
19             TST          DVTCT          ;TEST COUNT
20             BEQ          GNTXPR
21             JSR          PC,ULTPLS          ;UNLOAD POINTER
22             RTS          PC          ;RETURN TO CALLER
23
24     .SBTTL          GET INDEX BYTE AND WORD
25
26     ;**
27     ; FUNCTIONAL DESCRIPTION:      GETIND - GET INDEX FOR WORD AND BYTE
28     ; THIS ROUTINE GETS INDEX LOADED WITH INDEX AND INDW WITH INDEX
29     ; FOR WORD. IF TRIBLS ENTRY IS EQUAL TO TRIBN
30     ; OUTPUTS:          INDEX  BYTE INDEX
31     ;          INDW  - WORD INDEX
32     ; CALLING SEQUENCE:
33     ;          JSR          PC,GETIND
34     ;
35
36     GETIND:  MOV          # 1,R3          ;LOAD R3 WITH 1
37             INC          R3          ;BUMP R3
38             CMP          #32.,R3          ;ARE WE ALL DONE
39             BEQ          GETIND          ;IF SO GO BACK
40             CMPB        TRIBLS(R3),TRIBN ;ELSE COMPARE FOR THIS TRIB
41             BNE          GETI1          ;BRANCH IF NO MATCH
42     GETI2:  MOV          R3,INDEX          ;STORE OFF BYTE INDEX
43             ASL          R3          ;MAKE UP WORD INDEX
44             MOV          R3,INDW          ;STORE OFF WORD INDEX
45             RTS          PC          ;RETURN TO CALLER
    
```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 71  
 WRITE DEFAULTS TO TRIB AND GLOBAL SLOTS

```

1      .SBTTL          WRITE DEFAULTS TO TRIB AND GLOBAL SLOTS
2      ;**
3      ; FUNCTIONAL DESCRIPTION:      WRDEFP - WRITE  DEFAULT POLL PARAMETERS
4      ;
5      ;       THIS ROUTINE WRITES ALL POLLIS WITH DEFAULTS AND ALSO
6      ;       WRITES THE GLOBAL LIST WITH DEFAULTS
7      ;
8      ; INPUTS:
9      ;
10     ; CALLING SEQUENCE:
11     ;
12     ;
13     ; WRITE DEFAULT POLL PARMS FOR TRIBS
14
15     WRDEFP: MOV      R2,-(SP)          ;SAVE R2,R3,R4 ON THE STACK
16            MOV      R3,-(SP)
17            MOV      R4,-(SP)
18
19            MOV      #32.,R4
20            MOV      #POLLIS,R3
21     WRDESB: MOV      #POLDEF,R2
22     WRDESA: MOV      (R2)+,(R3)+
23            CMP      #GLBDEF,R2      ;ARE WE THRU ONE SET?
24            BNE      WRDESA
25            DEC      R4
26            BNE      WRDESB
27
28     ; WRITE DEFAULTS FOR GLOBAL
29
30            MOV      #GLBPLS,R3
31            MOV      #GLBDEF,R2
32     WRDESD: MOV      (R2)+,(R3)+
33            CMP      #GLBEND,R2
34            BNE      WRDESD
35            MOV      (SP)+,R4      ;RESTORE R4,R3,R2
36            MOV      (SP)+,R3
37            MOV      (SP)+,R2
38            RTS      PC          ;RETURN TO CALLING ROUTINE
39
40

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22

```

.SBTTL          LOG AND QUE REC BUFFERS
; **
; FUNCTIONAL DESCRIPTION:      LOGAQR - QUE AND LOG RX BUFFERS
;                               THIS ROUTINE QUEUES THE REC BUFFER POINTED TO BY
;                               CPTRR
; INPUTS:                      CPTRR - POINTS TO POINTER TABLE ENTRY
; IMPLICIT OUTPUTS:
;                               BUFFER QUEUED FOR THIS ENTRY
; CALLING SEQUENCE:
;                               JSR      PC,LOGAQR
;
LOGAQR: MOV      CPTRR,R2          ;LOAD R2 FROM POINTER
        MOV      (R2),TEMP2      ;SET UP ADDRESS FOR LOGGING
        MOV      (R2)+,DVRXA     ;SET UP ADDRESS FOR DEVICE
        MOV      (R2),TEMP3     ;SET UP CHAR COUNT FOR LOGGING
        MOV      (R2),DVRCC     ;SET UP COUNT FOR DEVICE
        MOV      R2,CPTRR       ;RESTORE POINTER
        JSR      PC,DVRXQ       ;QUEUE REC BUFFER
        JSR      PC,LOGRXQ      ;LOG RXQ
        RTS      PC            ;RETURN TO CALLER

```



```

1      .SBTTL          SHOW MODE OF OPERATION, LOOP TYPE AND QUALIFIERS
2
3
4      ;**
5      ; FUNCTIONAL DESCRIPTION:
6      ;     SHWOP     SHOW MODE OF OPERATION, LOOP, QUALIFIERS
7      ;               PRINTED ON THE OPERATOR'S CONSOLE.
8
9      ; INPUTS:
10     ;     DEV1=    MODE TYPE (MODTYP)
11     ;     DEV2=    MAINT LOOP TYPE (MLTYP)
12     ;     DEV3=    "RUN PASS" COUNT (RPASS)    COUNT DOWN
13     ;     DEV4=    PARAMTERS WORD (PARAM)
14
15     ; IMPLICIT INPUTS:
16     ;     MODES=   TABLE OF ADDRESSES OF MODE NAME STRINGS
17     ;     LOOPS=   TABLE OF ADDRESSES OF LOOP TYPE NAMES
18
19     ; CALLING SEQUENCE:
20     ;     JSR PC,SHWOP
21     ;--
22 047344 013702 020652      SHWOP:  MOV     DEV1,R2          ;GET THE MODE TYPE IN R2
23 047350 006302              ASL     R2              ;MAKE IT A WORD TABLE OFFSET
24 047352 016237 003344 017350  MOV     MODES(R2),TEMP  ;GET ADDRESS OF MODE IN ASCII
25 047360 013702 020654              MOV     DEV2,R2          ;GET MAINTENANCE LOOP TYPE
26 047364 006302              ASL     R2
27 047366 012737 026766 017356  MOV     @LP00,TEMP3     ;LOAD TEMP3 TO POINT TO "/LOOP="
28 047374 005702              TST     R2              ;SEE IF /LOOP=XXXXX OR NONE
29 047376 001003              BNE    10$             ; BR IF /LOOP= OF SOME KIND
30 047400 012737 026765 017356  MOV     @LP0,TEMP3      ;IF NO LOOP THEN DON'T PRINT /LOOP-
31 047406 016237 003362 017352 10$:  MOV     @OOPS(R2),TEMP1 ;GET ADDRESS OF LOOP-IN ASCII
32 047414 013737 020656 017354  MOV     DEV3,TEMP2      ;GET NUMBER OF PASSES
33 047422              PRINTS @SHF0,TEMP,TEMP3,TEMP1,TEMP2
34 047422 013746 017354              MOV     TEMP2,(SP)
35 047426 013746 017352              MOV     TEMP1,(SP)
36 047432 013746 017356              MOV     TEMP3,(SP)
37 047436 013746 017350              MOV     TEMP,(SP)
38 047442 012746 027531              MOV     @SHF0,(SP)
39 047446 012746 000005              MOV     @5,(SP)
40 047452 010600              MOV     SP,R0
41 047454 104416              TRAP   C$PNTS
42 047456 062706 000014              ADD     @14,SP
43
44 34
45 047462 005002              CLR     R2              ;NOW SET UP FOR QUALIFIERS IN ASCII
46 047464 012737 027045 017350  MOV     @PST,TEMP
47 047472 032737 000001 020660  BIT     @STATB,DEV4     ;SEE IF /STATUS OR /NOSTATUS
48 047500 001003              BNE    1$              ;BR IF /STATUS
49 047502 012737 027043 017350  MOV     @PNST,TEMP
50 047510 012737 027056 017352 1$:  MOV     @PCK,TEMP1
51 047516 032737 000002 020660  BIT     @DATCKB,DEV4    ;SEE IF /CHECK OR /NOCHECK
52 047524 001003              BNE    2$              ;BR IF /CHECK
53 047526 012737 027054 017352  MOV     @PNCK,TEMP1
54 047534 012737 027066 017354 2$:  MOV     @PEC,TEMP2
55 047542 032737 000004 020660  BIT     @ECHO8,DEV4     ;SEE IF /ECHO OR /NOECHO
56 047550 001003              BNE    3$              ;BR IF /ECHO
57 047552 012737 027064 017354  MOV     @PNEC,TEMP2
58

```

CZCLMCO DMP/V 11 DCLT MACRC V05.00 Thursday 22 Mar 84 16:24 Page 73 1  
 SHOW MODE OF OPERATION, LOOP TYPE AND QUALIFIERS

```

65
66 047560 012737 027075 017362 3$: MOV @PMS,TEMP5
67 047566 032737 000010 020660 BIT @MOCHK,DEV4 ;SEE IF MODEM OR /NOMODEM
68 047574 001003 BNE 5$ ;BRANCH IF MODEM
69 047576 012737 027073 017362 MOV @PNMS,TEMP5
70
71 047604 5$: PRINTS @SHF1,TEMP,TEMP1,TEMP2,TEMP5 ;,TEMP3,TEMP4 **RFU**
047604 013746 017362 MOV TEMP5,-(SP)
047610 013746 017354 MOV TEMP2,-(SP)
047614 013746 017352 MOV TEMP1,-(SP)
047620 013746 017350 MOV TEMP,(SP)
047624 012746 027567 MOV @SHF1,-(SP)
047630 012746 000005 MOV @5,-(SP)
047634 010600 MOV SP,RO
047636 104416 TRAP C$PNTS
047640 062706 000014 ADD @14,SP
72 047644 000207 RTS PC ;RETURN
73
74

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

```

.SBTTL          TRAVERSE COMMAND LINE SUBROUTINES

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

P$TRV:
MOV      P$BUFA,R4
MOV      P$TREE,R3
P$TR5:  TSTB   (R4)                                ;SEE IF ANY CHARS LEFT IN INPUT STRING
;BR IF NO
BEQ      P$EXIT
CMPB    (R3),#11.                                ;SEE IF SPECIAL CLI CHAR CODE OR ASCII
BGT     20$
MOVB    (R3),R5
ASL     R5
MOV     10$(R5),R5                               ;BUILD TRAVERSE ROUTINE ADDRESS
ADD     #10$,R5
JSR     PC,(R5)                                  ;JSR TO SPECIAL CLI TRAVERSE ROUTINE
BR      P$TR5                                    ;GO SEE IF MORE OF STRING LEFT

10$:    .WORD  TRVERR-10$                          ;TRAVERSE TABLE FOR "CLI FUNCTIONS"
        .WORD  TRVEXI-10$                          ;1
        .WORD  TRVBR-10$                            ;2
        .WORD  TRVBIF-10$                           ;3
        .WORD  TRVSPA-10$                           ;4
        .WORD  TRVNUM-10$                            ;5
        .WORD  TRVALP-10$                            ;6
        .WORD  TRVALN-10$                            ;7
        .WORD  TRVOCT-10$                            ;8
        .WORD  TRVDEC-10$                            ;9
        .WORD  TRVSTR-10$                           ;10

;NOT A SPECIAL CODE

20$:    CMPB   (R3),(R4)                            ;SEE IF FIRST CHAR OF STRING IS A MATCH
BEQ     22$
JSR     PC,TRVBRC
BR      P$TR5
;IF NOT A MATCH, GO TAKE MISS BRANCH
; THEN GO BACK PT'G TO MISS NODE
22$:    JSR     PC,TRVACT
ADD     #4,R3
;IF A MATCH, GO DO ACTION DEFINED BY
; ACTION CODE IN CLI NODE, THEN
; ADJUST PTR TO NEXT CLI NODE
;ADJUST BUF PTR TO NEXT CHAR IF MATCH
INC     R4
BR      P$TR5

```

```

19 047646
20 047646 013704 003374
21 047652 013703 003376
22 047656 105714
23 047660 001441
24 047662 121327 000013
25 047666 003023
26 047670 111305
27 047672 006305
28 047674 016505 047710
29 047700 062705 047710
30 047704 004715
31 047706 000763
34 047710 000114
35 047712 000134
36 047714 000152
37 047716 000162
38 047720 000204
39 047722 000270
40 047724 000604
41 047726 000650
42 047730 000270
43 047732 000256
44 047734 000736
48 047736 121314
49 047740 001403
50 047742 004737 050006
51 047746 000743
52 047750 004737 047766
53 047754 062703 000004
55 047760 005204
56 047762 000735

```

```

58 047764 000207          P$EXIT: RTS      PC          ;RETURN FROM PARSER
59
60
61
62
63 047766 116302 000001   ;GOTO USER ACTION ROUTINE
TRVACT: MOV      1(R3),R2      ;GET ACTION CODE FROM CLI NODE
64 047772 042702 177400   BIC      #177400,R2        ;CLEAR ANY SIGN EXTENSION
65 047776 013705 003400   MOV      P$ACT,R5         ;GET ADDRESS OF CLI ACTION ROUTINE
66 050002 004715          JSR      PC,(R5)          ;GO DO ACTION DEFINED BY CODE
67 050004 000207          RTS      PC              ;RETURN TO CALLING CODE
68
69
70 050006 016305 000002   ;TAKE BRANCH IN TREE
TRVBRC: MOV      2(R3),R5      ;GET BRANCH DISPLACEMENT FROM TREE
71 050012 060503          ADD      R5,R3           ; AND POINT R3 TO THE "MISS" NODE
72 050014 000207          RTS      PC              ; RETURN TO P$TRV
73
74
75 050016 062703 000004   ;NO BRANCH TAKEN
TRVNOB: ADD      #4,R3        ;THINGS OK, UPDATE R3 TO POINT TO NEXT
76 050022 000207          RTS      PC              ; NODE AND RETURN TO P$TRV
77
78
79 050024 004737 047766   ;TRVERR: JSR      PC,TRVACT  ;TAKE ERROR ACTION
80 050030 112737 177777   MOV      # -1,P$GDBD      ;SET ERROR RETURN FLAG
81 050036 005726          TST      (SP)+            ;GET RID OF "JSR PUSH TO TRVERR"
82 050040 000137 047764   JMP      P$EXIT           ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
83
84 050044 004737 047766   ;TRVEXI: JSR      PC,TRVACT  ;TAKE EXIT ACTION
85 050050 105037 003411   CLRB    P$GDBD            ;SET GOOD/BAD FLAG TO "SUCCESS (0)"
86 050054 005726          TST      (SP)+            ;GET RID OF "JSR PUSH TO TRVEXI"
87 050056 000137 047764   JMP      P$EXIT           ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
88
89 050062 004737 047766   ;TRVBR: JSR      PC,TRVACT  ;GO TAKE BRANCH ACTION
90 050066 000137 050006   JMP      TRVBRC
91
92 050072 004737 047766   ;TRVBIF: JSR      PC,TRVACT
93 050076 105737 003411   TSTB   P$GDBD            ;SEE IF P$GDBD SET OR CLEARED BY ACTION
94 050102 001402          BEQ     1$               ;IF CLEAR FALL THRU TO NEXT NODE
95 050104 000137 050006   JMP     TRVBRC           ;ELSE TAKE THE "MISS" BRANCH
96 050110 000137 050016   1$:   JMP     TRVNOB        ;JUST UPDATE TO NEXT NODE IF THINGS OK
97
98 050114 005005          TRVSPA: CLR      R5       ;CLEAR "SPACE OR TAB FOUND" FLAG
99 050116 121427 000011   1$:   CMPB   (R4),#11      ;SEE IF CHAR. IN CMD LINE = TAB
100 050122 001003          BNE    2$               ;BR IF NO, NOT A TAB
101 050124 005204          INC    R4               ;INC INPUT STRING POINTER
102 050126 005205          INC    R5               ;INDICATE A TAB FOUND
103 050130 000772          BR     1$               ;GO CHECK NEXT CHAR
104
105 050132 121427 000040   2$:   CMPB   (R4),#40      ;SEE IF CHAR. IN CMD LINE = SPACE
106 050136 001003          BNE    10$             ;BR IF NO, NON-SPACE OR NON TAB CHAR.
107 050140 005204          INC    R4               ;INC INPUT STRING POINTER
108 050142 005205          INC    R5               ;INDICATE A SPACE FOUND
109 050144 000764          BR     1$               ;GO CHECK NEXT CHAR
110 050146 005705          10$:  TST    R5               ;SEE IF ANY SPACES OR TABS FOUND
111 050150 001404          BEQ    15$             ;BR IF NO, TAKE NO ACTION
112 050152 004737 047766   JSR    PC,TRVACT        ;GO TAKE ACTION IF ANY FOUND
113 050156 000137 050016   JMP    TRVNOB           ;JUST GO UPDATE R3 TO NEXT NODE IF OK
114 050162 000137 050006   15$:  JMP    TRVBRC           ;TAKE BRANCH (MISS) IF NONE FOUND

```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22-Mar-84 16:24 Page 74 2  
 TRAVERSE COMMAND LINE SUBROUTINES

```

115
116
117 050166 012737 000012 003406 TRVDEC: MOV    #10.,P$RADX      ;USE DECIMAL AS RADIX AND ASSUME *
118 050174 000137 050206          JMP    TRVNMA
119 050200          TRVOCT: ;(SAME AS TRVNUM SINCE DEFAULT RADIX IS OCTAL)
120 050200 012737 000010 003406 TRVNUM: MOV    #8.,P$RADX      ;USE OCTAL AS RADIX AND ASSUME *
121 050206 005005          TRVNMA: CLR    R5              ;CLEAR DIGIT COUNTER
122 050210 121427 000053          CMPB   (R4),#0' *      ;SEE IF THERE'S A * SIGN THERE
123 050214 001001          BNE    10$           ; BR IF NO
124 050216 000406          BR     11$           ; ELSE P$RADX ALREADY SAYS *, JUST BR
125 050220 121427 000055          10$:  CMPB   (R4),#0' -  ;SEE IF THERE'S A - SIGN THERE
126 050224 001004          BNE    11$           ; BR IF NO
127 050226 11273  177777 003407          MOVB   #-1,P$RADX+1   ;SET "MINUS FLAG" (HI BYTE OF P$RADX)
128 050234 00520          11$:  INC    R4              ;BUMP R4 TO POINT TO FIRST CHAR
129
130 050236 121427 000060          1$:   CMPB   (R4),#60     ;SEE IF CHAR. LESS THAN A "0"
131 050242 002434          BLT    2$           ;BR IF YES (NOT NUMERIC)
132 050244 121427 000067          CMPB   (R4),#67     ;SEE IF CHAR. GREATER THAN A '7"
133 050250 003426          BLE    13$          ; BR IF YES
134 050252 123727 003406 000012          CMPB   P$RADX,#10.   ;SEE IF IN DECIMAL MODE
135 050260 001417          BEQ    12$          ; BR IF YES (CAN USE HIGHER LIMIT)
136 050262 121427 000071          CMPB   (R4),#71     ;SEE IF DIGIT WAS A 8 OR 9
137 050266 003022          BGT    2$           ;BR IF NON NUMERIC
138 050270          PRINTF #CLIBRX    ;ELSE WAS A 8 OR 9 WHEN IN OCTAL RADIX
139 050270 012746 023457          MOV    #CLIBRX,(SP)
140 050274 012746 000001          MOV    #1,-(SP)
141 050300 010600          MOV    SP,R0
142 050302 104417          TRAP   C$PNTF
143 050304 062706 000004          ADD    #4,SP
139 050310 112737 177777 003411          MOVB   #1,P$GDBD    ;SET ERROR RETURN FLAG
140 050316 000474          BR     5$           ; PRINT ERROR AND TAKE MISS
141
142 050320 121427 000071          12$:  CMPB   (R4),#71     ;SEE IF CHAR. GREATER THAN A "9"
143 050324 003003          BGT    2$           ;BR IF YES (NOT NUMERIC)
144 050326 005204          13$:  INC    R4           ;UPDATE CMD LINE PTR TO NEXT CHAR.
145 050330 005205          INC    R5           ;INDICATE A NUMERIC FOUND
146 050332 000741          BR     1$           ;GO LOOK AT NEXT CHAR.
147
148 050334 005705          2$:   TST    R5           ;SEE IF FOUND ANY NUMERICS
149 050336 001464          BEQ    5$           ;BR IF NO, TAKE "MISS" BRANCH
150 050340 010401          MOV    R4,R1        ;GET POINTER TO START OF NUMERIC STRING
151 050342 160501          SUB    R5,R1
152 050344 005037 003404          CLR    P$NUM        ;CLEAR LOC. WHERE VALUE WILL BE STORED
153 050350 112102          3$:   MOVB   (R1),R2      ;GET ASCII CHAR AND CONVERT IT TO A #
154 050352 162702 000060          SUB    #60,R2
155 050356 006337 003404          ASL    P$NUM        ;SHIFT CURRENT VALUE TO MAKE ROOM
156 050362 103437          BCS    7$           ;ERROR IF NUMBER TOO BIG
157 050364 013737 003404 003402          MOV    P$NUM,P$CNT  ;SAVE FOR LATER IN CASE DECIMAL RADIX
158 050372 006337 003404          ASL    P$NUM
159 050376 103431          BCS    7$           ;ERROR IF NUMBER TOO BIG
160 050400 006337 003404          ASL    P$NUM
161 050404 103426          BCS    7$           ;ERROR IF NUMBER TOO BIG
162 050406 123727 003406 000012          CMPB   P$RADX,#10.   ;SEE IF DECIMAL RADIX
163 050414 001004          BNE    4$           ;BR IF NOT EQUAL
164 050416 063737 003402 003404          ADD    P$CNT,P$NUM
165 050424 103416          BCS    7$           ;ERROR IF NUMBER TOO BIG
166 050426 060237 003404          4$:   ADD    R2,P$NUM

```

167	050432	103413			BCS	7:			;ERROR IF NUMBER TOO BIG
168	050434	005305			DEC		R5		
169	050436	001344			BNE		3:		
170	050440	105737	003407		TSTB		P\$RADX-1		;SEE IF NUM WAS PRECEDED BY A SIGN
171	050444	001402			BEQ		15:		; BR IF NO
172	050446	005437	003404		NEG		P\$NUM		; ELSE NEGATE THE NUMBER BEFORE LEAVING
173	050452	004737	047766		JSR	15:	PC,TRVACT		;SINCE NUMERIC FOUND, GO TAKE ACTION
174	050456	000137	050016		JMP		TRVNOB		;GO POINT R3 TO NEXT NODE
175									
176	050462					7:	PRINTF	@CLINBG	;PRINT NUMBER TOO BIG ERROR
	050462	012746	023435						MOV @CLINBG, (SP)
	050466	012746	000001						MOV @1, (SP)
	050472	010600							MOV SP,R0
	050474	104417							TRAP C\$PNTF
	050476	062706	000004						ADD @4,SP
177	050502	112737	177777	003411	MOVB		@-1,P\$GDBD		;SET ERROR RETURN FLAG
178	050510	000137	050006		JMP	5:	TRVBRC		;TAKE "MISS" BRANCH
179									
180									
181	050514	005005			TRVALP: CLR		R5		;CLEAR ALPHA FOUND FLAG
182	050516	121427	000101		1:		(R4),@101		;SEE IF CHAR. LESS THAN A "A"
183	050522	002406			BLT		2:		;BR IF YES (NOT ALPHA)
184	050524	121427	000132		CMPB		(R4),@132		;SEE IF CHAR. GREATER THAN A 'Z
185	050530	003003			BGT		2:		;BR IF YES (NOT ALPHA)
186	050532	005204			INC		R4		;UPDATE CMD LINE PTR TO NEXT CHAR
187	050534	005205			INC		R5		;INDICATE AN ALPHA WAS FOUND
188	050536	000767			BR		1:		;GO LOOK AT NEXT CHAR.
189	050540	005705			2:		TST R5		;SEE IF ANY ALPHA'S WERE FOUND
190	050542	001404			BEQ		3:		;BR IF NO
191	050544	004737	047766		JSR		PC,TRVACT		;IF ANY FOUND TAKE ACTION
192	050550	000137	050016		JMP		TRVNOB		;THEN UPDATE R3 TO NEXT NODE NO BRANCH
193	050554	000137	050006		JMP	3:	TRVBRC		;NONE FOUND. TAKE MISS BRANCH
194									
195	050560	005005			TRVALN: CLR		R5		;CLEAR ALPHANUM FOUND FLAG
196	050562	121427	000060		10:		(R4),@60		;SEE IF CHAR. LESS THAN A "0"
197	050566	002417			BLT		2:		;BR IF YES (NOT NUMERIC OR ALPHA)
198	050570	121427	000072		CMPB		(R4),@72		;SEE IF CHAR. GREATER THAN A "9"
199	050574	003003			BGT		1:		;BR IF YES (NOT NUMERIC)
200	050576	005204			INC		R4		;UPDATE CMD LINE PTR TO NEXT CHAR.
201	050600	005205			INC		R5		;INDICATE A NUMERIC FOUND
202	050602	000767			BR		10:		;GO LOOK AT NEXT CHAR.
203	050604	121427	000101		1:		CMPB (R4),@101		;SEE IF CHAR. LESS THAN A "A"
204	050610	002406			BLT		2:		;BR IF YES (NOT ALPHA)
205	050612	121427	000132		CMPB		(R4),@132		;SEE IF CHAR. GREATER THAN A 'Z
206	050616	003003			BGT		2:		;BR IF YES (NOT ALPHA)
207	050620	005204			INC		R4		;UPDATE CMD LINE PTR TO NEXT CHAR
208	050622	005205			INC		R5		;INDICATE AN ALPHA FOUND
209	050624	000756			BR		10:		;GO LOOK AT NEXT CHAR.
210	050626	005705			2:		TST R5		;SEE IF ANY ALPHANUM'S WERE FOUND
211	050630	001404			BEQ		3:		;BR IF NO
212	050632	004737	047766		JSR		PC,TRVACT		;IF ANY FOUND TAKE ACTION
213	050636	000137	050016		JMP		TRVNOB		;THEN UPDATE R3 TO NEXT NODE NO BRANCH
214	050642	000137	050006		JMP	3:	TRVBRC		;NONE FOUND. TAKE MISS BRANCH
215									
216									
217									
218	050646	010401			TRVSTR: MOV		R4,R1		;POINT R1 TO CMD STRING

CZCLMCO DMP, v 11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 74 4  
 TRAVERSE COMMAND LINE SUBROUTINES

219	050650	010305		MOV	R3,R5		
220	050652	062705	000006	ADD	#6,R5		;POINT R5 TO MATCH STRING FROM CLI NODE
221	050656	005037	003402	CLR	P#CNT		;CLEAR CHAR MATCH COUNT
222	050662	105715		2\$: TSTB	(R5)		;SEE IF END OF MATCH STRING YET
223	050664	001411		BEQ	10\$		;BR IF YES
224	050666	105711		TSTB	(R1)		;SEE IF END OF CMD LINE YET
225	050670	001407		BEQ	10\$		;BR IF YES
226	050672	121115		CMPB	(R1),(R5)		;SEE IF CHARACTERS MATCH
227	050674	001005		BNE	10\$		;BR IF NO
228	050676	005237	003402	INC	P#CNT		;MATCH -INCREMENT MATCH COUNT
229	050702	005201		INC	R1		;UPDATE STRING POINTERS
230	050704	005205		INC	R5		
231	050706	000765		BR	2\$		;BR TO CONTINUE CHECKING CHARS.
232							
233	050710	005737	003402	10\$: TST	P#CNT		;WHEN DONE SEE IF ANY MATCHES FOUND
234	050714	001406		BEQ	15\$		;BR IF NO, GO TAKE THE MISS BRANCH
235	050716	010104		MOV	R1,R4		;POINT CMD POINTER TO END OF STRING &
236	050720	004737	047766	JSR	PC,TRVACT		;IF A MATCH FOUND, GO DO MATCH ACTION
237	050724	066303	000004	ADD	4(R3),R3		;UPDATE R3 TO NEXT NODE (NO BRANCH)
238	050730	000207		RTS	PC		; (NO RETURN THRU TRVNOB SINCE DIFFERENT
239							; DISPLACEMENT DUE TO MATCH STRING)
240	050732	000137	050006	15\$: JMP	TRVBRC		; GO TAKE BRANCH
241							
242							
243							; (PARSED OK), 1 IF ILL CMD.....
244							

CZCLMCO DMP/V 11 DCLT  
REPORT CODING SECTION

MACRO V05.00 Thursday 22 Mar-84 16:24 Page 75

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
22  
23  
24  
25  
32  
33  
40  
41

.SBTTL REPORT CODING SECTION

\*\*\*  
; THE REPORT CODING SECTION CONTAINS THE  
; 'PRINTS" CALLS THAT GENFRATE STATISTICAL REPORTS.  
;

050736  
050736

BGNRPT

L\$RPT::

050736 004737 042554

JSR PC,REPORT

;CALL SUBROUTINE TO DUMP EVENT LOG  
; AND BASE TABLE

050742  
050742  
050742 104425

ENDRPT

L10011: TRAP C\$RPT



```

1      .SBTTL  PROTECTION TABLE
2
3      ;**
4      ; THIS TABLE IS USED BY THE RUNTIME SERVICES
5      ; TO PROTECT THE LOAD MEDIA.
6      ;
7
8 050744      BGNPROT
9 050744
10 050744 177777
11 050746 177777
12 050750 177777
13
14 050752
15

```

				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		

```

1          .SBTTL  INITIALIZE SECTION
2
3
4          ;
5          ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
6          ; AT THE BEGINNING OF EACH PASS.
7          ;
8 050752          BGNINIT
9 050752
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34 050752 005737 017376          TST      DCLFLG          ;IS DOCLEAN SET
35 050756 001403                    BEQ      INIT1          ;BRANCH IF NOT
36 050760 005037 017376          CLR      DCLFLG          ;IF SET CLEAR IT
37 050764          DOCLN
38 050766 005037 017416          CLR      RUNNING          ; INIT "DCLT RUNNING" FLAG
39 050772 012737 177777 017400  MOV      @-1,RESFLG      ;SET RESTART FLAG
40 051000          READDEF @EF.START ;IF HERE CAUSE OF START,DO SOME INIT
41 051000 012700 000040          MOV      @EF.START,R0
42 051004 104447          TRAP     C@REFG
43 051006          BCOMPLETE      START
44 051006 103417          READDEF @EF.RESTART ;IF HERE CAUSE OF RESTART, DO SOME INIT
45 051010          BCOMPLETE      RESTRT
46 051010 012700 000037          MOV      @EF.RESTART,R0
47 051014 104447          TRAP     C@REFG
48 051016          BCOMPLETE      RESTRT
49 051016 103514          READDEF @EF.CONTINUE ;SEE IF WE RE HERE CAUSE OF A CONTINUE
50 051020          BCOMPLETE      S1
51 051020 012700 000036          MOV      @EF.CONTINUE,R0
52 051024 104447          TRAP     C@REFG
53 051026          BCOMPLETE      S1
54 051026 103002          JMP      ENDIT
55 051030 000137 051652          READDEF @EF.NEW ;JMP IF HERE CAUSE OF A CONTINUE
56 051034          BCOMPLETE      NEW
57 051034 012700 000035          MOV      @EF.NEW,R0 ;SEE IF THIS IS A "NEW PASS"
58 051040 104447          TRAP     C@REFG
59 051042          BCOMPLETE      SETUP
60 051042 103522          BR      GETPRM ;IF YES, BR AROUND LOGUNIT @ SETUP
61 051044 000524          BR      GETPRM
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

62 051112          S2:  CLOCK  P,R1          ;LOOK FOR A P CLOCK SINCE NO LINE CLOCK
    051112 012700 000120          ;MOV @P,RO
    051116 104462          ;TRAP C#CLK
    051120 010001          ;MOV RO,R1
63 051122          BNCOMPLETE  S3          ; IF NONE THERE GO SEE IF THIS IS LSI
    051122 103017          ;BCC S3
64 051124 004737 041654          JSR  PC,CLKSET          ; ELSE GO SET UP CLOCK INFO & VECTOR
65 051130 062737 000002 017436  ADD  #2,CLKCSR          ;POINT CLKCSR TO P-CLK COUNT SET REG.
66 051136 012777 001600 146272  MOV  #PCLKCT,@CLKCSP    ;LOAD CLK SET REG. WITH COUNT VALUE
67 051144 162737 000002 017436  SUB  #2,CLKCSR          ;POINT CLKCSR BAC TO P CLK CSR
68 051152 012737 000111 017446  MOV  #PCLKEN,CLKEN     ;SETUP THE ENABLE THE P-CLK DATA
69 051160 000433          BR    RESTR
70
71 051162          S3:  READBUS          ;READ BUS TYPE TO SEE IF ON AN LSI
    051162 104407          ;TRAP C#RD&U
72 051164          BNCOMPLETE  S4          ;BR IF NOT, NO CHANCE OF A CLOCK
    051164 103021          ;BCC S4
73 051166 012737 000100 017442  MOV  #100,CLKVEC       ;LOAD 100 AS CLK VECTOR
74 051174 005037 017440          CLR  CLKBR             ;LOAD 0 AS CLK INT. LEVEL
75 051200 012737 017446 017436  MOV  #CLKEN,CLKCSR     ;KLUDGE UP THE CSR & ENABLE DATA LOCS
76 051206          GMANID  L5060,CLKHZ,D,377,50.,60.,YES
    051206 104443          ;TRAP C#GMAN
    051210 000406          ;BR 10000$
    051212 017444          ;.WORD CLKHZ
    051214 000052          ;.WORD T#CODE
    051216 027121          ;.WORD L5060
    051220 000377          ;.WORD 377
    051222 000062          ;.WORD T#LOLIM
    051224 000074          ;.WORD T#HILIM
    051226          ;10000$:
77 051226 000410          BR    RESTR
78
79
80 051230          S4:  PRINTF  #BDCLK          ;MOV #BDCLK, (SP)
    051230 012746 027232          ;MOV #1, -(SP)
    051234 012746 000001          ;MOV SP,RO
    051240 010600          ;TRAP C#PNIF
    051242 104417          ;ADD #4,SP
    051244 062706 000004          ;CLEAR TIME SINCE START LOCATIONS
81 051250 005037 017450          RESTRT: CLR  TIMMIN          ;LOAD TICKS/SEC
82 051254 005037 017452          CLR  TIMSEC          ;INIT EVENT TABLE TO ALL 1 S AFTER EACH
83 051260 013737 017444 017454  MOV  CLKHZ,TIMTCK     ; START OR RES AND INIT TABLE POINTER
84 051266 012702 017466          MOV  #EVTLOG,R2
85 051272 010237 017464          MOV  R2,EVTPTIR
86 051276 012722 177777          1$:  MOV  #1,(R2)
87 051302 020227 020522          CMP  R2,#EVTEND
88 051306 001373          BNE  1$
89
90 051310 012737 177777 017372  NEW:  MOV  #1,LOGUNT          ;INITIALIZE LOGICAL UNIT #
91
92 051316 005237 017372          GETPRM: INC  LOGUNT          ;POINT TO NEXT LOGICAL UNIT
93 051322 023737 017372 002012  CMP  LOGUNT,L#UNIT     ;SEE IF PAST MAX. LOG. UNIT #
94 051330 002367          BGE  NEW              ;BR IF YES, AND START OVER
95
96 051332          GPWARD  LOGUNT,R1          ;GET THE P TABLE FOR THIS LOG. UNIT
    051332 013700 017372          ;MOV LOGUNT,R0
    051336 104442          ;TRAP C#GPWARD

```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 77 2  
 INITIALIZE SECTION

```

    97 051340 010001
    98 051342          BNCOMPLETE      GETPRM          ;IF NO P TABLE AVAIL., GO GET NEXT ONE
    99 051342 103365          ;MOV      R0,R1
    99 051344 011137 017406  MOV      (R1),FMDPLX          ;PUT FULL OR HALF DUPLEX ANSWER IN LOC.
    100
    113
    114          ;DEVICE DEPENDENT PART OF GETTING INFO FROM P TABLE
    115
    116 051350 016137 000002 023052  MOV      2(R1),SELO          ;STORE AWAY CSR ADDRESSES
    117 051356 016137 000002 023054  MOV      2(R1),BSEL1
    118 051364 005237 023054          INC      BSEL1
    119 051370 016137 000002 023056  MOV      2(R1),SEL2
    120 051376 062737 000002 023056  ADD      #2,SEL2
    121 051404 016137 000002 023060  MOV      2(R1),BSEL3
    122 051412 062737 000003 023060  ADD      #3,BSEL3
    123 051420 016137 000002 023062  MOV      2(R1),SEL4
    124 051426 062737 000004 023062  ADD      #4,SEL4
    125 051434 016137 000002 023064  MOV      2(R1),BSEL5
    126 051442 062737 000005 023064  ADD      #5,BSEL5
    127 051450 016137 000002 023066  MOV      2(R1),SEL6
    128 051456 062737 000006 023066  ADD      #6,SEL6
    129 051464 016137 000002 023070  MOV      2(R1),BSEL7
    130 051472 062737 000007 023070  ADD      #7,BSEL7
    131
    132 051500 016137 000004 023072  MOV      4(R1),INVEC          ;STORE AWAY INPUT INTERRUPT VECTOR
    133 051506 016137 000004 023074  MOV      4(R1),OUTVEC
    134 051514 062737 000004 023074  ADD      #4,OUTVEC          ;BUILD OUTPUT INTERRUPT VECTOR
    135 051522 016137 000006 023076  MOV      6(R1),INTPRI          ;STORE AWAY INTERRUPT PRIORITY
    136 051530 016137 000010 023102  MOV      10(R1),DEVPAR          ;STORE AWAY PARAMS
    137 051536 016137 000012 023100  MOV      12(R1),OPTYP          ;STORE AWAY DEVICE OPTION TYPE
    138 051544 032737 000003 023100  BIT      #3,OPTYP          ;IS THIS A DMV
    139 051552 001417          BEQ      11$
    140 051554 012737 000454 016202  MOV      #300.,DMVDF1
    141 051562 012737 001130 016204  MOV      #600.,DMVDF2
    142 051570 012737 001130 016210  MOV      #600.,DMVDF3
    143 051576 012737 000024 016212  MOV      #24,DMVDF4
    144 051604 012737 001750 016214  MOV      #1000.,DMVDF5          ;SET UP DMV DEFAULTS
    145 051612 005037 023104          11$: CLR      STATYP          ;CLEAR STATION TYPE
    146 051616 032737 000001 023102  BIT      #MTP,DEVPAR          ;IS THIS MULTIPOINT
    147 051624 001407          BEQ      1$
    148 051626 052737 000004 023104  BIS      #BIT2,STATYP          ;IF MULTIPOINT SET BIT
    149 051634 032737 000002 023102  BIT      #TRBB,DEVPAR          ;IS THIS A TRIB
    150 051642 001003          BNE      ENDIT
    151 051644 052737 000002 023104  1$: BIS      #BIT1,STATYP          ;SET STATION TYPE
    152 051652          ENDIT:
    153 051652          SETVEC  CLKVEC,#CLKINT,#340          ;SETUP CLOCK VECTOR
    154          MOV      #340.(SP)
    155          MOV      #CLKINT.(SP)
    156          MOV      CLKVEC.(SP)
    157          MOV      #3.(SP)
    158          TRAP  C$VEC
    159          ADD      #10,SP
    165 051700          ;DEVICE DEPENDENT VECTOR SETUP
    165 051700          SETVEC  INVEC,#DVINS,INTPRI          ;SETUP INPUT INTERRUPT VECTOR
    
```

CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 77 3  
INITIALIZE SECTION

	051700	013746	023076				MOV	INTPRI, -(SP)
	051704	012746	066656				MOV	#DVINS, (SP)
	051710	013746	023072				MOV	INVEC, -(SP)
	051714	012746	000003				MOV	#3, -(SP)
	051720	104437					TRAP	C\$SVEC
	051722	062706	000010				ADD	#10, SP
166	051726			SETVEC	OUTVEC, #DVOUTS, INTPRI	; SETUP OUTPUT INTERRUPT VECTOR		
	051726	013746	023076				MOV	INTPRI, (SP)
	051732	012746	066674				MOV	#DVOUTS, -(SP)
	051736	013746	023074				MOV	OUTVEC, -(SP)
	051742	012746	000003				MOV	#3, (SP)
	051746	104437					TRAP	C\$SVEC
	051750	062706	000010				ADD	#10, SP
167								
168	051754			SETPRI	#PRI00	; SET THE "RUN" PRIORITY TO 0		
	051754	012700	000000				MOV	#PRI00, R0
	051760	104441					TRAP	C\$SPRI
169	051762			EXIT	INIT			
	051762	104432					TRAP	C\$EXIT
	051764	000002					.WORD	L10013 .
170								
182								
183				.EVEN				
184								
185	051766			ENDINIT				
	051766					L10013:		
	051766	104411					TRAP	C\$INIT

1  
2  
3  
4  
5  
6  
7  
8  
9  
10 051770  
051770  
11  
18  
19 051770  
051770  
051770 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

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 79  
 CLEANUP CODING SECTION

```

1          .SBTTL  CLEANUP CODING SECTION
2
3          ;**
4          ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
5          ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
6          ;
7
8 051772          BGNCLN
9 051772          L$CLEAN::
10
11 18 051772 005037 017416          CLR  RUNING          ; INIT DCLT RUNNING FLAG
12 19 051776 012737 177777 017320  MOV  @ 1,CLNSET      ; SET THE CLEANUP FLAG
13 20 052004 004737 064776          JSR  PC,HLTTRB      ; HALT ALL TRIBS
14 21 052010 005037 017320          CLR  CLNSET
15 22 052014 105037 015756          CLR  TRIBN          ;
16 23 052020 005037 017350          CLR  TEMP          ; MODEM SIGNALS TO CLEAR
17 24 052024 004737 064732          JSR  PC,WRMCS      ; GO CLEAR MODEM SIGNALS
18 25 052030 005077 145402          CLR  @CLKCSR       ; DISABLE CLOCK
19 26 052034          SETPRI @PRI07          ; SET PROCESSOR PRIORITY BACK TO 7
20 052034 012700 000340          MOV  @PRI07,RO
21 052040 104441          TRAP  C$SPRI
22
23 27 052042          EXIT  CLN
24 052042 104432          TRAP  C$EXIT
25 052044 000002          .WORD L10015 .
26
27 28
28 40
29 41          .EVEN
30 42
31 43 052046          ENDCLN
32 052046          L10015:
33 052046 104412          TRAP  C$CLEAN

```

```

1          .SBTTL  DROP UNIT SECTION
2
3          ;**
4          ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
5          ; TO NO LONGER BE TESTED.
6          ;--
7
8 052050          BGNDU
9                L$DU::
10
11
12
13
14
15
16
17
18
19 052050          EXIT  DU
20 052050          000167          .WORD  J$JMP
21 052052          000000          .WORD  L10016 2 .
22
23
24
25
26
27
28
29
30
31
32
33          .EVEN
34
35 052054          ENDDU
36                L10016:
37                TRAP  C$DU

```



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
19  
20  
21  
33  
34  
35  
36  
37  
38

.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.  
;

052056  
052056

BGNAU

L\$AU::

052056 000167  
052060 000000

EXIT AU

.WORD JSJMP  
.WORD L10017-2-

052062  
052062 104452  
052062

ENDAU

L10017: TRAP C\$AU

```

1      .SBTTL TEST 1: SETUP AND MODES OF OPERATION
2
3
4
5      ;**
6      ; TEST TO DETECT FAULTS IN THE DATA COMMUNICATION LINK. THIS TEST WILL
7      ; THE PROVIDE COVERAGE NECESSARY TO ISOLATE FAILURES TO THE COMPUTER
8      ; EQUIPMENT, THE COMMUNICATION LINK, OR THE MODEM.
9      ;--
10
11
12
13
14
15
16
17
18
19
20
21
22
23      052064      BGNTST
24      052064
25
26
27
28
29
30
31      .SBTTL      PROGRAM SETUP SECTION
32
33      052064      013777      017446      145344      MOV      CLKEN,@CLKCSR      ;ENABLE THE CLOCK
34
35      052072
36      052072      005001      GTRAXB:
37      052074      012737      000001      017456      GTRA2:  CLR      R1
38      052102      005737      017456      1$:      MOV      @1,TIMER1      ;SET TIMER TO COUNT 1 TICK
39      052106      001412      TST      TIMER1      ;CHECK FOR IT TO BE COUNTED OFF
40      052110      005301      BEQ      GTRA3      ;BRANCH IF CLOCK EXISTS (COUNTED A TICK)
41      052112      001373      DEC      R1
42      052114      PRINTF  @NOCLK      ;KEEP CHECKING UNTIL R1 DOES FULL COUNTDOWN
43      052114      012746      027256      MOV      @NOCLK,(SP)
44      052120      012746      000001      MOV      @1,-(SP)
45      052124      010600      MOV      SP,RO
46      052126      104417      TRAP    C$PNTF
47      052130      062706      000004      ADD     @4,SP
48
49      052134      005737      017400      GTRA3:  TST      RESFLG      ;SEE IF HERE AFTER A RESTART.
50      052140      001120      BNE      GTRA5      ;BR IF HERE CAUSE OF A RESTART
51
52      ; CLEAR COUNTS AND SET UP DEFAULTS
53
54      052142      005037      017344      GTRA4:  CLR      TOTCC      ;CLEAR TOTAL CHAR. COUNT TEMP. LOC.
55      052146      005037      017262      CLR      TTOTCC      ; CLEAR TOTAL CHAR. COUNT FOR TX BUFF
56      052152      005037      017244      CLR      CTOTCC      ; CLEAR TOTAL CHAR. COUNT FOR CMP BUFF
57      052156      012737      011416      017236      MOV      @PTRTAB, TXPTR ;INIT TRANSMIT MESSAGE POINTER
58
59      052164      005037      017234      CLR      RXPTR      ; ZERO RX POINTER
60      052170      012737      011512      017240      MOV      @PTR13,CMPPTR ;INIT COMP POINTER
61
62      052176      012737      000005      017332      MOV      @5,MSGTYP    ;SET UP DEFAULT MSG TYPE (QUICK FOX -- ITEMP MSG)
63      052204      013737      002162      017334      MOV      MSG5C,CURCC  ;SET UP DEFAULT CHAR COUNT
64      052212      012737      003416      017264      MOV      @TXBUF,TCURAD ;SET UP CURRENT ADDR TO START OF TX BUFFER
65      052220      012737      004416      017246      MOV      @CMPBUF,CCURAD ;SET UP CURRENT ADDR TO START OF CMP BUFFER
66
67      052226      013737      017264      017342      MOV      TCURAD,CURADD ;SETUP CURRENT ADDR TO START OF TXBUF
68      052234      013737      017236      017340      MOV      TXPTR,CPTR   ;SETUP CURRENT POINTER TABLE POINTER FOR TXBUF
69      052242      004737      045446      JSR      PC,BLDBUF    ; GO BUILD POINTER TABLE AND BUFFER
70      052246      012737      000001      017260      MOV      @1,TXMTOT    ;BUMP TOTAL MESSAGE COUNT
71
72      052254      013737      017240      017340      MOV      CMPPTR,CPTR  ;SET UP START OF COMPARE POINTER TABLE
  
```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 82 1  
PROGRAM SETUP SECTION

```

68 052262 013737 017246 017342      MOV      CCURAD,CURADD      ;SET UP CURRENT ADDR. TO START OF CMPBUF
69 052270 012737 000005 017332      MOV      #5,MSGTYP
70 052276 013737 002162 017334      MOV      MSG5C,CURCC
71 052304 004737 045446                JSR      PC,HLDBUF          ;PUT DEFAULT MESSAGE INTO CMPBUF
72 052310 012737 000001 017242      MOV      #1,CMPTOT         ;BUMP THE COMP MSG COUNT
73 052316 012737 000003 017402      MOV      #ACT,MODTYP       ;SET DEFAULT MODE= ACTIVE
74 052324 005037 017404                CLR      MLTYP              ;SET DEFAULT MAINTENANCE LOOP MODE =NONE
75 052330 012737 000001 017412      MOV      #1,RPASS          ;SET UP DEFAULT "RUN PASS" COUNT TO 1
76 052336 012737 000002 017410      MOV      #2,PARAM          ;SET UP PROG. PARAMETERS DATA CHECKING ENABLD
77                                     ;OPERATOR STATUS MSGS. PRINT OFF
78 052344 012737 000061 003252      MOV      #KTRB,KEYWD1      ;SET UP KEYWRD.
79 052352 004737 055472                JSR      PC,ACTKAL          ;ZERO TRIB LIST
80
81 052356 004737 047214                JSR      PC,WRDEFP          ;GO WRITE DEFAULTS TO TRIBS
82
83 052362                                PRINTF   #HLPO
      052362 012746 024040                MOV      #HLPO,-(SP)
      052366 012746 000001                MOV      #1,(SP)
      052372 010600                MOV      SP,RO
      052374 104417                TRAP     C$PNTF
      052376 062706 000004                ADD      #4,SP
84 052402 010637 017364                GTRAS:  MOV      SP,SAVSP      ;SAVE OFF STACK
85 052406 013737 017402 020652      MOV      MODTYP,DEV1
86 052414 013737 017404 020654      MOV      MLTYP,DEV2
87 052422 013737 017412 020656      MOV      RPASS,DEV3
88 052430 013737 017410 020660      MOV      PARAM,DEV4
89 052436 004737 047344                JSR      PC,SHWOP          ;PRINT TO OPERATOR THE CURRENT MODE.....
90
91 052442                                MANUAL
      052442 104450                                ;SEE IF MANUAL INTERVENTION ALLOWED
92 052444                                BCOMPLETE      GETCL      ; BR IF YES (UAM=0 AND NOT CHAINED)
      052444 103412                                TRAP     C$MANI
93 052446 005737 017412                TST      RPASS              ;SEE IF THIS IS FIRST "DCLT PASS"
94 052452 001002                BNE      1$                  ; BR IF NOT COMPLETED 1 PASS
95 052454                                EXIT      TST                  ; IF DONE 1 PASS IN UNATTENDED MODE
      052454 104432                                TRAP     C$EXIT
      052456 014226                                .WORD   L10020-.
96 052460 012737 000001 017404 1$:    MOV      #TTL,MLTYP         ;SET UP DEFAULT FOR UNATTENDED MODE
97 052466 000137 057056                JMP      GTR9                ; "R M=ACT/LO=I/PAS=1/NOST/CH" AND RUN
98
99                                .SBTTL      COMMAND LINE FETCH & INTERPRETATION SECTION
100
101 052472 105037 003411                GETCL:  CLRB   P$GDBD          ;CLEAR CMD LINE PARSING ERROR FLAGS
102 052476 105037 003410                CLRB   P$NNUF
103 052502                                GMANID   CLI$PM,CMDBUF,A,0.1.72.,NO      ;GET A COMMAND LINE FROM OPR.
      052502 104443                                TRAP     C$GMAN
      052504 000406                                BR       10000$
      052506 003130                                .WORD   CMDBUF
      052510 000142                                .WORD   T$CODE
      052512 023354                                .WORD   CLI$PM
      052514 000000                                .WORD   0
      052516 000001                                .WORD   T$LOLIM
      052520 000110                                .WORD   T$HILIM
      052522                                10000$:
104 052522 012737 003130 003374      MOV      #CMDBUF,P$BUFA
105 052530 012737 021170 003376      MOV      #CLITRE,P$TREE
106 052536 012737 053436 003400      MOV      #CLIACT,P$ACT

```

CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 22-Mar 84 16:24 Page 82 2  
 COMMAND LINE FETCH & INTERPRETATION SECTION

```

107 052544 005037 003254          CLR      QUALFG          ;CLEAR QUALIFIER FLAG LOCATION
108 052550 004737 047646          JSR      PC,P$TRV        ;GO PARSE COMMAND LINE
109 052554 105737 003411          TSTB    P$GDBD          ;SEE IF PARSED OK OR AN ERROR
110 052560 001412
111 052562          PRINTF  $CLIERM
                                MOV      $CLIERM, (SP)
                                MOV      $1, (SP)
                                MOV      SP,RO
                                TRAP    C$PNTF
                                ADD      $4,SP
112 052602 000137 052472          JMP      GETCL
113 052606 105737 003410          1$:    TSTB    P$NNUF        ;SEE IF INCOMPLETE COMMAND TYPED
114 052612 001412
115 052614          PRINTF  $CLINUF
                                MOV      $CLINUF, (SP)
                                MOV      $1, (SP)
                                MOV      SP,RO
                                TRAP    C$PNTF
                                ADD      $4,SP
116 052634 000137 052472          JMP      GETCL
117
118 052640 023727 003252 000067 10$:  CMP      KEYWD1,$SETET   ;WAS "SET E=T" ENTERED ?
119 052646 001711          BEQ      GETCL          ;YES,BRANCH
120 052650 023727 003252 000004          CMP      KEYWD1,$RUN    ;SEE IF RUN WAS TYPED
121 052656 001002          BNE     11$            ; BR IF NO
122 052660 000137 057056          JMP      GTR9           ; START EXEC. IF YES
123 052664 023727 003252 000052 11$:  CMP      KEYWD1,$DMPS   ;IS IT DUMP
124 052672 001004          BNE     14$
125 052674 004737 045212          JSR      PC,DUMPSR     ;GO TO DUMPSR
126 052700 000137 052472          JMP      GETCL          ;AND GO BACK
127 052704 023727 003252 000066 14$:  CMP      KEYWD1,$EXIT   ;IS IT EXIT
128 052712 001005          BNE     40$            ;BRANCH IF NOT
129 052714 012737 177777 017376          MOV      $-1,DCLFLG    ;SET DO CLEAN FLAG
130 052722          EXIT
                                TRAP    C$EXIT
                                .WORD  L10020-.
131 052726 023727 003252 000010 40$:  CMP      KEYWD1,$SETEXP ;SEE IF SET EXPECTED
132 052734 001001          BNE     4$             ; BR IF YES (A SETEXP WAS TYPED)
133 052736 000525          BR      2$
134 052740 023727 003252 000011 4$:  CMP      KEYWD1,$SETTRN ;SEE IF SET TX
135 052746 001407          BEQ     5$             ; BR IF YES
136 052750 105737 003412          TSTB    WRFLG
137 052754 001402          BEQ     77$
138 052756 004737 045776          JSR      PC,DOGLOB     ;DO GLOBAL
139 052762 000137 052472          77$:  JMP      GETCL
140
141 052766 013737 017262 017344 5$:  MOV      TTOTCC,TOTCC
142 052774 023727 017344 001000          CMP      TOTCC,$BUFLIM ;SEE IF BUFFER ALREADY FULL
143 053002 002414          BLT     15$            ; BR IF NOT FULL (BUFLIM $ OF CHARS.)
144 053004          PRINTF  $MSGTRN,$BUFEX ; ELSE TELL OPR. AND DON'T BUILD MSG.
                                MOV      $BUFEX, (SP)
                                MOV      $MSGTRN, (SP)
                                MOV      $2, (SP)
                                MOV      SP,RO
                                TRAP    C$PNTF
                                ADD      $6,SP
145 053030 000137 052472          JMP      GETCL          ; THEN GO GET A NEW COMMAND

```

146	053034	005737	017262		15\$:	TST	TTOTCC		; IF FIRST "SET" THEN GET RID OF DEFAULT
147	053040	001002				BNE	6\$		
148	053042	005037	017260			CLR	TXMTOT		
149	053046	012737	011416	017236	6\$:	MOV	#PTRTAB, TXPTR		; GET POSITION OF END OF TX LIST
150	053054	013701	017260			MOV	TXMTOT, R1		
151	053060	020127	000017			CMP	R1, #MSGLIM		; SEE IF MSG COUNT EXCEEDED.
152	053064	002414				BLT	17\$		; BR IF NO
153	053066					PRINTF	#MSGTRN, #TABEX		; ELSE TELL OPR. AND DON'T BUILD MSG.
	053066	012746	027337					MOV	#TABEX, (SP)
	053072	012746	027415					MOV	#MSGTRN, (SP)
	053076	012746	000002					MOV	#2, -(SP)
	053102	010600						MOV	SP, R0
	053104	104417						TRAP	C\$PNTF
	053106	062706	000006					ADD	#6, SP
154	053112	000137	052472			JMP	GETCL		; THEN GO GET A NEW COMMAND.
155	053116	006301			17\$:	ASL	R1		; # OF MSGS *4 = NEXT FREE PTR BLOCK
156	053120	006301				ASL	R1		
157	053122	060137	017236			ADD	R1, TXPTR		
158	053126	013737	017236	017340		MOV	TXPTR, CPTR		; SETUP CHAR. COUNT, CURRENT ADDR. & PTR
159	053134	013737	017264	017342		MOV	TCURAD, CURADD		
160	053142	004737	045350			JSR	PC, ADDCC		; ADD IN CHAR. COUNT AND CHECK TOTAL
161	053146	004737	045446			JSR	PC, BLDBUF		; GO BUILD MESSAGE IN BUFFER AND PTRS.
162	053152	013737	017340	017236		MOV	CPTR, TXPTR		
163	053160	013737	017344	017262		MOV	TOTCC, TTOTCC		; UPDATE CHAR. COUNT, CURR ADDR. & PTR
164	053166	013737	017342	017264		MOV	CURADD, TCURAD		
165	053174	005237	017260			INC	TXMTOT		
166	053200	005337	003256			DEC	QUALVL		; DEC THE COPY COUNT
167	053204	001270				BNE	5\$		
168	053206	000137	052472			JMP	GETCL		
169									
170	053212	013737	017244	017344	2\$:	MOV	CTOTCC, TOTCC		; SETUP CHAR. COUNT, CURR. ADDR. & PTR
171	053220	023727	017344	001000		CMP	TOTCC, #BUFLIM		; SEE IF BUFFER ALREADY FULL
172	053226	002414				BLT	16\$		; BR IF NOT FULL (BUFLIM # OF CHARS.)
173	053230					PRINTF	#MSGTRN, #BUFEX		; ELSE TELL OPR. AND DON'T BUILD MSG.
	053230	012746	027377					MOV	#BUFEX, (SP)
	053234	012746	027415					MOV	#MSGTRN, (SP)
	053240	012746	000002					MOV	#2, (SP)
	053244	010600						MOV	SP, R0
	053246	104417						TRAP	C\$PNTF
	053250	062706	000006					ADD	#6, SP
174	053254	000137	052472			JMP	GETCL		; THEN GO GET A NEW COMMAND
175	053260	005737	017244		16\$:	TST	CTOTCC		; IF FIRST "SET" THEN GET RID OF DEFAULT
176	053264	001002				BNE	7\$		
177	053266	005037	017242			CLR	CMPTOT		
178	053272				7\$:				
179	053272	012737	011512	017240		MOV	#PTR13, CMPPTR		; INIT COMPARE MESSAGE POINTER
180	053300	013701	017242			MOV	CMPTOT, R1		
181									
182	053304	020127	000017			CMP	R1, #MSGLIM		; SEE IF MSG COUNT EXCEEDED.
183	053310	002414				BLT	18\$		; BR IF NO
184	053312					PRINTF	#MSGTRN, #TABEX		; ELSE TELL OPR. AND DON'T BUILD MSG.
	053312	012746	027337					MOV	#TABEX, (SP)
	053316	012746	027415					MOV	#MSGTRN, (SP)
	053322	012746	000002					MOV	#2, (SP)
	053326	010600						MOV	SP, R0
	053330	104417						TRAP	C\$PNTF
	053332	062706	000006					ADD	#6, SP

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 82 4  
 COMMAND LINE FETCH & INTERPRETATION SECTION

185	053336	000137	052472		JMP	GETCL	; THEN GO GET A NEW COMMAND.
186	053342	006301		184:	ASL	R1	; # OF MSGS *4 = NEXT FREE PTR BLOCK
187	053344	006301			ASL	R1	
188	053346	060137	017240		ADD	R1,CMPPTR	
189	053352	013737	017240	017340	MOV	CMPPTR,CPIR	
190	053360	013737	017246	017342	MOV	CCURAD,CURADD	
191	053366	004737	045350		JSR	PC,ADDCC	; ADD IN XHAR. COUNT AND CHECK TOTAL
192	053372	004737	045446		JSR	PC,BLDBUF	
193	053376	013737	017340	017240	MOV	CPTR,CMPPTR	
194	053404	005237	017242		INC	CMPTOT	
195	053410	013737	017342	017246	MOV	CURADD,CCURAD	; UPDATE CHAR. COUNT, CURR ADDR. & PTR
196	053416	013737	017344	017244	MOV	TOTCC,CTOTCC	
197	053424	005337	003256		DEC	QUALVL	; IF COPY WAS GIVEN, PUT MSG IN BUFF
198	053430	001270			BNE	24	; AGAIN
199	053432	000137	052472		JMP	GETCL	; GO BACK UNTIL GET A 'RUN'
200							
201							
202							
203							
204							

```

1
2
3 .SBTTL ACTION TABLE AND ROUTINES
4 ; USER MUST CLEAR/SET PIGDBD IF USE CLIBIF IN CONNECTION WITH ACTION
5 ; R2 WILL HOLD ACTION CODE FROM PARSING (CLI) NODE
6 CLIACT:
7 ASL R2 ;MULTIPLY ACTION CODE BY 2
8 MOV 10*(R2),R2 ;OFFSET VALUE
9 ADD @10*,R2 ;ADD BASE VALUE
10 JSR PC,(R2) ;GO DO ACTION
11 RTS PC ;RETURN TO TRVACT:
12
13 10*: .WORD ACTNUL-10* ;BRIEF DESCRIPTION OF ACTIONS TAKEN
14 .WORD ACTCLR-10* ;NULL
15 .WORD ACTSHO-10* ;CLEAR
16 .WORD ACTCHK-10* ;SHOW
17 .WORD ACTRUN-10* ;CHECK
18 .WORD ACTHLP-10* ;RUN
19 .WORD ACTCSE-10* ;HELP
20 .WORD ACTCST-10* ;CLEAR OR SHOW EXPECTED
21 .WORD ACTSTE-10* ;CLEAR OR SHOW TRANSMIT
22 .WORD ACTSTT-10* ;SET EXPECTED
23 .WORD ACTSIZE-10* ;SET TRANSMIT
24 .WORD ACTCOP-10* ;SIZE
25 .WORD ACTNUM-10* ;COPY
26 .WORD ACTOPM-10* ;NUMERIC VALUE FOR SIZE OR COPY
27 .WORD ACTSTS-10* ;QUOTED MESSAGE FROM USER
28 .WORD ACTEQO-10* ;STATUS
29 .WORD ACTMSO-10* ;END OF QUOTED MESSAGE FROM USER
30 .WORD ACTMS1-10* ;ONES DATA
31 .WORD ACTMS2-10* ;ZEROS DATA
32 .WORD ACTMS3-10* ;IALT
33 .WORD ACTMS4-10* ;OACT
34 .WORD ACTMS5-10* ;ITEP
35 .WORD ACTMS6-10* ;CCITT
36 .WORD ACTATV-10* ;ALPHA
37 .WORD ACTPAS-10* ;ACTIVE MODE
38 .WORD ACTREC-10* ;PASSIVE MODE
39 .WORD ACTLIS-10* ;RECEIVE MODE
40 .WORD ACTDLL-10* ;LISTEN MODE
41 .WORD ACTTRA-10* ;DOWNLINE LOAD
42 .WORD ACTTAL-10* ;TRANSMIT MODE
43 .WORD ACTNO-10* ;TALK MODE
44 .WORD ACTECH-10* ;/NO
45 .WORD ACTCRC-10* ;ECHO
46 .WORD ACTPRO-10* ;SET CRC BIT
47 .WORD ACTRPS-10* ;SET PROTOCOL BIT
48 .WORD ACTMOP-10* ;STATUS
49 .WORD ACTTLP-10* ;REMOTE STATION IN MAINTENACE LOOP MODE
50 .WORD ACTCLP-10* ;INTERNAL TTL
51 .WORD ACTLLP-10* ;CABLE LOOP
52 .WORD ACTRLP-10* ;LOCAL MODEM LOOP
53 .WORD ACTNUF-10* ;REMOTE MODEM LOOP
54 .WORD ACTBCR-10* ;MORE COMMAND LINE NEEDED
55 .WORD ACTDMS-10* ;BAD CHARACTER IN OPERATOR MESSAGE
56 .WORD ACTDME-10* ;DUMP MEMORY START ADDRESS
57 .WORD ACTDMQ-10* ;DUMP MEMORY END ADDRESS
;DUMP WORD

```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 83 1  
 ACTION TABLE AND ROUTINES

58	053606	000264	.WORD	ACTPRT-10#	;PRINT
59	053610	001610	.WORD	ACTMOS 10#	;MODEM STATUS CHANGE
60	053612	002474	.WORD	ACTSLS 10#	;SHOW TRIB LIST
61	053614	001776	.WORD	ACTETB 10#	;ESTABLISH TRIB
62	053616	002006	.WORD	ACTKTB-10#	;KILL TRIB
63	053620	002016	.WORD	ACTKAL-10#	;KILL ALL
64	053622	002730	.WORD	ACTEKT 10#	;FLAG TRIB KILLED
65	053624	003334	.WORD	ACTCKT-10#	;CHECK VALID TRIB
66	053626	002100	.WORD	ACTEWS 10#	;POLL PARAMETERS
67	053630	000254	.WORD	ACTEXT-10#	;EXIT
68	053632	001310	.WORD	ACTSEX 10#	;SET E-T COMMAND REV B EC
69					



1												
2	053634	112737	177777	003410	ACTNUF: MOV	# 1,P#NNUF						
3	053642	000207			ACTNUL: RTS	PC						
4												
5	053644	012737	000001	003252	ACTCLR: MOV	#CLEAR,KEYWD1						
6	053652	000207			RTS	PC						
7												
8	053654	012737	000002	003252	ACTSHO: MOV	#SHOW,KEYWD1						
9	053662	000207			RTS	PC						
10												
11	053664	012702	003260		ACTHLP: MOV	#HLP,KEYWD1						
12	053670				1\$: PRINTF	#HLPF,(R2),						
	053670	012246										MOV (R2).. (SP)
	053672	012746	024116									MOV #HLPF, (SP)
	053676	012746	000002									MOV #2, (SP)
	053702	010600										MOV SP,R0
	053704	104417										TRAP C#PNTF
	053706	062706	000006									ADD #6,SP
13	053712	020227	003304		CMP	R2,#HLPEND						
14	053716	001364			BNE	1\$						
15	053720	012737	000005	003252	MOV	#HLP,KEYWD1						
16	053726	000207			RTS	PC						
17	053730	012737	000066	003252	ACTEXT: MOV	#EXIT,KEYWD1						
18	053736	000207			RTS	PC						
19	053740	012737	000055	003252	ACTPRT: MOV	#PRT,KEYWD1						
20	053746	004737	042554		JSR	PC,REPORT						
21	053752	000207			RTS	PC						
22												
23	053754	012737	000004	003252	ACTRUN: MOV	#RUN,KEYWD1						
24	053762	112737	177777	003410	MOV	#-1,P#NNUF						
25	053770	012737	000001	017412	MOV	#1,RPASS						
26	053776	000207			RTS	PC						
27												
28	054000	012737	011512	017240	ACTCSE: MOV	#PTR13,CMPPTR						
29	054006	013701	017240		MOV	CMPPTR,R1						
30												
31	054012	013702	017242		MOV	CMPTR,R2						
32	054016	105037	003410		CLRB	P#NNUF						
33	054022	023727	003252	000002	CMP	KEYWD1,#SHOW						
34	054030	001471			BEQ	ACTSHW						
35	054032	012737	000001	017242	MOV	#1,CMPTR						
36	054040	005037	017244		CLR	CMPTR						
37												
38	054044	012737	011512	017240	MOV	#PTR13,CMPPTR						
39	054052	013737	017240	017340	MOV	CMPPTR,CPTR						
40	054060	012701	004416		MOV	#CMPBUF,R1						
41	054064	010137	017246		MOV	R1,CCURAD						
42	054070	000431			BR	ACTCLB						
43												
44	054072	012701	011416		ACTCST: MOV	#PTRTAB,R1						
45	054076	013702	017260		MOV	TXMTOT,R2						
46	054102	105037	003410		CLRB	P#NNUF						
47	054106	023727	003252	000002	CMP	KEYWD1,#SHOW						
48	054114	001437			BEQ	ACTSHW						
49	054116	012737	000001	017260	MOV	#1,TXMTOT						
50	054124	005037	017262		CLR	TXMTOT						
51	054130	012737	011416	017236	MOV	#PTRTAB,IXPTR						

```

52 054136 013737 017236 017340      MOV      TXPTR,CPTR
53 054144 012701 003416              MOV      @TXBUF,R1
54 054150 010137 017264              MOV      R1,TCURAD
55
56 054154 012702 001000      ACTCLB: MOV      @BUFLIM,R2
57 054160 010137 017342              MOV      R1,CURADD      ;SET UP TO PUT DEFAULT MSG IN LIST AFTER 033'S
58 054164 012737 000005 017332      MOV      #5,MSGTYP
59 054172 013737 002162 017334      MOV      MSG5C,CURCC
60 054200 105021      1$: CLRB      (R1),      ;FILL EXPT OR TRAN BUFFER WITH 0'S IF A CLEAR
61 054202 005302              DEC      R2              ;DO "BUFLIM" NUMBER OF BYTE LOCATIONS
62 054204 001375              BNE      1$
63 054206 004737 045446      JSR      PC,BLDBUF      ;"CLEAR" REALLY MEANS TO PUT DEFAULT MSG IN
64 054212 000207              RTS      PC              ;WHEN DONE, RETURN TO PARSER
65
66
67 054214 012705 003334      ACTSHW: MOV      @SHTAB,R5
68 054220 122571 000000      5$:  CMPB      (R5),@(R1)      ;LOOK AT FIRST BYTE OF MSG TO DECIPHER TYPE
69 054224 001404              BEQ      6$
70 054226 020527 003343      CMP      R5,@SHTEND      ;SEE IF LOOKED AT ALL OF DEFAULTS YET
71 054232 001372              BNE      5$
72 054234 005205              INC      R5              ;MUST BE OPR. SPEC'D THEN
73 054236 162705 003335      6$:  SUB      @SHTAB+1,R5
74 054242 006305              ASL      R5
75 054244 016137 000002 017350      MOV      2(R1),TEMP
76 054252              PRINTF  @SHMSG,SHMTYB(R5),TEMP ;PRINT MSG SIZE & TYPE
      MOV      TEMP,(SP)
      MOV      SHMTYB(R5),(SP)
      MOV      @SHMSG,-(SP)
      MOV      #3,-(SP)
      MOV      SP,R0
      TRAP   C$PNTF
      ADD    @10,SP
054252 013746 017350
054256 016546 003314
054262 012746 025613
054266 012746 000003
054272 010600
054274 104417
054276 062706 000010
77 054302 062701 000004      ADD      #4,R1      ;BUMP R1 TO NEXT SET OF POINTERS
78 054306 005302      DEC      R2
79 054310 001341      BNE      ACTSHW
80 054312 013737 017402 020652      MOV      MODTYP,DEV1
81 054320 013737 017404 020654      MOV      MLTYP,DEV2
82 054326 013737 017412 020656      MOV      RPASS,DEV3
83 054334 013737 017410 020660      MOV      PARAM,DEV4
84 054342 004737 047344      JSR      PC,SHWOP      ;SHOW THE OPERATOR THE CURRENT MODE..... ALSO
85 054346 000207      RTS      PC
86
87 054350 013737 003404 017312      ACTDMS: MOV      P$NUM,STADD      ;SETUP STARTING ADDRESS FOR DUMP
88 054356 005037 017316      CLR      BYTBIT      ;SET DEFAULT OF WORD DUMP
89 054362 012737 000052 003252      MOV      @DMP5,KEYWD1      ;FLAG THAT A DUMP WAS TYPED
90 054370 000403      BR
91
92 054372 012737 177777 017316      ACTDMQ: MOV      #1,BYTBIT      ;SET DUMP FLAG TO 'DUMP WORD
93 054400 013737 003404 017314      ACTDME: MOV      P$NUM,ENADD      ;SETUP END ADDRESS FOR DUMP (+START IF NO 'EEE
94 054406 105037 003410      ACTDMX: CLRB      P$NUF      ;CLEAR NOT ENOUGH FLAG, DUMP N/N/B' IS VALID
95 054412 000207      RTS      PC
96

```

```

1
2
3 054414 012737 000010 003252 ACTSTE: MOV    #SETEXP,KEYWD1
4 054422 000403                BR      ACTSTX
5
6 054424 012737 000011 003252 ACTSTT: MOV    #SETTRN,KEYWD1
7 054432 012737 000001 003256 ACTSTX: MOV    #1,QUALVL      ;SET UP DEFAULT COPY TO 1 (/COPY=0)
8 054440 000207                RTS     PC
9
10 054442 012737 000012 003254 ACTSIZE: MOV   #SIZE,QUALFG
11 054450 000207                RTS     PC
12
13 054452 012737 000013 003254 ACTCOP: MOV   #QCOPY,QUALFG
14 054460 000207                RTS     PC
15
16 054462 023727 003254 000012 ACTNUM: CMP   QUALFG,#SIZE      ;SEE IF A SIZE OR COPY TYPED
17 054470 001023                BNE    1$                ;BR IF IT WAS A COPY
18 054472 005737 003404                TST   P#NUM              ;CHECK TO BE SURE DIDN'T TRY SIZE=0
19 054476 001014                BNE    3$                ; BR IF NO
20 054500                PRINTF #CLISEO
    054500 012746 023651                MOV   #CLISEO,(SP)
    054504 012746 000001                MOV   #1,(SP)
    054510 010600                MOV   SP,RO
    054512 104417                TRAP  C$PNTF
    054514 062706 000004                ADD   #4,SP
21 054520 112737 177777 003411                MOV   B 1,P#GDBD      ;SET ERROR IN-CMD FLAG
22 054526 000411                BR    2$
23 054530 013737 003404 017334 3$:  MOV   P#NUM,CURCC      ;IF A SIZE LOAD CURCC WITH BYTE COUNT
24 054536 000405                BR    2$
25 054540 013737 003404 003256 1$:  MOV   P#NUM,QUALVL     ;IF A COPY, LOAD COPY COUNT
26 054546 005237 003256                INC   QUALVL           ;INCREMENT SO FIRST DEC MAKES IT REAL #
27 054552 000522                2$:  BR    ACTMEX
28
29 054554 012737 000007 017332 ACTUPM: MOV   #7,MSGTYP
30 054562 010437 017350                MOV   R4,TEMP
31 054566 005237 017350                INC   TEMP
32 054572 000207                RTS     PC
33
34 054574 010402                ACTEQO: MOV   R4,R2
35 054576 163702 017350                SUB   TEMP,R2
36 054602 010237 017334                MOV   R2,CURCC        ;CALC BYTE COUNT FOR QUOTED TEXT*
37 054606 010237 002166                MOV   R2,OPCNT
38 054612 013701 017350                MOV   TEMP,R1
39 054616 012705 002524                MOV   #OPBUF,R5
40 054622 112125                1$:  MOV   (R1),.(R5)      ;COPY QUOTED TEXT TO OPBUF
41 054624 005302                DEC   R2
42 054626 001375                BNE   1$
43 054630 000473                BR    ACTMEX
44
45 054632                ACTBCR: PRINTF #CLIBCR      ;BAD CHAR. IN OPR. QUOTED STRING
    054632 012746 023604                MOV   #CLIBCR,(SP)
    054636 012746 000001                MOV   #1,(SP)
    054642 010600                MOV   SP,RO
    054644 104417                TRAP  C$PNTF
    054646 062706 000004                ADD   #4,SP
46 054652 000207                RTS     PC
47

```

48	054654	012737	017332		ACTMS0: CLR	MSGTYP	
49	054660	000435			BR	ACTME1	
50	054662	012737	000001	017332	ACTMS1: MOV	#1,MSGTYP	
51	054670	000431			BR	ACTME1	
52	054672	012737	000002	017332	ACTMS2: MOV	#2,MSGTYP	
53	054700	000425			BR	ACTME1	
54	054702	012737	000003	017332	ACTMS3: MOV	#3,MSGTYP	
55	054710	000421			BR	ACTME1	
56	054712	012737	000004	017332	ACTMS4: MOV	#4,MSGTYP	
57	054720	000415			BR	ACTME1	
58	054722	012737	000005	017332	ACTMS5: MOV	#5,MSGTYP	
59	054730	013737	002162	017334	MOV	MSG5C,CURCC	;SETUP DEFAULT SIZE FOR THIS TYPE
60	054736	000430			BR	ACTMEX	
61	054740	012737	000006	017332	ACTMS6: MOV	#6,MSGTYP	
62	054746	013737	002164	017334	MOV	MSG6C,CURCC	;SETUP DEFAULT SIZE FOR THIS TYPE
63							
64	054754	012737	000100	017334	ACTME1: MOV	#64,CURCC	;SETUP DEFAULT SIZE FOR MSGO 4
65	054762	000416			BR	ACTMEX	;EXIT
66							
67							
68							
69	054764	022737	000010	003252	ACTSEX: ;REV B BY EC CMP	#SETEXP,KEYWD1	;DID WE GET HERE FROM "SET E =" COMMAND?
70	054772	001404			BEQ	10\$	;YES,BRANCH
71	054774	112737	177777	003411	MOVB	#1,P;GDBD	;SET ERROR FLAG
72	055002	000406			BR	ACTMEX	;GO TO EXIT
73	055004	004737	045572		10\$:	JSR	PC,FACSIMILE
74	055010	012737	000067	003252	MOV	#SETET,KEYWD1	;GO COPY TRANMIT BUFFER TO EXPECT BUFFER
75	055016	000400			BR	ACTMEX	;SET FLAG TO BE USED IN T1::
76							;GO TO EXIT
77							
78							
79	055020	105037	003410		ACTMEX: CLR	B;NNUF	;CLEAR NOT ENOUGH FLAG
80	055024	000207			RTS	PC	
81							

1	055026	012737	000003	017402	ACTATV: MOV	@ACT,MODTYP	
2	055034	000432			BR	ACTM2X	
3							
4	055036	012737	000002	017402	ACTPAS: MOV	@PAS,MODTYP	
5	055044	105037	003410		CLRB	P#NNUF	;CLEAR NOT ENOUGH FLAG
6	055050	005037	017404		CLR	MLTYP	;CLEAR MAINT LOOP TYPE
7	055054	000207			RTS	PC	
8							
9	055056	005037	017402		ACTREC: CLR	MODTYP	
10	055062	000417			BR	ACTM2X	
11							
12	055064	012737	000006	017402	ACTLIS: MOV	@LIS,MODTYP	
13	055072	000413			BR	ACTM2X	
14							
15	055074	012737	000004	017402	ACTDLL: MOV	@DOW,MODTYP	
16	055102	000407			BR	ACTM2X	
17							
18	055104	012737	000001	017402	ACTTRA: MOV	@TRA,MODTYP	
19	055112	000403			BR	ACTM2X	
20							
21	055114	012737	000005	017402	ACTTAL: MOV	@TAL,MODTYP	
22							
23	055122	042737	000004	017410	ACTM2X: BIC	@ECHOB,PARAM	;DISABLE /ECHO (ALL BUT PASSIVE MODE)
24	055130	105037	003410		CLRB	P#NNUF	;CLEAR NOT ENOUGH FLAG
25	055134	005037	017404		CLR	MLTYP	;CLEAR MAINT LOOP TYPE
26	055140	000207			RTS	PC	
27							

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22-Mar 84 16:24 Page 87  
 ACTION TABLE AND ROUTINES

1	055142	012737	000036	003254	ACTNO:	MOV	#NO,QUALFG		
2	055150	000207				RTS	PC		
3									
4	055152	022737	000036	003254	ACTECH:	CMP	#NO,QUALFG		
5	055160	001422				BEQ	1\$		
6	055162	052737	000004	017410		BIS	#ECHOB,PARAM		
7	055170	022737	000002	017402		CMP	#PAS,MODTYP		;BE SURE IN PASSIVE MODE IF
8	055176	001416				BEQ	2\$		;IF TRYING TO SET /ECHO
9	055200					PRINTF	#CLINPS		
	055200	012746	023541						MOV #CLINPS,(SP)
	055204	012746	000001						MOV #1,(SP)
	055210	010600							MOV SP,RO
	055212	104417							TRAP C\$FNTF
	055214	062706	000004						ADD #4,SP
10	055220	012737	177777	003411		MOVB	#-1,P\$GDBD		
11	055226	042737	000004	017410	1\$:	BIC	#ECHOB,PARAM		
12	055234	005037	003254		2\$:	CLR	QUALFG		;CLEAR "NO" OUT OF QUALIFIER FLAG
13	055240	000501				BR	ACTLXX		
14									
15	055242	012701	000002		ACTCHK:	MOV	#DATCKB,R1		;SET DATA CHECK BIT
16	055246	007413				BR	ACTQFG		
17									
18	055250	012701	000001		ACTSTS:	MOV	#STATB,R1		;SET THE STATUS BIT
19	055254	000410				BR	ACTQFG		
20									
21	055256	012701	000020		ACTCRC:	MOV	#CRCB,R1		;SET THE CRC BIT
22	055262	000405				BR	ACTQFG		
23									
24	055264	012701	000010		ACTMOS:	MOV	#MOCHK,R1		;SET THE MODEM BIT
25	055270	000402				BR	ACTQFG		
26									
27	055272	012701	000040		ACTPRO:	MOV	#PROTOB,R1		;SET THE PROTOCOL BIT
28									
29	055276	050137	017410		ACTQFG:	BIC	R1,PARAM		
30	055302	022737	000036	003254		CMP	#NO,QUALFG		
31	055310	001002				BNE	1\$		
32	055312	040137	017410			BIC	R1,PARAM		
33	055316	005037	003254		1\$:	CLR	QUALFG		;CLEAR "NO" OUT OF QUALIFIER FLAG
34	055322	000450				BR	ACTLXX		
35									
36	055324	013737	003404	017412	ACTRPS:	MOV	P\$NUM,RPASS		;GET NUMBER OF "RUN PASSES"
37	055332	000444				BR	ACTLXX		
38									
39	055334	012737	000005	017404	ACTMOP:	MOV	#5,MLTYP		
40	055342	000417				BR	ACTLPX		
41	055344	012737	000001	017404	ACTTLP:	MOV	#1,MLTYP		
42	055352	000413				BR	ACTLPX		
43	055354	012737	000002	017404	ACTCLP:	MOV	#2,MLTYP		
44	055362	000407				BR	ACTLPX		
45	055364	012737	000003	017404	ACTLLP:	MOV	#3,MLTYP		
46	055372	000403				BR	ACTLPX		
47	055374	012737	000004	017404	ACTRLP:	MOV	#4,MLTYP		
48									
49	055402	022737	000003	017402	ACTLPX:	CMP	#ACT,MODTYP		;BE SURE IN ACTIVE IF TRYING TO SET LOOP
50	055410	001415				BEQ	ACTLXX		; BR IF IN ACTIVE
51	055412	112737	177777	003411		MOVB	#1,P\$GDBD		
52	055420	005037	017404			CLR	MLTYP		;CLEAR ANY LOOP TYPE THAT MAY HAVE GOT SET

J13

SEQ 165

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22-Mar-84 16:24 Page 87 1  
ACTION TABLE AND ROUTINES

53	055424			PRINTF	@CLIBDL				
	055424	012746	023477					MOV	@CLIBDL, (SP)
	055430	012746	000001					MOV	@1, (SP)
	055434	010600						MOV	SP,RO
	055436	104417						TRAP	C\$PNTF
	055440	062706	000004					ADD	@4,SP
54	055444	105037	003410	ACTLXX:	CLRB	P\$NNUF			
55	055450	000207			RTS	PC			
56									

;CLEAR NOT ENOUGH FLAG

```

1 055452 012737 000060 003252 ACTETB: MOV    #ETRB,KEYWD1 ; RECORD THAT ESTABLISH TYPED
2 055460 000207                RTS      PC      ;RETURN TO CALL
3
4 055462 012737 000061 003252 ACTKTB: MOV    #KTRB,KEYWD1 ; RECORD THAT KILLTRIB TYPED
5 055470 000207                RTS      PC      ;RETURN TO CALL
6
7 055472 105037 003410                ACTKAL: CLRB   P$NNUF      ; CLEAR INCOMPLETE INFO FLAG
8 055476 022737 000061 003252          CMP    #KTRB,KEYWD1 ; BE SURE "ALL" IS AFTER A "KILL"
9 055504 001403                BEQ    11$          ; BR IF YES
10 055506 112737 177777 003411        MOVB   #-1,P$GDBD    ; ELSE ERROR IN CMD
11 055514 105737 003411        11$:  TSTB   P$GDBD    ; SEE IF WAS AN ERROR FROM .KTB
12 055520 001401                BEQ    10$          ; BR IF NO
13 055522 000413                BR     2$          ; ELSE EXIT
14 055524 005037 015754        10$:  CLR    TRBTOT   ; ZERO TOTAL # OF TRIB ADDRESSES
15 055530 012702 015712        MOV    #TRIBLS,R2  ; PT R2 TO TRIB ADDRESS TABLE
16 055534 012705 000020        MOV    #16.,R5    ; SETUP R5 AS COUNTER
17 055540 005022        1$:  CLR    (R2)+    ; CLEAR 32 BYTES OF TABLE
18 055542 005305                DEC    R5
19 055544 001375                BNE    1$
20 055546 004737 047214        JSR    PC,WRDEFP  ;WRITE DEFAULTS TO POLL PARMS
21 055552 000207        2$:  RTS      PC      ;RETURN TO CALL
22
23 055554 010246                ACTEWS: MOV    R2,-(SP) ;SAVE R2,R3,R4 ON THE STACK
24 055556 010346                MOV    R3,-(SP)
25 055560 010446                MOV    R4,-(SP)
26 055562 005737 003414        TST    VALTRB     ;VALID TRIB? REV B EC
27 055566 001517                BEQ    ACTW7B     ;NO,BRANCH REV B EC
28 055570 112737 177777 003412 ACTWS9: MOVB   #-1,WRFLG ;SET WRITE GLOBAL FLAG
29 055576 013746 015760                PRINTF #POLPM,INOW ;PRINT POLL PARAMS FOR TRIB #
                                MOV    INOW,(SP)
                                MOV    #POLPM,(SP)
                                MOV    #2,(SP)
                                MOV    SP,R0
                                TRAP   C$PNTF
                                ADD    #6,SP
30 055622 005037 017354                CLR    TEMP2
31 055626 012737 000020 017350        MOV    #16.,TEMP ;USE 16 BYTES AS MULTIPLIER
32 055634 013737 015762 017232        MOV    INDEX,MPLY ;USE TRIB INDEX [BYTE]
33 055642 004737 046436                JSR    PC,MTPLY  ;ON RETURN TEMP2=START ADDR OF
34                                ;THIS TRIBS POLL PRAMS
35 055646 012702 000027                MOV    #27,R2    ;INIT INDEX OF POLL PARAMS
36 055652 032737 000002 023102        BIT    #TRBB,DEVPAR ;IS THIS TRIB
37 055660 001002                BNE    ACTWS5    ;BRANCH IF NOT A TRIB
38 055662 012702 000034                MOV    #34,R2    ;ONLY 35 IS GOOD FOR TRIBS
39 055666 005202                ACTWS5: INC    R2
40 055670 116205 020762        MOVB   TSSIND(R2),R5 ;R5 = 0 FOR WORD 2 FOR BYTE
41 055674 010204                MOV    R2,R4
42 055676 006304                ASL    R4        ;MAKE R4 WORD INDEX
43 055700 010403                MOV    R4,R3
44 055702 042703 177760        BIC    #+C<17>,R3 ;MAKE R3 POLPARM INDEX
45 055706 063703 017354        ADD    TEMP2,R3
46 055712 016337 016220 017350        MOV    POLLIS(R3),TEMP ;GETS DEFAULT
47 055720 016437 020662 017366        MOV    TSSLST(R4),CONOTM
48 055726 000175 055732                JMP    @ACTWS1(R5) ;GO TO CORRECT ACTION
49 055732 055736                ACTWS1: .WORD  ACTWS2
50 055734 056036                .WORD  ACTWS3
51 055736                ACTWS2: PRINTF  CONOTM,TEMP

```



	055736	013746	017350						MOV	TEMP, -(SP)
	055742	013746	017366						MOV	CONOTM, -(SP)
	055746	012746	000002						MOV	#2, -(SP)
	055752	010600							MOV	SP, R0
	055754	104417							TRAP	C\$PNTF
	055756	062706	000006						ADD	#6, SP
52	055762			GMANID	EQUQ, TEMP, 0, 1, 0, 1, YES					;GET INPUT
	055762	104443							TRAP	C\$GMAN
	055764	000406							BR	10001\$
	055766	017350							.WORD	TEMP
	055770	000032							.WORD	T\$CODE
	055772	025055							.WORD	EQUQ
	055774	177777							.WORD	-1
	055776	000000							.WORD	T\$LOLIM
	056000	177777							.WORD	T\$HILIM
	056002									10001\$:
53	056002	013763	017350	016220	ACTWS7:	MOV	TEMP, POLLIS(R3)			;PUT ANSWER BACK
54	056010	032737	000002	023102		BIT	#TRBB, DEVPAR			;IS THIS TRIB
55	056016	001403				BEQ	ACTW7B			;BRANCH IF TRIB
56										
57	056020	022702	000037		ACTW7A:	CMP	#37, R2			;ALL DONE
58	056024	001320				BNE	ACTWS5			
59	056026	012604			ACTW7B:	MOV	(SP)+, R4			;RESTORE R4, R3, R2
60	056030	012603				MOV	(SP)+, R3			
61	056032	012602				MOV	(SP)+, R2			
62	056034	000207				RTS	PC			;RETURN TO CALLING ROUTINE
63										
64										;GET INPUT FOR LO AND HI BYTES
65										
66	056036				ACTWS3:	PRINTF	CONOTM, <B, TEMP><B, TEMP+1>			
	056036	005046							CLR	(SP)
	056040	153716	017351						BISB	TEMP+1, (SP)
	056044	005046							CLR	(SP)
	056046	153716	017350						BISB	TEMP, (SP)
	056052	013746	017366						MOV	CONOTM, (SP)
	056056	012746	000003						MOV	#3, (SP)
	056062	010600							MOV	SP, R0
	056064	104417							TRAP	C\$PNTF
	056066	062706	000010						ADD	#10, SP
67	056072			GMANID	EQUQ1, TEMP, 0, 377, 0, 377, YES					
	056072	104443							TRAP	C\$GMAN
	056074	000406							BR	10002\$
	056076	017350							.WORD	TEMP
	056100	000032							.WORD	T\$CODE
	056102	025111							.WORD	EQUQ1
	056104	000377							.WORD	377
	056106	000000							.WORD	T\$LOLIM
	056110	000377							.WORD	T\$HILIM
	056112									10002\$:
68	056112	113737	017351	017356	MOV	B	TEMP+1, TEMP3			
69	056120				GMANID	EQUQ2, TEMP3, 0, 377, 0, 377, YES				
	056120	104443							TRAP	C\$GMAN
	056122	000406							BR	10003\$
	056124	017356							.WORD	TEMP3
	056126	000032							.WORD	T\$CODE
	056130	025151							.WORD	EQUQ2
	056132	000377							.WORD	377

Address	OpCode	Operand 1	Operand 2	Operand 3	Instruction	Comment	Register	Label
056134	000000							
056136	000377							
056140								
70 056140	113737	017356	017351		MOVB TEMP3,TEMP+1			
71 056146	000715				BR ACTWS7			
72								
73 056150	105037	003410			ACTSLS: CLR P\$NNUF	; CLEAR THE INCOMPLETE CMD FLAG		
74 056154	012737	000002	003252		MOV \$SHOW,KEYWD1	; SET UP TO LOOK LIKE A SHOW CMD		
75 056162	105737	003411			TSTB P\$GDBD	; SEE IF WAS AN ERROR FROM .KTB		
76 056166	001401				BEQ 10\$	; BR IF NO		
77 056170	000504				BR 5\$	; ELSE EXIT		
78 056172	005037	017300			10\$: CLR LNCNT	; INIT ADDR/LINE COUNTER		
79 056176	005737	015754			TST TRBTOT	; SEE IF LIST EMPTY		
80 056202	001011				BNE 1\$	; BR IF NO		
81 056204					PRINTS \$SHTRE	; PRINT THE TRIB LIST IS EMPTY		
056204	012746	025727					MOV \$SHTRE,(SP)	
056210	012746	000001					MOV \$1,(SP)	
056214	010600						MOV SP,R0	
056216	104416						TRAP C\$PNTS	
056220	062706	000004					ADD \$4,SP	
82 056224	000456				BR 4\$			
83 056226	012702	015712			1\$: MOV \$TFIBLS,R2	; POINT R2 TO THE TRIB ADDR LIST		
84 056232	012705	000040			MOV \$32.,R5	; SETUP R5 AS A COUNTER		
85 056236					PRINTS \$SHTRH	; PRINT TRIB LIST HEADER		
056236	012746	025766					MOV \$SHTRH,-(SP)	
056242	012746	000001					MOV \$1,(SP)	
056246	010600						MOV SP,R0	
056250	104416						TRAP C\$PNTS	
056252	062706	000004					ADD \$4,SP	
86 056256	105712				2\$: TSTB (R2)	; SEE IF A NULL ENTRY		
87 056260	001435				BEQ 3\$	; BR IF YES		
88 056262	111237	017350			MOVB (R2),TEMP			
89 056266					PRINTS \$SHTAP,<B,TEMP>			
056266	005046						CLR -(SP)	
056270	153716	017350					BISB TEMP,(SP)	
056274	012746	026017					MOV \$SHTAP,(SP)	
056300	012746	000002					MOV \$2,(SP)	
056304	010600						MOV SP,R0	
056306	104416						TRAP C\$PNTS	
056310	062706	000006					ADD \$6,SP	
90 056314	005237	017300			INC LNCNT	; INCREMENT PRINT COUNTER		
91 056320	022737	000010	017300		CMP \$8.,LNCNT	; SEE IF TIME FOR A CR YET		
92 056326	001012				BNE 3\$			
93 056330					PRINTS \$CR			
056330	012746	031566					MOV \$CR,(SP)	
056334	012746	000001					MOV \$1,(SP)	
056340	010600						MOV SP,R0	
056342	104416						TRAP C\$PNTS	
056344	062706	000004					ADD \$4,SP	
94 056350	005037	017300			3\$: CLR LNCNT			
95 056354	005202				INC R2	; INCREMENT TABLE ADDRESS		
96 056356	005305				DEC R5	; SEE IF CHECKED ALL OF LIST		
97 056360	001336				BNE 2\$	; BR BACK IF NO		
98 056362					4\$: PRINTS \$CR	; ELSE PRINT A PARTING CR		
056362	012746	031566					MOV \$CR,(SP)	
056366	012746	000001					MOV \$1,(SP)	
056372	010600						MOV SP,R0	

Address	Code	Label	Comment	Trap	C\$PNTS
056374	104416			TRAP	
056376	062706	000004		ADD	04,SP
99 056402	000207	5\$: RTS	PC ;RETURN TO CALL		
100					
101 056404		ACTEKT:			
102 056404	005037	003414	CLR VALTRB ;INIT. VALID TRIB FLAG REV B EC		
103 056410	105037	003410	CLRB P\$NNUF ;CLEAR NOT ENOUGH INFO FLAG		
104 056414	105737	003411	TSTB P\$GDBD ; SEE IF WAS AN ERROR FROM ..KTB		
105 056420	001401		BEQ 10\$ ; BR IF NO		
106 056422	000571		BR ACTEXX ; ELSE EXIT		
107 056424	013701	003404	10\$: MOV P\$NUM,R1		
108 056430	005701		TST R1 ; SEE THAT TRIB ADDR NOT 0,377		
109 056432	001403		BEQ 1\$		
110 056434	022701	000377	CMP #377,R1		
111 056440	103012		BHIS 2\$ ;:REV B EC		
112 056442		1\$: PRINTS	#SHTIV,R1		
056442	010146			MOV	R1, -(SP)
056444	012746	026543		MOV	#SHTIV, (SP)
056450	012746	000002		MOV	#2, (SP)
056454	010600			MOV	SP,R0
056456	104416			TRAP	C\$PNTS
056460	062706	000006		ADD	#6,SP
113 056464	000550		BR ACTEXX		
114 056466	022737	000060	003252 2\$: CMP #ETRB,KEYWD1 ; SEE IF KILL OR ESTABLISH		
115 056474	001452		BEQ ACTEKE ; BR IF WAS AN ESTABLISH		
116 056476	012705	000040	MOV #32.,R5 ; ELSE LOOK FOR ADDR TO KILL		
117 056502	012702	015712	MOV #TRIBLS,R2 ; SETUP TABLE PTR AND COUNTER		
118 056506	122201		3\$: CMPB (R2)+,R1 ; LOOK FOR ADDRESS TO KILL		
119 056510	001414		BEQ 4\$ ; BR IF FOUND		
120 056512	005305		DEC R5		
121 056514	001374		BNE 3\$ ; LOOP TIL ALL CHECKED		
122 056516			PRINTF #SHTNF,R1		
056516	010146			MOV	R1, (SP)
056520	012746	026165		MOV	#SHTNF, (SP)
056524	012746	000002		MOV	#2, -(SP)
056530	010600			MOV	SP,R0
056532	104417			TRAP	C\$PNTF
056534	062706	000006		ADD	#6,SP
123 056540	000522		BR ACTEXX		
124 056542	105042		4\$: CLRB -(R2) ;DELETE FOUND TRIB ADDR		
125 056544	005337	015754	DEC TRBTOT ; DECREMENT TOTAL # OF TRIBS		
126 056550	162702	015712	SUB #TRIBLS,R2 ;MOV INDEX TO R2		
127 056554	010237	017232	MOV R2,MPLY		
128 056560	012737	000020	017350 MOV #16.,TEMP		
129 056566	012737	016220	017354 MOV #POLLIS,TEMP2 ;GET THE START ADDR OF THE		
130 056574	004737	046436	JSR PC,MTPLY ;POLL PARMS FOR THIS TRIB TO TEMP2		
131 056600	013705	017354	MOV TEMP2,R5 ;THE PUT IT IN R5		
132 056604	012702	016166	ACTE5B: MOV #POLDEF,R2 ;PUT START ADDR. OF DEFAULT LIST IN R2		
133 056610	012225		ACTE5A: MOV (R2)+,(R5)+ ;MOVE A DEFAULT PARAM TO LIST		
134 056612	022702	016206	CMP #GLBDEF,R2 ;DONE ONE SET ?		
135 056616	001374		BNE ACTE5A ;IF NOT GO BACK AND FINISH		
136 056620	000472		BR ACTEXX ;ALL DONE EXIT.		
137					
138 056622	012737	000040	017350 ACTEKE: MOV #32.,TEMP ; SET UP TO ENTER A TRIB ADDRESS		
139					
140 056630	032737	000003	023100 BIT #3,OPTYP		
141 056636	001403		BEQ 1\$ ;BRANCH IF DMP		

```

142 056640 012737 000014 017350      MOV    #12.,TEMP
143 056646 023737 015754 017350 1$:    CMP    TRBITOT,TEMP      ; SEE IF LIST ALREADY FULL
144 056654 002412                    BLT    2$                ; BR IF NOT FULL YET
145 056656                    PRINTF #SHTFL,R1           ;PRINT ERROR IS LIST FULL
                                MOV    R1,(SP)
                                MOV    #SHTFL,-(SP)
                                MOV    #2,-(SP)
                                MOV    SP,R0
                                TRAP   C$PNTF
                                ADD    #6,SP

146 056700 000442                    BR     ACTEXX
147 056702 012702 015712 2$:    MOV    #TRIBLS,R2      ; NOW CHECK TO SEE ADDR IS UNIQUE
148 056706 013705 017350      MOV    TEMP,R5
149 056712 122201 3$:    CMPB  (R2)+,R1        ; CHECK EACH ADDR AGAINST NEW ONE
150 056714 001423      BEQ   5$                ; BR IF EQUAL
151 056716 005305      DEC   R5
152 056720 001374      BNE   3$                ; LOOP TIL ENTIRE TABLE CHECKED
153
154 056722 012702 015712      MOV    #TRIBLS,R2      ; ONCE CHECKED LIST
155 056726 105722 4$:    TSTB  (R2)+           ; LOOK FOR EMPTY SLOT TO LOAD
156 056730 001376      BNE   4$
157 056732 110142      MOVB  R1,-(R2)        ; LOAD TRIB ADDR IN EMPTY SLOT
158 056734 005237 015754      INC   TRBITOT          ; INC TOTAL # OF TRIB ADDRESSES
159 056740 162702 015712      SUB   #TRIBLS,R2      ;SUBTRACT START OF LIST FROM POINT TO
160
161 056744 012737 177777 003414    MOV    #-1,VALTRB     ;SET VALID TRIB FLAG REV B EC
162 056752 010237 015762      MOV    R2,INDEX
163 056756 010137 015760      MOV    R1,INDW
164 056762 000411      BR    ACTEXX
165 056764 5$:    PRINTF #SHTUN,R1      ; PRINT ADDR NOT UNIQUE ERROR
                                MOV    R1,(SP)
                                MOV    #SHTUN,(SP)
                                MOV    #2,(SP)
                                MOV    SP,R0
                                TRAP   C$PNTF
                                ADD    #6,SP

166
167 057006      ACTEXX:
168 057006 000207      RTS    PC              ;RETURN TO CALL
169
170 057010      ACTCKT:
171 057010 112737 177777 003410    MOVB  #1,P$INUF       ; SET INCOMPLETE INFO FLAG
172 057016 032737 000001 023102    BIT   #MTP,DEVPAR     ; SEE IF IN PT PT OR MULTIPT MODE
173 057024 001013      BNE   1$              ; BR IF IN MULTIPT MODE
174 057026      PRINTF #CLIPPE      ; TRIB CMDS INVALID IN PT PT MODE
                                MOV    #CLIPPE,-(SP)
                                MOV    #1,(SP)
                                MOV    SP,R0
                                TRAP   C$PNTF
                                ADD    #4,SP

175 057046 112737 177777 003411    MOVB  #1,P$GDBD      ; SET THE ERROR IN CMD FLAG
176 057054 000207 1$:    RTS    PC              ;RETURN TO CALL
177
178
179

```

```

1
2
3
4 057056 032737 000002 017410 GTR9: BIT @DATCKB,PARAM ;IS THIS DATA CHECK
5 057064 001421 BEQ 44$ ;BRANCH IF NO
6 057066 005737 017404 TST MLTYP
7 057072 001416 BEQ 44$ ;BRANCH IF NOT LOOP
8 057074 023737 017242 017260 CMP CMPTOT, TXMTOT ;ARE TX AND EX EQUAL
9 057102 001412 BEQ 44$ ;BRANCH IF YES
10 057104 PRINTF @CLIPW
    057104 012746 023750 MOV @CLIPW, (SP)
    057110 012746 000001 MOV @1, (SP)
    057114 010600 MOV SP, R0
    057116 104417 TRAP C$PNTF
    057120 062706 000004 ADD @4, SP
11 057124 000137 052472 JMP GETCL
12 057130 032737 000001 023102 44$: BIT @MTP, DEVPAR ;IS THIS MULTIPOINT
13 057136 001004 BNE 3$ ;BRANCH IF MULTIPOINT
14 057140 112737 000001 015712 MOVB @1, TRIBLS ;MAKE TRIBLS =1
15 057146 000570 BR 2$
16 057150 005737 015754 3$: TST TRBTOT ;IS TRIB TOTAL
17 057154 001013 BNE 4$ ;ZERO?..BR IF NOT
18 057156 PRINTF @SHTLPA ;PRINT ERROR MUST ESTABLISH TRIB
    057156 012746 026316 MOV @SHTLPA, (SP)
    057162 012746 000001 MOV @1, (SP)
    057166 010600 MOV SP, R0
    057170 104417 TRAP C$PNTF
    057172 062706 000004 ADD @4, SP
19
20 057176 112737 177777 003411 MOVB @-1, P$GDBD ;SET ERROR FLAG
21 057204 023727 017404 000001 44$: CMP MLTYP, @TTL ;IS LOOP CABLE OR REMOTE
22 057212 003413 BLE 5$ ;BRANCH IF INT OR NONE
23 057214 PRINTF @SHTLP ;PRINT ERROR LOOP MUST BE INT FOR MTP
    057214 012746 026231 MOV @SHTLP, (SP)
    057220 012746 000001 MOV @1, -(SP)
    057224 010600 MOV SP, R0
    057226 104417 TRAP C$PNTF
    057230 062706 000004 ADD @4, SP
24 057234 112737 177777 003411 MOVB @-1, P$GDBD ;SET ERROR FLAG
25 057242 022737 000001 017404 54$: CMP @TTL, MLTYP ;IS IT INTERNAL
26 057250 001057 BNE 10$ ;IF NOT THEN CHECK COMPARE TOTALS
27 057252 032737 000002 023102 BIT @TRBB, DEVPAR ;IS THIS CONTROL OR TRIB
28 057260 001013 BNE 6$ ;BRANCH IF CONTROL
29 057262 PRINTF @SHTLPB ;PRINT ERROR MUST BE CONTROL
    057262 012746 026371 MOV @SHTLPB, (SP)
    057266 012746 000001 MOV @1, (SP)
    057272 010600 MOV SP, R0
    057274 104417 TRAP C$PNTF
    057276 062706 000004 ADD @4, SP
30 057302 112737 177777 003411 MOVB @-1, P$GDBD ;SET ERROR FLAG
31 057310 022737 000001 015754 64$: CMP @1, TRBTOT ;IS TRIB TOATAL = 1
32 057316 001011 BNE 7$ ;BRANCH IF MORE
33 057320 012737 177777 015762 MOV @-1, INDEX
34 057326 004737 046462 JSR PC, GIVIND ;GET TRIBN WITH ADDRESS
35 057332 022737 000001 015756 CMP @1, TRIBN
36 057340 001423 BEQ 10$ ;OK IF ADD 1
37 057342 PRINTF @SHTLPC

```



CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 89 2  
ACTION TABLE AND ROUTINES

```

80
81 057674 012737 177777 015762 GTRX2: MOV      # 1,INDEX      ;MAKE INDEX = 1
82 057702 013737 017334 017350 GTRX2C: MOV     CURCC,TEMP
83 057710 032737 000001 023102      BIT      #MTP,DEVPAR
84 057716 001404                BEQ      GTRX22      ;IF NOT MULTI GO TO 22
85 057720 032737 000002 017410      BIT      #DATCKB,PARAM ;IS THERE DATA CHECKING
86 057726 001005                BNE      GTRX2A      ;BRANCH IF CHECKING
87 057730 012737 001000 017334 GTRX22: MOV     #BUFLIM,CURCC ;SET UP CHAR COUNT TO 'BUFLIM
88 057736 005037 017350                CLR      TEMP
89 057742 004737 046462 GTRX2A: JSR     PC,GTVIND      ;GET VALID INDEX
90 057746 022737 000040 015762      CMP      #32.,INDEX      ;IS IT 32
91 057754 001423                BEQ      GTRX2B      ;YES.. ALL DONE GO EXECUTE MODE
92
93                                ;GET RXBUF PTR FIGURE
94
95 057756 012737 005416 017354      MOV      #RXBUF,TEMP2     ;TEMP = 0 FOR PTP OR MTP/W NO CHK
96 057764 013737 015762 017232      MOV      INDEX,MPLY      ;INDEX X TEMP. RXBUF ADDR =
97 057772 004737 046436                JSR      PC,MTPLY        ;NEW RXBUF ADDR.
98 057776 013737 017354 017342      MOV      TEMP2,CURADD     ;SET UP RX BUFFER ADDRESS
99
100                                ;GET CURRENT POINTER FIGURE
101
102 060004 004737 046546                JSR      PC,GRPTCP
103
104                                ;GO LOAD '33' TO BUFFER
105
106 060010 012737 000010 017332      MOV      #10,MSGTYP      ;SET UP FOR 33 TO FILL RX BUFFERS
107 060016 004737 045446                JSR      PC,BLDBUF      ;CLEAR RX BUFFER
108 060022 000727                BR       GTRX2C          ;GO BACK FOR MORE
109 060024 013702 017402 GTRX2B: MOV     MODTYP,R2
110 060030 006302                ASL      R2
111 060032 000172 017420                JMP      @MODE(R2)      ;MODE DISPATCH
112

```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 90  
 RECEIVE MODE SECTION

```

1      .SBTTL          RECEIVE MODE SECTION
2      ;**
3      ; FUNCTIONAL DESCRIPTION:
4      ; RECEIVE-ONLY (OR ONE-WAY-IN) ROUTINE
5      ; IN THIS MODE OF TESTING THE DEVICE'S RECEIVER IS ENABLED IN EXPECTATION
6      ; OF RECEIVING A MESSAGE. AFTER RECEIVING AN "EXPECTED" NUMBER OF
7      ; MESSAGES, THE DATA RECEIVED CAN BE COMPARED AGAINST A LIST OF "EXPECT
8      ; TO RECEIVE" MESSAGES IF DATA-CHECKING IS ENABLED.
9      ;
10     ; SUBORDINATE ROUTINES USED:
11     ; "ALLTR"
12     ;
13     ; CALLING SEQUENCE:
14     ; JMP          @MODE(R2)          ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
15     ; --
16
17 060036 052737 000104 017414 RXONLY: BIS    @QRX!ERX,FLAG    ;SET UP RX QUE
18 060044 004737 046234          JSR      PC,LCPRLS    ;LOAD CPTRLS (RX PTRS)
19 060050 004737 046164          JSR      PC,RXQUAL    ;GO QUE ALL VALID RX S
20 060054 005037 017340 RXON3: CLR    CPTR          ;
21 060060 000137 060210          JMP     ALLTR        ;GO RX.
22

```



CZCLMCO DMP/V-11 DCLT  
TRANSMIT MODE SECTION

MACRO V05.00 Thursday 22-Mar 84 16:24 Page 91

```

1          .SBTTL          TRANSMIT MODE SECTION
2
3          ;**
4          ; FUNCTIONAL DESCRIPTION:
5          ;   TRANSMIT-ONLY (OR ONE-WAY-OUT) ROUTINE
6          ;   IN THIS MODE OF TESTING A LIST OF MESSAGES IS TRANSMITTED WITHOUT
7          ;   EXPECTING ANY DATA TO BE RECEIVED.  A REPETITION COUNT CAN BE
8          ;   SPECIFIED TO REPETITIVELY TRANSMIT THE LIST.
9
10         ; SUBORDINATE ROUTINES USED:
11         ;   "ALLTR"
12
13         ; CALLING SEQUENCE:
14         ;   JMP      SMODE(R2)          ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
15         ; -
16
17 060064 042737 000002 017410 TXONLY: BIC      #DATCKB,PARAM      ;SET NOCHECK
18 060072 004737 046364 TXON2:  JSR      PC,LCPTLS      ;LOAD TX POINTERS AND TX COUNTS
19 060076 052737 000210 017414      BIS      #QTX!ETX,FLAG      ;SET THE QUE TX FLAG
20 060104 004737 046340      JSR      PC,CLRPLS      ;CLEAR RXPRT LIST
21 060110 012737 000040 015762      MOV      #32.,INDEX
22 060116 000137 060210      JMP      ALLTR          ;GO TX.
23

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

```
.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      SMODE(R2)          ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
; -

PLCK:  JSR      PC,LCPTLS      ;LOAD TX POINTERS AND TX COUNTS
        JSR      PC,LCPRLS     ;SET UP CPTRR TO REC POINTERS
        BIS      #QRX!ERX,FLAG ;SET UP Q AND EXPECT RX
        JSR      PC,RXQUAL     ;QUE ALL
        JMP      ALLTR        ;AND GO RX FIRST MSG.
```

```
19 060122 004737 046364
20 060126 004737 046234
21 060132 052737 000104 017414
22 060140 004737 046164
23 060144 000137 060210
```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22-Mar-84 16:24 Page 93  
ACTIVE MODE SECTION

```

1      .SBTTL          ACTIVE MODE SECTION
2
3      ;**
4      ; FUNCTIONAL DESCRIPTION:
5      ; ACTIVE MODE SECTION
6      ; IN THIS MODE OF TESTING A LIST OF MESSAGES IS TRANSMITTED AND
7      ; MESSAGES ARE EXPECTED TO BE RECEIVED.  RECEIVED DATA CAN BE COMPARED
8      ; AGAINST "EXPECTED" DATA IF DATA-CHECKING IS ENABLED.
9      ; NOTE: IF BOTH ENDS OF THE LINK ARE IN ACTIVE MODE, THEN THE
10     ; LINK MUST BE A FULL DUPLEX LINK!
11
12     ; SUBORDINATE ROUTINES USED:
13
14     ; "ALLTR"
15
16     ; CALLING SEQUENCE:
17     ;     JMP      @MODE(R2)          ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
18     ; --
19
20 060150 ALCK:
21 060150 032737 000002 017410      BIT      @DATCKB,PARAM      ;IS IT DATA CHECK
22 060156 001003                                BNE      1$              ;BRANCH IF CHECK
23 060160 013737 017260 017276      MOV      TXMTOT,RXMTOT   ;IF NOCH MAKE RX=TX
24 060166 004737 046364      1$:     JSR      PC,LCPILS      ;LOAD TX POINTERS AND COUNTS
25 060172 004737 046234                                JSR      PC,LCPRLS      ;LOAD RX POINTERS
26 060176 052737 000314 017414      BIS      @QRX!QTX!ETX!ERX,FLAG
27 060204 004737 046164                                JSR      PC,RXQUAL      ;QUE UP 1 RX BUFFER FOR ALL VOID
28

```

```

1      .SBTTL          TRANSMIT  RECEIVE FOR ALL STANDARD MODES
2
3      ;**
4      ; FUNCTIONAL DESCRIPTION:
5      ; THIS CODE PERFORMS THE FOLLOWING FUNCTIONS
6      ; 1.) IF RX BUFFERS ARE TO BE QUED, TELL DEVICE
7      ;     CODE TO QUE THEM ,LOG RECEIVE QUED.
8      ; 2.) IF TX BUFFERS ARE TO BE QUED ,TELL DEVICE
9      ;     CODE TO QUE THEM, LOG TRANSMIT QUED.
10     ; 3.) WAIT FOR EITHER RECIVE BUFFER OR TRANSMIT BUFFER OR
11     ;     BOTH TO COMPLETE
12     ; 4.) IF RECEIVE COMPLETE LOG IT UPDATE RX TABLE IF DATA
13     ;     CHECKING.
14     ; 5.) IF TRANSMIT COMPLETE LOG IT.
15     ; 6.) WHEN BOTH TRANSMIT AND RECEIVE LISTS ARE DONE
16     ;     GO TO THE COMPARE BUFFER CODE
17
18     ; SUBORDINATE ROUTINES USED:
19     ; "DVRXQ"  QUE RECEIVE BUFFER SPACE TO DEVICE
20     ; "LOGRXQ"-LOG RECEIVE BUFFER SPACE TO EVENT LOG
21     ; "LOGTXQ"-LOG TRANSMIT BUFFER QUED TO EVENT LOG
22     ; "DVTXRX"-QUE TRANSMIT BUFFER AND WAIT FOR RX
23     ;             OR TX TO COMPLETE
24     ; "LOGRXC" LOG RECEIVE BUFFER COMPLETED TO EVENT LOG
25     ; "LOGTXC" LOG TRANSMIT BUFFER COMPLETED TO EVENT LOG
26
27     ; USE OF FLAG BITS:
28     ; QRX  SET ON INPUT TO ALLTR IF REC IS TO BE QUED TO
29     ;     DEVICE. CLEARED BY DVRXQ AND THEN SET BY DVTXRX
30     ;     WHEN RX BUFFER IS COMPLETED.
31     ; QTX  - SET ON INPUT TO ALLTR IF TRANSMIT IS TO BE QUED TO
32     ;     DEVICE. CLEARED ON ENTRY TO DVTXRX AND SET BY DVTXRX
33     ;     WHEN TX BUFFER IS COMPLETED.
34     ; ETX  - USED BY DVTXRX TO DETERMINE IF TX BUFFER COMPLETED IS
35     ;     EXPECTED.
36     ; ERX  - USED BY DVTXRX TO DETERMINE IF RX BUFFER COMPLETED IS
37     ;     EXPECTED.
38
39     ; CALLING SEQUENCE:
40     ;     JMP  ALLTR          ;GO TO TRANSMIT RECEIVE FOR ALL STANDARD MODES
41     ;
42     ;--
43
44
45 060210 ALLTR:
46 060210 032737 000004 017414 ALCK5: BIT  #QTX,FLAG
47 060216 001406 BEQ  ALCK1          ;IF NOT RX GO TO TX'S
48 060220 004737 046526 ALCK5B: JSR  PC,ULRPLS ;GET RX INDEX
49 060224 004737 047302 JSR  PC,LOGAQR  ;LOG AND QUE REC.
50 060230 00477 046506 JSR  PC,LDRPLS ;RESTORE RX PTR TO LIST
51 060234 032737 000010 017414 ALCK1: BIT  #QTX,FLAG
52 060242 001422 BEQ  ALCK2          ;IF NO TX'S GO TO 2
53 060244 004737 047102 JSR  PC,GNTXPR
54 060250 013702 017340 MOV  CPTR,R2
55 060254 011237 017354 MOV  (R2),TEMP2
56 060260 012237 017250 MOV  (R2)+,DVTXA
57 060264 011237 017356 MOV  (R2),TEMP3
    
```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 94 1  
 TRANSMIT RECEIVE FOR ALL STANDARD MODES

```

58 060270 012237 017252          MOV      (R2)+,DVTCC
59 060274 010237 017340          MOV      R2,CPTR
60 060300 004737 046644          JSR      PC,LDTPLS      ;RELOAD LIST
61 060304 004737 042022          JSR      PC,LOGTXQ
62
63 060310 004737 064070          ALCK2:  JSR      PC,DVTXRX      ;GO TO TX AND RX SUB ROUT.
64
65 060314 032737 000004 017414    BIT      @QRX,FLAG      ;CHECK FOR REC. MSG.
66 060322 001532                BEQ      ALCK3
67 060324 013737 017270 017354    MOV      DVRXA,TEMP2
68 060332 013737 017272 017356    MOV      DVRCC,TEMP3
69 060340 013737 017266 015756    MOV      DVRTB,TRIBN
70 060346 004737 042074          JSR      PC,LOGRXC      ;LOG REC COMPLETE
71 060352 032737 000004 017410    UPTABL: BIT      @ECHO8,PARAM  ;IS THIS ECHO MODE(PASSIVE)
72 060360 001410                BEQ      UPTA4          ;IF NOT GO TO 4
73 060362 004737 046664          JSR      PC,ULTPLS
74 060366 013702 017340          MOV      CPTR,R2      ;ELSE SET R2 TO PRESENT TX TABLE
75 060372 013722 017354          MOV      TEMP2,(R2)+  ;STORE OFF RX ADD
76 060376 013712 017356          MOV      TEMP3,(R2)  ;AND CC
77 060402 032737 000002 017410    UPTA4:  BIT      @DATCKB,PARAM  ;IS DATA CHECKING ASKED FOR
78 060410 001012                BNE      UPTA1        ;IF SO GO TO UPTA1
79 060412 004737 047154          JSR      PC,GETIND     ;GET INDEX
80 060416 004737 046266          JSR      PC,LCPRL1    ;RESTORE POINTER
81 060422 013737 017354 017336    MOV      TEMP2,CPTRR  ;RESTORE POINTER
82 060430 004737 046506          JSR      PC,LDRPLS   ;LOAD COUNT AND LIST
83 060434 000430                BR       UPTEX
84
85 060436 004737 046526          UPTA1:  JSR      PC,ULRPLS   ;GET PTR FROM LIST
86 060442 013702 017336          MOV      CPTRR,R2
87 060446 011237 017350          MOV      (R2),TEMP
88 060452 163737 017356 017350    SUB      TEMP3,TEMP   ;LOAD TEMP WITH PREV. COUNT
89 060460 013722 017356          MOV      TEMP3,(R2)+ ;LOAD TEMP WITH PREV.COUNT CURRENT
90 060464 063737 017356 017354    ADD      TEMP3,TEMP2
91 060472 013722 017354          MOV      TEMP2,(R2)+ ;STORE OF NEW ADD
92 060476 013712 017350          MOV      TEMP,(R2)   ;AND NEW CC
93 060502 162702 000002          SUB      @2,R2       ;PUT POINTER BACK TO ADDR.
94 060506 010237 017336          MOV      R2,CPTRR   ;AND RESTORE IT.
95 060512 004737 046506          JSR      PC,LDRPLS
96 060516                UPTEX:
97 060516 022737 000002 017402    CMP      @PAS,MODTYP
98 060524 001011                BNE      ALCK2A
99 060526 005337 015762          DEC      INDEX      ;IF NOT PASSIVE LOOP THEN GO TO 2A
100 060532 042737 000004 017414    BIC      @QRX,FLAG   ;IF PASSIVE NEXT TXQ WILL BE FOR THIS TRIB
101 060540 052737 000210 017414    BIS      @QTX!ETX,FLAG ;CLEAR BOTH EXPECTED AND COMPLETED FLAGS
102 060546 000632                ER       ALCK1      ;SET THE TX FLAGS
103
104 060550 004737 046624          ALCK2A: JSR      PC,ULRCLS   ;GET COUNT
105 060554 005337 017274          DEC      DVRCT       ;DEC REC COUNT
106 060560 004737 046604          JSR      PC,LDRCLS   ;RESTORE COUNT
107 060564 005737 017274          TST     DVRCT       ;IS IT ALL DONE
108 060570 001007                BNE      ALCK3
109 060572 042737 000004 017414    BIC      @QRX,FLAG   ;NO. GO CHECK TX
110 060600 005037 017336          CLR     CPTRR       ;CLEAR THE RX FLAG
111 060604 004737 046506          JSR      PC,LDRPLS   ;YES. CLEAR POINTER
112 060610 032737 000010 017414    ALCK3:  BIT      @QTX,FLAG  ;AND RELOAD LIST
113 060616 001467                BEQ     ALCK4        ;IS IT TX
114 060620 013737 017250 017354    MOV     DVTXA,TEMP2  ;IF NOT TX THEN GO BACK

```

CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 94 2  
 TRANSMIT - RECEIVE FOR ALL STANDARD MODES

```

115 060626 013737 017252 017356      MOV      DVTCC,TEMP3      ;LOG TX COMPLETED
116 060634 013737 017254 015756      MOV      DVTTB,TRIBN
117 060642 004737 042040              JSR      PC,LOGTXC
118 060646 004737 046724              JSR      PC,ULTCLS      ;GET COUNT TO DVTCT
119 060652 005337 017256              DEC      DVTCT          ;DEC TX COUNT
120 060656 004737 046704              JSR      PC,LDTCLS      ;AND RELOAD LIST
121 060662 022737 000002 017402      CMP      #PAS,MODTYP
122 060670 001020              BNE      ALCK3A          ;IF NOT PASSIVE MODE GO TO 3A
123 060672 042737 000010 017414      BIC      #QTX,FLAG      ;CLEAR THE TX FLAGS
124 060700 005737 017256              TST      DVTCT
125 060704 001403              BEQ      ALCK3D          ;IF NO MORE MMSG TO RX FOR THIS TRIB
126                                     ;EXIT WITHOUT RESETTING QRX
127 060706 052737 000104 017414      BIS      #QRX+ERX,FLAG  ;AND SET THE RX FLAGS
128 060714 004737 047016      ALCK3D: JSR      PC,GATCFL
129 060720 005737 017256              TST      DVTCT
130 060724 001007              BNE      ALCK3C          ;IF MORE TX'S TO IT
131 060726 000137 061022              JMP      CMPSR           ; ELSE COMPARE
132 060732 004737 047016      ALCK3A: JSR      PC,GATCFL  ;GET ALL TX COUNTS FROM LIST
133 060736 005737 017256              TST      DVTCT          ;IS IT ALL DONE
134 060742 001404              BEQ      ALCK3B          ;IF NOT GO BACK TO 5
135 060744 004737 047154      ALCK3C: JSR      PC,GETIND
136 060750 000137 060210              JMP      ALCK5
137 060754 005037 017340      ALCK3B: CLR      CPTR          ;CLEAR POINTER
138 060760 042737 000010 017414      BIC      #QTX,FLAG      ;CLEAR TX FLAG
139 060766 032737 000002 017410      BIT      #DATCKB,PARAM  ;IS IT DAT CHECK
140 060774 001405              BEQ      ALCK4A          ;IF NOT THEN END WO CKING RX.
141 060776 004737 046744      ALCK4: JSR      PC,GARPFL
142 061002 005737 017336              TST      CPTRR
143 061006 001356              BNE      ALCK3C          ;IF SOME RX'S LEFT GO BACK
144 061010 005737 017340      ALCK4A: TST      CPTR
145 061014 001402              BEQ      CMPSR          ;BRANCH IF ANY TX S LEFT
146 061016 000137 060310              JMP      ALCK2
147
148
149
150

```

```

1          .SBTTL          DATA COMPARISON CODE
2
3
4          ;**
5          ; FUNCTIONAL DESCRIPTION:
6          ;
7          ;     CMPSR   COMPARE CODE
8          ;     THIS CODE COMPARES THE RECEIVED DATA AGAINST THE
9          ;     EXPECTED AND FILLS THE EVENT LOG WITH 1 OF 3 MSGS.
10         ;
11         ;     NOTE: IF NO DATA CHECKING SKIP THIS CODE
12         ;
13         ;     1.) A DATA COMPARISON ENTRY WHICH REPORTS THE NUMBER
14         ;     OF COMPARISON ERRORS FOUND
15         ;     2.) A DATA COMPARISON ENTRY WHICH REPORTS DIFFERENCES
16         ;     IN REC LENGTH TO COMPARE LENGTH.
17         ;     3.) A DATA COMPARISON STARTED ENTRY WHICH REPORTS ADDRESS
18         ;     OF RECEIVE BUFFER AND BYTE COUNT.
19         ;     THIS CODE ALSO REPORTS SOFT ERRORS FOR DATA COMPARISON
20         ;     (THE FIRST 5 ONLY), LENGTH ERROR, AND TOTAL NUMBER OF ERRORS
21         ;
22         ;
23         ; SUBORDINATE ROUTINES USED:
24         ;
25         ;     "LOGCMP" - SEE ITEM 3 ABOVE
26         ;     "LOGCML" - SEE ITEM 2 ABOVE
27         ;     "LOGCMD"  SEE ITEM 1 ABOVE
28         ;
29         ; CALLING SEQUENCE:
30         ;     JMP      CMPSR          ; JUMP TO DATA COMPARISON CODE
31         ; --
32
33 061022 032737 000002 017410 CMPSR: BIT      #DATCKB,PARAM ; IS DATA CHECKING TO BE DONE
34 061030 001534          BEQ      CMPSEX          ; IF NOT THEN EXIT
35 061032 012737 177777 015762      MOV      # 1,INDEX
36 061040 004737 046462          CMPNEW: JSR      PC,GTVIND
37 061044 022737 000040 015762      CMP      #32,INDEX
38 061052 001523          BEQ      CMPSEX          ; END IF NO MORE TRIBS
39
40 061054 004737 046546          JSR      PC,GRPTCP
41 061060 013737 017240 017336      MOV      CMPPTR,CPTRR ; AND START OF COMPARE POINTS TO CPTRR
42 061066 013737 017276 017274      MOV      RXMTOT,DVRCT
43
44 061074          CMPS3:
45 061074 013702 017340          MOV      CPTR,R2          ; MOVE CURRET RX PT. TO R2
46 061100 011237 017354          MOV      (R2),TEMP2      ; MOVE RX ADD TO EVENT LOG
47 061104 012201          MOV      (R2)+,R1        ; SET R1 TO START ADD OF RX
48 061106 012237 017356          MOV      (R2)+,TEMP3     ; SET CHAR COUNT TO EVENT LOG
49 061112 010237 017340          MOV      R2,CPTR        ; RESTORE RX POINT
50
51 061116 013702 017336          MOV      CPTRR,R2       ; PUT R2 AT COMPARE TABLE
52 061122 012203          MOV      (R2)+,R3       ; SET R3 TO COMPARE ADD
53 061124 012204          MOV      (R2)+,R4       ; SET R4 TO COMP CC
54 061126 010237 017336          MOV      R2,CPTRR      ; RESTORE POINTER
55 061132 010437 017360          MOV      R4,TEMP4
56 061136 004737 042170          JSR      PC,LOGCMP      ; LOG COMPARE START.
57

```

```

58 061142 020437 017356      CMP      R4,TEMP3      ;IS COMPARE COUNT = TO RX COUNT
59 061146 001410      BEQ      CMPS7          ;IF SO GO TO 7
60 061150 005237 017310      INC      ERRCNT
61 061154      ERRSOFT 1,EDDLE,ERR10 ;PRINT ERROR
    061154 104457
    061156 000001          TRAP      C$ERSOFT
    061160 030222          .WORD    1
    061162 041430          .WORD    EDDLE
    061164 004737 042206      JSR      PC,LOGCML     ;LOG LENGTH ERROR
    061164 004737 042206      JSR      PC,LOGCML     ;LOG LENGTH ERROR
63
64 061170 005037 017360      CMPS7:  CLR      TEMP4      ;CLEAR BAD BYTE COUNTER
65 061174 012737 000001 017346  MOV      #1,OFSET      ;SET OFFSET BYTE COUNT TO 1
66 061202 122123      CMPS1:  CMPB     (R1)+,(R3)+ ;COMPARE RX WITH EXPETED
67 061204 001422      BEQ      CMPS6          ;IF EQUAL THEN GO TO 6
68
69 061206 005237 017360      CMPS2:  INC      TEMP4      ;INC BAD COUNT
70 061212 023727 017360 000005  CMP      TEMP4,#5      ;IS IT MORE THEN 5
71 061220 101014      BHI     CMPS6          ;IF SO GO FOR MORE
72 061222 114337 017370      MOVB   -(R3),GOOD     ;STORE GOOD BYTE FOR ERROR
73 061226 114137 017371      MOVB   -(R1),BAD      ;STORE BAD BYTE FOR ERROR
74 061232 005237 017310      INC      ERRCNT
75 061236      ERRSOFT 2,EDDDE,ERR1 ;REPORT COMPARISON FAILURE TO OPR.
    061236 104457          TRAP      C$ERSOFT
    061240 000002          .WORD    2
    061242 030257          .WORD    EDDDE
    061244 041340          .WORD    ERR1
76 061246 005201      INC      R1
77 061250 005203      INC      R3
78 061252 005237 017346      CMPS6:  INC      OFSET      ;INC OFFSET
79 061256 005304      DEC      R4            ;ELSE DEC CHAR COUNT AND SEE IF 0
80 061260 001350      BNE     CMPS1          ;IF NOT GO BACK
81 061262 005737 017360      TST     TEMP4          ;SEE IF ANY CMP ERRS FOR THIS MSG
82 061266 001410      BEQ     CMPS5A         ;BR IF NONE
83 061270 005237 017310      INC      ERRCNT
84 061274      ERRSOFT 3,EDDDE,ERR2 ;REPORT # OF MISMATCHES FOR MESSAGE
    061274 104457          TRAP      C$ERSOFT
    061276 000003          .WORD    3
    061300 030257          .WORD    EDDDE
    061302 041402          .WORD    ERR2
85 061304 004737 042224      CMPS5:  JSR      PC,LOGCMD ;LOG DATA ERROR IN COMPARE
86 061310      CMPS5A:
87 061310 005337 017274      DEC      DVRCT
88 061314 001267      BNE     CMPS3
89 061316 000137 061040      JMP     CMPNEW         ;IF NOT ALL DONE GO BACK
    061316 000137 061040      JMP     CMPNEW         ;GO BACK FOR NEXT TRIB
90
    
```



```

1          .SBTTL          INTERNAL END OF PASS CODE
2
3
4
5          ;**
6          ; FUNCTIONAL DESCRIPTION:
7          ; THIS CODE INCREMENTS THE PASS COUNT FOR THE
8          ; EVENT LOG. LOGS THE END OF PASS EVENT
9          ; IF "RPASS" IS A MINUS ONE RETURN TO MODE
10         ; DISPATCHER. IF NOT 1 THEN DECREMENT RPASS
11         ; AND IF "RPASS" IS THEN = TO 0 GO TO DCLT PROMPT
12         ; IN NOT = TO 0 THEN GO BACK TO MODE DISPATCHER
13
14         ; SUBORDINATE ROUTINES USED:
15
16         ;          "LOGEOP" - LOG END OF PASS TO EVENT LOG
17
18 061322 005237 017306          CMPSEX: INC          PSCNT          ;BUMP PASS COUNT
19
20 061326 013737 017304 017362          MOV          OPVAR1,TEMP5          ;LOG TX THRES
21 061334 013737 017302 017360          MOV          OPVAR,TEMP4          ;LOG RX THRESH
22 061342 013737 017306 017354          MOV          PSCNT,TEMP2          ;LOG PASS COUNT
23 061350 013737 017310 017356          MOV          ERRCNT,TEMP3
24 061356 004737 042250          JSR          PC,LOGEOP          ;LOG END OF PASS
25
26 061362 022737 177777 017412          CMP          #-1,RPASS          ;SEE IF RPASS=-1
27 061370 001403          BEQ          18          ;IF IT IS DON'T DECREMENT, LOOP FOREVER
28 061372 005337 017412          DEC          RPASS          ;DEC PASS COUNT
29 061376 001402          BEQ          28          ;IF DONE EXIT TEST
30 061470 000137 057674          18:          JMP          GTRX2          ;ELSE GO BACK AND DISPATCH
31 061404 005037 017416          28:          CLR          RUNNING          ; INIT "DCLT RUNNING" FLAG
32 061410 004737 064776          JSR          PC,HLTTRB          ;GO HALT ALL TRIBS BEFORE GOING BACK
33 061414 000137 052402          JMP          GTRAS          ;WHEN RPASS=0 GO BACK TO DCLT
34
35

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55

.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  
: DMP 11 MICROCODE STARTING AT LOC. 0. THIS IMAGE DATA WILL CONTAIN A  
: PROGRAM THAT WILL PRINT A MSG THAT DOWN LINE LOAD WAS SUCCESSFUL.

: 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 BMODE(R2) ;DISPATCH TO MODE BASED ON MODE TYPE IN R2

---

36 061420 012737 177777 015762 DLL: MOV #1,INDEX  
37 061426 004737 046462 JSR PC,GTVIND ;GET VALID INDEX ALSO FIRST TRIBN  
38 061432 013737 015756 017356 MOV TRIBN,TEMP3 ;MOV TRIBN TO TEMP3 FOR MTP DEFAULT  
39 061440 032737 00C001 023102 BIT #MTP,DEVPAR ;IS THIS MULTIPPOINT  
40 061446 001010 BNE 18 ;IF SO BRANCH  
41 061450 GMANID DLLQ1,TEMP3,0,377,0,377,NO  
061450 104443 TRAP C\$GMAN  
061452 000406 BR 100043  
061454 017356 .WORD TEMP3  
061456 000022 .WORD T\$CODE  
061460 025210 .WORD DLLQ1  
061462 000377 .WORD 377  
061464 000000 .WORD T\$LOLIM  
061466 000377 .WORD T\$MILIM  
061470 100043:

44 061470 113737 017356 002650 18: MOVB TEMP3,PASS1  
45 061476 113737 017356 002651 MOVB TEMP3,PASS2  
46 061504 113737 017356 002652 MOVB TEMP3,PASS3  
47 061512 113737 017356 002653 MOVB TEMP3,PASS4  
48 061520 052737 000100 017414 BIS #ERX,FLAG ;SET EXPECTED TO RX  
49 061526 042737 000002 017410 BIC #DATCKB,PARAM ;CLEAR NOCHECK  
50 061534 012737 002647 017342 MOV #DLLM1,CURADD ;SET THE DOWN LINE LOAD MSG TO #1  
51 061542 013737 002172 017334 MOV DLLM1,CURCC ;SET THE CC  
52 061550 004737 061642 JSR PC,DLTXRX ;GO TO THE DOWN LINE TX RX ROUTINE

;RETURN WHEN TX AND RX ARE COMPLETED

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 97 1  
 DOWN LINE LOAD SECTION

```

56 061554 012737 002654 017342      MOV      @DLLM2,CURADD      ;SET THE DOWN LINE LOAD MSG TO @
57 061562 013737 002174 017334      MOV      DLLM2C,CURCC      ;SET CC
58 061570 042737 001000 017414      BIC      @DLLGA,FLAG      ;CLEAR THE GO AHEAD FLAG
59 061576 004737 061642                JSR      PC,DLTXRX        ;GO TO THE DOWN LINE TX RX ROUTINE
60
61                                ; RETURN WHEN TX AND RX ARE COMPLETED
62 061602      DLLPRI:
63 061602      PRINTF  @DLLCM
                                MOV      @DLLCM,-(SP)
                                MOV      @1,(SP)
                                MOV      SP,RO
                                TRAP    C$PNTF
                                ADD     @4,SP
                                061602 012746 027160
                                061606 012746 000001
                                061612 010600
                                061614 104417
                                061616 062706 000004
64 061622 000137 052402      JMP      GTRAS
65
66 061626      DLLEA:
67 061626      ERRSOFT 13,DLLAB,ERR14
                                TRAP    C$ERSOFT
                                .WORD  13
                                .WORD  DLLAB
                                .WORD  ERR14
                                061626 104457
                                061630 000015
                                061632 040022
                                061634 041512
68
69 061636 000137 052402      JMP      GTRAS      ;PRINT ABORT AND EXIT
70
71
72
73 061642      DLTXRX:
74 061642 052737 000004 017414      BIS      @QRX,FLAG      ;SET THE QUE RX FLAG
75 061650 012737 005416 017270      MOV      @RXBUF,DVRXA   ;SET THE DEVICE RX BUFFER TO RXBUF
76 061656 012737 005416 017354      MOV      @RXBUF,TEMP2   ;SET UP FOR LOG
77 061664 012737 000400 017272      MOV      @256.,DVRCC    ;SET UP FOR CC OF 256
78 061672 012737 000400 017356      MOV      @256.,TEMP3    ;SET UP FOR LOG
79 061700 004737 064006                JSR      PC,DVRXQ        ; GO QUE RX
80 061704 004737 042056                JSR      PC,LOGRXQ      ;AND LOG IT...
81
82 061710 013737 017342 017250      MOV      CURADD,DVTXA   ;SET UP FOR TX
83 061716 013737 017342 017354      MOV      CURADD,TEMP2   ;AND LOG
84 061724 013737 017334 017252      MOV      CURCC,DVTCC    ;SE UP FOR TX COUNT
85 061732 013737 017334 017356      MOV      CURCC,TEMP3    ;AND LOG IT
86 061740 004737 042022                JSR      PC,LOGTXQ      ;LOG THE TX QUEUED
87 061744 052737 000210 017414      BIS      @QTX!ETX,FLAG  ;SET UP TO QUE AND EXPECTED
88 061752 004737 064070      DLLE2: JSR      PC,DVTXRX     ;GO TO DEVICE ROUTINE
89 061756 032737 001000 017414      BIT      @DLLGA,FLAG    ;TEST FOR GO AHEAD BIT
90 061764 001047                BNE     DLLE1           ;IF SET GO TO ONE
91 061766 032737 000010 017414      BIT      @QTX,FLAG      ;ELSE CHECK FOR TX DONE
92 061774 001020                BNE     DLLE6           ;IF DONE THEN BRANCH
93                                ;ELSE ERROR
94 061776 012737 041232 017366      MOV      @TXNC,CONOTM   ;
95 062004 013737 005416 017356      DLLE7: MOV      RXBUF,TEMP3
96 062012 013737 003416 017360      MOV      TXBUF,TEMP4
97 062020 012737 040022 017354      DLLE7A: MOV      @DLLAB,TEMP2
98 062026 004737 042104                JSR      PC,LGDVE       ;LOG ERROR
99 062032 000137 061626                JMP      DLLEA          ;ADORT TEST
100
101 062036 013737 017250 017354      DLLE6: MOV      DVTXA,TEMP2
102 062044 013737 017252 017356      MOV      DVTCC,TEMP3
103 062052 004737 042040                JSR      PC,LOGTXC     ;LOG TX DONE
    
```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 97 2  
 DOWN LINE LOAD SECTION

```

104 062056 042737 000210 017414 BIC #QTX!ETX,FLAG ;CLEAR QUE AND EXPECTED
105 062064 052737 001000 017414 BIS #DLLGA,FLAG ;SET THE GO AHEAD BIT
106 062072 023737 002174 017252 CMP DLLM2C,DVTCC
107 062100 001475 BEQ DLLE5 ;EXIT IF SECOND MSG.
108 062102 000723 BR DLLE2 ;AND GO BACK TO 2
109 062104 032737 000004 017414 DLLE1: BIT #QRX,FLAG ;IS THE A RX COMPLETED
110 062112 001004 BNE DLLE8 ;IF SO GO TO 8
111 062114 012737 041252 017366 MOV #RXNC,CONOTM ;ELSE SET UP ERROR AND ABORT.
112 062122 000730 BR DLLE7
113 062124 013737 017270 017354 DLLE8: MOV DVRXA,TEMP2
114 062132 013737 017272 017356 MOV DVRCC,TEMP3
115 062140 004737 042074 JSR PC,LOGRXC ;LOG RECEIVE COMPLETE
116 062144 122737 000010 005416 CMPB #10,RXBUF ;CHECK FOR FIRST WORD OF RX
117 ;SEC BOOT MSG.
118 062152 001404 BEQ DLLE3
119 062154 012737 041272 017366 DLLE4: MOV #RXM1,CONOTM ;SET UP MMSG AND ABORT
120 062162 000710 BR DLLE7 ;ABORT TEST
121
122 062164 122737 000001 005420 DLLE5: CMPB #1,RXBUF+2 ;IS SECOND WORD 1 ?
123 062172 001407 BEQ DLLE5A ;YES,BRANCH
124 062174 012737 041315 017366 MOV #RXM2,CONOTM
125 062202 013737 005420 017356 MOV RXBUF+2,TEMP3
126 062210 000703 BR DLLE7A ;SET UP MESSAGE AND ABORT
127
128
129 ;;PRINT ID OF DEVICE REQUESTING LOAD REV B BY EC
130 062212 012737 032062 017350 DLLE5A: MOV #UNKM,TEMP ;SET UP FOR UNKNOWN DEVICE
131 062220 113703 005417 MOVB RXBUF+1,R3 ;GET DEVTYPE FROM MESSAGE
132 062224 120327 000042 CMPB R3,#34. ;OUT OF LEGAL RANGE ?
133 062230 101006 BHI DLLE5B ;YES,BRANCH
134 062232 132703 000001 BITB #1,R3 ;ODD ?
135 062236 001003 BNE DLLE5B ;YES,BRANCH
136 062240 016337 023222 017350 MOV DLLIND(R3),TEMP ;GET ASCIZ MESSAGE FROM TABLE
137
138 062246 DLLE5B: PRINTF #SECRM,TEMP,R3 ;PRINT ID MESSAGE
139 MOV R3,-(SP)
140 MOV TEMP,(SP)
141 062250 013746 017350 MOV #SECRM,-(SP)
142 062254 012746 031724 MOV #3,(SP)
143 062260 012746 000003 MOV SP,R0
144 062264 010600 TRAP C#PNTF
062266 104417 ADD #10,SP
062270 062706 000010

```

```

1          .SBTTL          TALK MODE SECTION
2
3          ;**
4          ; FUNCTIONAL DESCRIPTION:
5          ; TALK MODE SECTION
6          ; IN THIS MODE, THE "TALK" END OF THE LINK TRANSMITS OPERATOR
7          ; SPECIFIED MESSAGES UNTIL A "EXIT" MESSAGE IS TYPE. AT THAT POINT,
8          ; THIS END OF THE LINK GOES INTO "LISTEN" MODE.
9
10         ; SUBORDINATE ROUTINES USED:
11
12         ; "LOGTXQ" - LOG TX BUFFER QUED TO EVENT LOG
13         ; "DVTXRX"  QUE TX BUFFER TO DEVICE AND WAIT FOR COMPLETE
14         ; "LOGTXC"  LOG TX COMPLETE TO EVENT LOG
15
16         ; CALLING SEQUENCE:
17         ;     JMP     @MODE(R2)          ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
18         ;--
19
20 062276 012737 177777 015762 TALCK: MOV     @-1,INDEX
21 062304 004737 046462          JSR     PC,GTVIND          ;GET FIRST TRIB
22 062310 042737 000002 017410 BIC     @DATCKB,PARAM      ;SET NOCHECK
23 062316 012702 002524          MOV     @OPBUF,R2
24 062322 012722 177777          1$:   MOV     @-1,(R2)+        ;CLEAR OUT OPBUFFER FIRST
25 062326 022702 002646          CMP     @OPEND,R2
26 062332 001373          BNE     1$
27 062334          GMANID OPRMM,OPBUF,A,0,1,72,.NO      ;GET TALK MESSAGE
28          062334 104443          TRAP   C$GMAN
29          062336 000406          BR     10005$
30          062340 002524          .WORD OPBUF
31          062342 000142          .WORD T$CODE
32          062344 027114          .WORD OPRMM
33          062346 000000          .WORD 0
34          062350 000001          .WORD T$LOLIM
35          062352 000110          .WORD T$HILIM
36          062354          10005$:
37 062354 005002          CLR     R2                ;NOW GET CHAR COUNT
38 062356 122762 000377 002524 2$:   CMPB   @377,OPBUF(R2)
39 062364 001402          BEQ     3$
40 062366 005202          INC     R2
41 062370 000772          BR     2$
42 062372 010237 002166          3$:   MOV     R2,OPCNT
43 062376 012737 002524 017250 MOV     @OPBUF,DVTXA       ;SET UP TX ADDR.
44 062404 012737 002524 017354 MOV     @OPBUF,TEMP2
45 062412 013737 002166 017356 MOV     OPCNT,TEMP3
46 062420 013737 002166 017252 MOV     OPCNT,DVTCC       ;SET UP TX CC
47 062426 004737 042022          JSR     PC,LOGTXQ
48 062432 052737 000210 017414 BIS     @QTX!ETX,FLAG      ;SET UP FLAGS
49 062440 005037 017336          CLR     CPTRR            ;CLEAR RX POINTER
50
51 062444 004737 064070          JSR     PC,DVTXRX
52
53 062450 013737 017250 017354 MOV     DVTXA,TEMP2
54 062456 013737 017252 017356 MOV     DVTCC,TEMP3
55 062464 004737 042040          JSR     PC,LOGTXC
56 062470 022737 054105 002524 CMP     @"EX,OPBUF        ;CHECK FOR EXIT

```

G15

SEQ 18A

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 98 1  
TALK MODE SECTION

49	062476	001277			BNE	TALCK	
50	062500	022737	052111	002526	CMP	#"IT,OPBUF+2	
51	062506	001273			BNE	TALCK	
52	062510	042737	000210	017414	BIC	#QTX!ETX,FLAG	;CLEAR THE TX BITS
53	062516	012737	000006	017402	MOV	#LIS,MODTYP	;CHANGE TO LISTEN MODE
54	062524	000137	057674		JMP	GTRX2	;AND GO BACK TO DISPATCH

```

1      .SBTTL          LISTEN MODE SECTION
2
3      ;**
4      ; FUNCTIONAL DESCRIPTION:
5      ; LISTEN MODE SECTION
6      ; IN THIS MODE, THE "LISTEN" END OF THE LINK PRINTS ALL OF THE MESSAGES
7      ; RECEIVED BY THE DEVICE ON THE OPERATOR'S CONSOLE. IF THE MESSAGE
8      ; RECEIVED IS AN "EXIT" MESSAGE, THEN THE NODE ENTERS 'TALK' MODE.
9
10     ; SUBORDINATE ROUTINES USED:
11
12     ; "DVRXQ" - QUE RECEIVE BUFFER SPACE TO DEVICE
13     ; "LOGRXQ" LOG RECEIVE BUFFER QUED TO EVENT LOG
14     ; "DVTXRX" WAIT FOR RX TO COMPLETE
15     ; "LOGRXC" LOG RX COMPLETE TO EVENT LOG
16
17     ; CALLING SEQUENCE:
18     ;     JMP      @MODE(R2)          ;DISPATCH TO MODE BASED ON MODE TYPE IN R2
19     ; --
20
21 062530 012737 177777 015762 LISCK:  MOV      @-1,INDEX
22 062536 004737 046462          JSR      PC,GTVIND          ;GET FIRST TRIB
23 062542 042737 000002 017410          BIC      @DATCKB,PARAM      ;CLEAR CHECK BIT
24 062550          PRINTF      @LISP          ;PRINT PROMPT FOR OPR.
25 062550 012746 027103          MOV      @LISP,(SP)
26 062554 012746 000001          MOV      @1,-(SP)
27 062560 010600          MOV      SP,R0
28 062562 104417          TRAP    C$PNTF
29 062564 062706 000004          ADD     @4,SP
30 062570 012737 002524 017270 LISCKA: MOV     @OPBUF,DVRXA      ;SET DEVICE UP TO REC AT OPBUF
31 062576 012737 002524 017354          MOV     @OPBUF,TEMP2
32 062604 012737 000122 017272          MOV     @82.,DVRCC         ;SET UP CHAR COUNT TO 82.
33 062612 012737 000122 017356          MOV     @82.,TEMP3
34 062620 052737 000104 017414          BIS     @QRX!ERX,FLAG      ;SET UP FLAG
35 062626 005037 017340          CLR     CPTR              ;CLEAR THE TX.
36
37          JSR     PC,DVRXQ      ;QUE RX
38 062632 004737 064006          JSR     PC,LOGRXQ
39 062636 004737 042056
40
41          JSR     PC,DVTXRX     ;GO TO DEVICE RX. SUBROUTINE
42
43 062642 004737 064070
44
45 062646 013737 017270 017354          MOV     DVRXA,TEMP2
46 062654 013737 017272 017356          MOV     DVRCC,TEMP3        ;SET UP ADDR. AND CC.
47 062662 004737 042074          JSR     PC,LOGRXC         ;LOG COMPLETED
48 062666 063737 017270 017272          ADD     DVRXA,DVRCC
49 062674 105077 134372          CLRB   @DVRCC
50 062700          PRINTF  @OPBFPT
51 062700 012746 002520          MOV     @OPBFPT,(SP)
52 062704 012746 000001          MOV     @1,(SP)
53 062710 010600          MOV     SP,R0
54 062712 104417          TRAP    C$PNTF
55 062714 062706 000004          ADD     @4,SP
56
57 062720 022737 054105 002524          CMP     @"EX,OPBUF        ;COMPARE FOR EX OF 'EXIT'
58 062726 001320          BNE     LISCKA           ;IF NOT EXIT THEN GO BACK
59 062730 022737 052111 002526          CMP     @"IT,OPBUF+2     ;IF FIRST HALF OK CHECK NEXT PART
60 062736 001314          BNE     LISCKA           ;IF NOT EXIT THEN GO BACK
61 062740 012737 000005 017402          MOV     @TAL,MODTYP      ;CHANGE MODE TO TALK

```

I15

CZCLMCO DMP/V 11 DCL T MACRO V05.00 Thursday 22 Mar 84 16:24 Page 99-1  
LISTEN MODE SECTION

SEQ 190

48 062746 000137 057674  
49  
50

JMP

GTRX2

;RETURN TO DISPATCHER



```

1          .SBTTL          DEVICE FUNCTION SUBROUTINES
2
34
35
36
37          .SBTTL          DEVICE INIT SUBROUTINE
38
56
57          ;**
58          ; FUNCTIONAL DESCRIPTION:
59          ;   DVINIT DEVICE INIT ROUTINE
60          ;   THIS ROUTINE IS DEVICE DEPENDENT CODE THAT INITIS
61          ;   THE DEVICE BEING TESTED.
62
63          ; INPUTS:          "FHDPLX" INDICATES IF MODE IS FULL OR HALF DUPLEX. (1=FULL)
64          ;                   ADDRESS POINTERS (SELO,...) ALREADY POINT TO DEVICE'S REG.S
65
66          ; SUBORDINATE ROUTINES USED:
67
68          ;                   "LGDVE" - LOG DEVICE ERROR TO EVENT LOG
69
70
71          ; CALLING SEQUENCE:
72          ;                   JSR          PC,DVINIT
73          ; -
74
75          DVINIT:
76
83          062752 012737 001000 017456          MOV          #1000,TIMER1          ;SET UP TIMER 1 FOR 1000(OCTAL) TICKS
84          062760 022737 000004 017402          CMP          #00W,MODTYP
85          062766 001034                          BNE          DVIN4          ;BRANCH IF NOT DLL
86          062770 022737 000001 023102          CMP          #1,DEVPAR          ;IS THIS TRIB
87          062776 001030                          BNE          DVIN4          ;BRANCH IF CONTROL OR PTP
88          063000 012777 060000 140044          MOV          #60000,@SELO          ;SET MCLR AND ENTER P MOP
89          063006 005737 017456          DVIN4A: TST          TIMER1
90          063012 001375                          BNE          DVIN4A          ;IF TIMER RUNS OUT IT MEANS
91          063014 012737 037725 017354          MOV          #DVEM8,TEMP2          ;SWITHCES ARE NOT SET CORRECTLY
92          063022 017737 140024 017356          MOV          @SELO,TEMP3
93          063030 017737 140022 017360          MOV          @SEL2,TEMP4
94          063036 004737 042104          JSR          PC,LGDVE
95          063042 005237 017310          INC          ERRCNT
96          063046          ERRSOFT 14,DVEM8,ERR13
97          063046 104457                          TRAP          C$ERSOFT
98          063050 000016                          .WORD          14
99          063052 037725                          .WORD          DVEM8
100         063054 041460                          .WORD          ERR13
101         97 063056 000735
102         98 063060 012777 040000 137764 DVIN4: BR          DVINIT
103         99 063066 022737 000004 023100          MOV          #MCLR,@SELO          ;DO A MASTER CLEAR
104         100 063074 001005                          CMP          #DMP6,OPTYP          ;IS THIS A 8206
105         101 063076 112777 000200 137750          BNE          DVIN6          ;IF NOT GO TO 6
106         102 063104 000240                          MOVB         #200,@BSEI 1          ;SET RUN FOR 8206
107         103 063106 000240                          NOP
108         104 063110 005777 137736          DVIN6: TST          @SELO          ;SLIGHT DELAY
109         105 063114 100426                          BMI          DVIN1          ;IS RUN BIT SET
110         106 063116          BREAK          ;IF YES GO TO 1 ELSE...
111         063116 104422                          TRAP          C$BRK
    
```



CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 22-Mar-84 16:24 Page 100 2  
 DEVICE INIT SUBROUTINE

```

155
156 063334 042737 000003 017414 ; DVIN12: BIC #3,FLAG ;CLEAR INPUT AND OUTPUT INTERRUPT FLAGS
157 063342 112777 000221 137502 ; MOVB #221,@BSELO ;SET RQI, IEO, AND IEI
158 063350 004737 065114 ; JSR PC,TOORIO ;GO WAIT FOR INPUT INTERRUPT(OK TO WRITE)
159
160 ;
161 ; NOW SET UP NETWORK CONFIGURATION AND LINE CHARACTERISTIC
162 063354 113777 023104 137504 ; MOVB STATYP,@BSEL6 ;SET UP STATION TYPE(PT-PT,MULTI-PT CNTL/TRIB)
163 063362 005737 017406 ; TST FHDPLX ;HALF/DUPLEX ?
164 063366 001403 ; BEQ 1$ ;YES,BRANCH
165 063370 052777 000001 137470 ; BIS #BIT0,@BSEL6 ;SET FULL DUPLEX BIT
166 063376 112777 000002 137452 1$: MOVB #2,@BSEL2 ;DO MODE DEFINITION COMMAND
167 063404 142777 000010 137442 ; BICB #BIT3,@BSEL1 ; CLEAR DIAGNOSTIC MODE (DMV ONLY)
168
169 ;
170 ; NOW CHECK TO SEE IF ITS A DMP AND INTERNAL LOOPBACK REV C EC
171 063412 022737 000001 017404 ; CMP #1,MLTYP ; INTERNAL LOOP ?
172 063420 001007 ; BNE 3$ ; NO, BRANCH
173 063422 005737 023100 ; TST OPTYP ; DMP ?
174 063426 001012 ; BNE DVES1A ; NO, BRANCH
175
176 ;
177 ; NOW SET THE DMP INTERNAL LOOP BIT
178
179 063430 152777 000010 137416 ; BISB #BIT3,@BSEL1 ; SET LU LOOP
180 063436 000406 ; BR DVES1A ; SKIP OVER WAIT
181
182 ;
183 ; NOW WAIT A SECOND FOR THINGS TO SETTLE
184
185 063440 012737 000001 017462 3$: MOV #1,TIMERS ; SET TIMER FOR 1 SECOND
186 063446 005737 017462 4$: TST TIMERS ; DONE ?
187 063452 001375 ; BNE 4$ ; NO,BRANCH
188
189 ;
190 ;WRITE GLOABL PARMAS
191
192 063454 ; DVES1A:
193 063454 005037 015756 ; CLR TRIBN ;MAKE TRIBN 0
194 063460 012737 017220 021022 ; MOV #GLBPLS,TSSE ;TSSE POINTS TO LIST
195 063466 004737 064624 ; JSR PC,WRIPPG ;WRITE POLL PARAMS
196 063472 012737 000110 017350 ; MOV #110,TEMP
197 063500 022737 000003 017404 ; CMP #MODLOC,MLTYP ;IS THIS MODEM LOCAL
198 063506 001407 ; BEQ 1$ ;BRANCH IF MODEM LOCAL
199 063510 012737 000104 017350 ; MOV #104,TEMP
200 063516 022737 000004 017404 ; CMP #MODREM,MLTYP ;IS THIS REM
201 063524 001002 ; BNE 2$ ;BRANCH IF NOT
202 063526 004737 064732 1$: JSR PC,WRMCS ;GO WRITE MODEM CONTROL
203
204 063532 012737 177777 015762 2$: MOV #-1,INDEX ;MAKE INDEX =-1
205 063540 004737 046462 DVES1: JSR PC,GTVIND ;GET VALID INDEX
206 063544 022737 000040 015762 ; CMP #32,INDEX ;DONE
207 063552 001475 ; BEQ DVINEX ;IF SO EXIT
208
209 ;ESTABLISH TRIB
210
211 063554 152777 000200 137270 DVEST: BISB #RQI,@BSELO ;DO REQUEST IN

```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 100 3  
 DEVICE INIT SUBROUTINE

```

212 063562 004737 065114          JSR    PC,TOORIO      ;WAIT TIL PORT IS OURS
213 063566 113777 015756 137264  MOVB   TRIBN,@SEL3   ;SET UP TRIB NO
214 063574 012777 000001 137264  MOV    #01,@SEL6    ;ESTABLISH TRIB
215 063602 112777 000001 137246  MOVB   #01,@SEL2    ;CLEAR RDI AND DO COMMAND.
216
217                                ;WRITE POLL PARAMS IF NESC.
218 063610 022737 000003 023102  CMP    #3,DEVPAR    ;IS THIS A MULTIPOINT CONTROL
219 063616 001022                BNE    POLLEN       ;BRANCH IF NOT.
220 063620 004737 047154          JSR    PC,GETIND     ;GET VALID INDEX
221 063624 013737 015762 017232  MOV    INDEX,MPLY   ;MOVE INDEX TO MULTIPLIER
222 063632 012737 000020 017350  MOV    #16.,TEMP    ;
223 063640 012737 016220 017354  MOV    #POLLIS,TEMP2
224 063646 004737 046436          JSR    PC,MTPLY     ;RETURN WITH ADDRESS
225                                ;OF FIRST WORD IN TEMP2
226
227 063652 013737 017354 021022  MOV    TEMP2,TSSE   ;
228 063660 004737 064634          JSR    PC,WRIPP     ;WRITE POLL PARAMS
229
230                                ; ISTRT TRIB
231
232 063664 152777 000200 137160  POLLEN: BISB   #RQI,@SELO   ;REQUEST IN
233 063672 004737 065114          JSR    PC,TOORIO    ;WAIT TIL PORT IS OURS.
234 063676 113777 015756 137154  MOVB   TRIBN,@SEL3
235 063704 012777 000004 137154  MOV    #04,@SEL6    ;MAKE IT MAINT MODE
236 063712 022737 000004 017402  CMP    #DOW,MODTYP  ;IS THIS DOWN LINE LOAD
237 063720 001406                BEQ    POLLE2
238 063722 012777 000003 137136  MOV    #03,@SEL6    ;TO ISTRT
239 063730 052737 000400 017414  BIS    #RUNST,FLAG  ;SET THE RUN STATE FLAG
240 063736 112777 000001 137112  POLLE2: MOVB   #01,@SEL2  ;DO COMMAND
241 063744 000675                BR     DVES1        ;GO BACK
242 063746 052737 002000 017414  DVINEX: BIS    #INOV,FLAG ;INDICATE INIT CODE IS DONE
243 063754 000207                RTS    PC           ;RETURN TO CALLER
244
245
246
247
248

```

1  
2  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
39  
40  
41  
42  
43  
44  
45

.SBTTL DFVICE GET MODEM STATUS SUBROUTINE

```

:++
: FUNCTIONAL DESCRIPTION:
:   "DVMODS" GET MODEM STATUS
:
: IMPLICIT INPUTS:
:   THE BIT POSITION AND AVAILABILITY OF THE MODEM SIGNALS CTS,DSR,...RI,,
:   IN THE DEPENDENT PORTION OF THE GLOBAL EQUATES SECTION.
:
: OUTPUTS:
:   CURRENT MODEM SIGNAL VALUES IN "MODS"
:
: SUBORDINATE ROUTINES USED:
:
: CALLING SEQUENCE:
:   JSR PC,DVMODS
:
    
```

```

40 063756 152777 000200 137066 DVMODS: BISB #RQI,@BSEL0 ;SET RQI
41 063764 004737 065114 JSR PC,TOORIO ;GO TIME OUT CHECK
42 063770 012777 000020 137070 MOV #20,@SEL6 ;READ MODEM STATUS
43 063776 112777 000001 137052 MOVB #01,@BSEL2 ;DO CONTROL IN
44 064004 000207 RTS PC ;RETURN TO CALLER
45
    
```

```

1      .SBTTL                DEVICE QUEUE RECEIVE SPACE SUBROUTINE
13
14
15      ;**
16      ; FUNCTIONAL DESCRIPTION:
17      ;   DVRXQ  THIS SUB ROUTINE QUES THE REC BUFFER SPACE TO THE
18      ;           DEVICE, THEN CLEARS THE QRX BIT OF THE FLAG WORD.
19
20      ; INPUTS:
21      ;   DVRXA = ADDRESS OF RX BUFFER SPACE
22      ;   DVRCC = BYTE CHAR COUNT OF RX BUFFER
23      ;   QRX FLAG BIT = SET BY CALLING ROUTINE
24
25      ; OUTPUTS:
26      ;   QRX FLAG BIT = CLEARED BY ROUTINE
27
28      ; SUBORDINATE ROUTINES USED:
29
30      ; CALLING SEQUENCE:
31      ;   JSR    PC,DVRXQ
32      ;
33
34      DVRXQ:
35      064006 032737 000004 017414    BIT    #QRX,FLAG
36      064014 001424                    BEQ    DVREX                ;IF NOT RX THEN EXIT
37
38      064016 042737 000004 017414    BIC    #QRX,FLAG          ;CLEAR FLAG FOR RX
39
40      064024 152777 000200 137020    BISB  #RQI,#BSELO        ;SET UP REQUEST
41      064032 004737 065114                    JSR    PC,TOORIO          ;GO CHECK FOR IN OR OUT
42
43      064036 013777 017270 137016    MOV    DVRXA,#SEL4
44      064044 013777 017272 137014    MOV    DVRCC,#SEL6        ;LOAD CC AND ADDR
45      064052 113777 015756 137000    MOVB  TRIBN,#SEL3        ;SET UP TRIB NO.
46      064060 112777 000000 136770    MOVB  #0,#SEL2          ;DO COMMAND.
47
48
49
50
51
52
53
54      064066 000207                    DVREX: RTS    PC          ;RETURN TO CALLER
55

```

```

1          .SBTTL                DEVICE TRANSMIT AND RECEIVE SUBROUTINE
2
25
26
27          ;**
28          ; FUNCTIONAL DESCRIPTION:
29          ; DVTXRX-DEVICE TRANSMIT AND RECEIVE ROUTINE
30          ; THIS CODE QUES THE TRANSMIT BUFFER TO THE DEVICE
31          ; IF NEEDED. THE CODE THEN WAITS FOR A TX COMPLETE,
32          ; RX COMPLETE OR BOTH. THE CODE REPORTS A TIME OUT
33          ; ERROR IF NEITHER IS REPORTED BACK IN
34          ; 60 SECONDS. AFTER REPORTING ERROR TIMER IS RE ,STARTED
35          ; AND DEVICE WILL CONTINUE TO WAIT FOR INTERRUPT.
36
37          ; INPUTS:
38          ; "DVTXA" = ADDRESS OF TRANSMIT MSG.
39          ; "DVTCC" = BYTE COUNT OF TRANSMIT MSG.
40          ; "QTX" BIT = SET IF TRANSMIT REQUESTED
41          ; "ETX" BIT = SET IF TRANSMIT EXPECTED
42          ; "ERX" BIT = SET IF RECEIVE EXPECTED
43
44          ; OUTPUTS:
45          ; "DVTXA" = ADDRESS OF TX MSG. COMPLETED
46          ; "DVTCC" = BYTE COUNT OF TX MSG. COMPLETED
47          ; "QTX" = SET IF TX COMPLETED
48          ; "DVRXA" = ADDRESS OF RX MSG. COMPLETED
49          ; "DVRCC" = BYTE COUNT OF RX MSG. COMPLETED
50          ; "QRX" = SET IF RX COMPLETED
51
52          ; SUBORDINATE ROUTINES USED:
53
54
55          ; CALLING SEQUENCE:
56          ; JSR PC,DVTXRX
57          ;
58
59 064070 032737 000010 017414 DVTXRX: BIT @QTX,FLAG ;ANY TX TO QUE
60 064076 001424 ;BEQ DVTR3 ;IF NOT GO WAIT FOR OUTPUT
61 064100 042737 000010 017414 BIC @QTX,FLAG ;CLEAR FLAG
62
63
64
65
66
67
68
69 064106 152777 000200 136736 BISB @RQI,@BSELO ;SET REQUEST
70 064114 004737 065114 JSR PC,TOORIO ;GO CHECK FOR IN OR OUT
71 064120 013777 017250 136734 MOV DVTXA,@SEL4
72 064126 013777 017252 136732 MOV DVTCC,@SEL6
73 064134 113777 015756 136716 MOVB TRIBN,@BSEL3 ;SET UP TRIB NO.
74 064142 112777 000004 136706 MOVB @4,@BSEL2 ;DO COMMAND
75
76 064150 DVTR3:
77 064150 012737 000074 017462 MOV @60 ,TIMERS ;SET TIMER FOR 60 SECS
78 064156 005737 015766 TOINOT: TST CRX
79 064162 001050 BNE DVTR4 ;BRANCH IF RX COMPLETED
80 064164 005737 015764 TST CIX
81 064170 001045 BNE DVTR4 ;BRANCH IF TX COMPLETED
82
83
84
85
86
87
88
89 064172 005737 017462 TST TIMERS ;IS TIMER EXPIRED
    
```

```

90 064176 001025          BNE      TOIN1
91 064200 012737 037257 017354    MOV      @DVEM2,TEMP2
92 064206 017737 136640 017356    MOV      @SELO,TEMP3
93 064214 017737 136636 017360    MOV      @SEL2,TEMP4
94 064222 117737 136632 015756    MOVB     @SEL3,TRIBN
95 064230 004737 042104          JSR      PC,LGDVE
96 064234 005237 017310          INC      ERRCNT
97 064240          ERRSOF T 6,DVEM2,ERR13
    064240 104457          TRAP    C1ERSOF T
    064242 000006          .WORD  6
    064244 037257          .WORD  DVEM2
    064246 041460          .WORD  ERR13
98 064250 000737          BR       DVTR3          ;RETURN TO CHECK TIMER
99
100
101 064252          TOIN1:  BREAK
    064252 104422          TRAP    C1BRK
102 064254 032737 000002 017414    TOIN2:  BIT      @OTINT,FLAG
103 064262 001735          BEQ     TOINOT          ;IF NOT OUTPUT GO BACK AND
                                ;CHECK TIMER AGAIN
104                                ;ELSE HANDLE OUTPUT AND RETURN
105 064264 004737 065254          JSR      PC,OUTHDL
106 064270 005737 015766          TST     CRX
107 064274 001003          BNE     DVTR4          ;IF TX GO TO 4
108 064276 005737 015764          TST     CTX
109 064302 001725          BEQ     TOINOT          ;BRANCH IF NOT RX OR TX COMPLETED
110 064304 005737 015764          TST     CTX          ;IS IT TX COMPLETED
111 064310 001456          BEQ     DVTR5          ;IF NOT TRY RX
112 064312 032737 000200 017414    BIT      @ETX,FLAG      ;IF SO SHOULD IT BE
113 064320 001023          BNE     DVTR4A         ;IF IT SHOULD GO TO 4A
114 064322 012737 037514 017354    MOV      @DVEM5,TEMP2   ;ELSE LOG ERROR
115 064330 013737 066644 017356    MOV      TSEL4,TEMP3
116 064336 013737 066642 017360    MOV      TSEL6,TEMP4
117 064344 013737 066646 015756    MOV      TSEL3,TRIBN
118 064352 004737 042104          JSR      PC,LGDVE
119 064356          ERRSOF T 9,DVEM5,ERR13 ;REPORT ERROR
    064356 104457          TRAP    C1ERSOF T
    064360 000011          .WORD  9
    064362 037514          .WORD  DVEM5
    064364 041460          .WORD  ERR13
120
121 064366 000425          BR       DVTR4B        ;THEN CLEAR COMPL.FLAG
122 064370 013702 015774          DVTR4A: MOV      TSPTR,R2
123 064374 014237 017350          MOV      -(R2),TEMP
                                ;UNLOAD TRIBN
124 064400 113737 017351 015756    MOVB     TEMP+1,TRIBN
125 064406 105037 015757          CLRB    TRIBN+1
126 064412 013737 015756 017254    MOV      TRIBN,DVTTB    ;UNLOAD TRIB NUMBER
127 064420 014237 017250          MOV      (R2),DVTXA    ;UNLOAD CC
128 064424 014237 017252          MOV      (R2),DVTCC    ;UNLOAD ADDRESS
129 064430 010237 015774          MOV      R2,TSPTR
130 064434 052737 000010 017414    BIS      @QTX,FLAG      ;AND SET TX COMPL FLAG
131 064442 005337 015764          DVTR4B: DEC     CTX      ;AND COUNT DOWN FLAG
132 064446 005737 015766          DVTR5:  TST     CRX
133 064452 001463          BEQ     DVTREX        ;IF NOT THEN EXIT.
134 064454 032737 000100 017414    BIT      @ERX,FLAG      ;TEST IS THIS SUPPOSED TO BE RX
135 064462 001023          BNE     DVTR5A        ;IF YES PROCESS AS SUCH
136 064464 012737 037571 017354    MOV      @DVEM6,TEMP2
137 064472 013737 066650 017356    MOV      RSEL4,TEMP3
    
```



CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 103 2  
 DEVICE TRANSMIT AND RECEIVE SUBROUTINE

```

138 064500 013737 066654 015756      MOV      RSEL3,TRIBN
139 064506 013737 066652 017360      MOV      RSEL6,TEMP4      ;ELSE
140 064514 004737 042104                JSR      PC,LGDVE         ;LOG ERROR
141 064520                ERRSOFT 10,DVEM6,ERR13
      064520 104457
      064522 000012
      064524 037571
      064526 041460
142
143 064530 000432                BR       DVTRX1          ;AND EXIT
144
145 064532 013702 015770      DVTR5A: MOV      RSPTRS,R2
146 064536 012237 017272                MOV      (R2)+,DVRCC
147 064542 012237 017270                MOV      (R2)+,DVRXA      ;UNLOAD ADDR
148 064546 012237 017350                MOV      (R2)+,TEMP
149 064552 113737 017351 015756      MOV      TEMP+1,TRIBN
150 064560 105037 015757                CLR      TRIBN+1
151 064564 013737 015756 017266      MOV      TRIBN,DVRTB      ;UNLOAD TRIBN
152 064572 020227 016166                CMP      R2,@RXSKEN      ;IS IT AT THE END
153 064576 001002                BNE      2$
154 064600 012702 016012                MOV      @RXSTAK,R2      ;START OVER
155 064604 010237 015770 2$:      MOV      R2,RSPTRS      ;RELOAD POINTER
156 064610 052737 000004 017414      BIS      @QRX,FLAG
157 064616 005337 015766      DVTRX1: DEC      CRX      ;COUNT DOWN CRX
158
159 064622 000207      DVTR5A: RTS      PC      ;AND EXIT
160

```

```

TRAP      C#ERSOFT
.WORD     10
.WORD     DVEM6
.WORD     ERR13

```

```

1      .SBTTL          DEVICE DEPENDENT SUBROUTINES
2      .SBTTL          WRITE POLL PARAMETERS
3      ;**
4      ; FUNCTIONAL DESCRIPTION:      WRIPP  WRITE POLL PARAMETERS
5      ; WRITE ALL POLLING PARAMETERS FROM LIST
6      ; POINTED TO BY TSSE FOR TRIB NUMBER IN TRIBN
7      ;
8      ; INPUTS:
9      ; TRIBN  TRIB NUMBER OF WRITE
10     ; TSSE  - ADDRESS OF POLL LIST
11     ;
12     ;
13     ; CALLING SEQUENCE:
14     ; JSR    PC,WRIPP          ;FOR TRIBS
15     ; JSR    PC,WRIPPG        ;FOR GLOBAL
16     ;
17 064624 012737 000233 021024 WRIPPG: MOV    #233,TSSA      ;LOAD TSSA WITH ADDR UF IS GLOBAL PP.
18 064632 000403                BR      WRIP1          ;THEN GO TO 1
19 064634 012737 000230 021024 WRIPP:  MOV    #230,TSSA      ;LOAD TSSA WITH ADDR OF 1ST POLPAR.
20 064642 152777 000200 136202 WRIP1: BISB  #RQI,@SELO    ;DO REQUEST IN
21 064650 004737 065114                JSR    PC,TOORIO      ;WAIT TIL PORT IS OURS
22 064654 113777 015756 136176                MOVB  TRIBN,@SEL3     ;SET UP TRIBN
23 064662 017777 134134 136172                MOV   @TSSE,@SEL4    ;MOVE DATA INTO SEL4
24 064670 113777 021024 136170                MOVB  TSSA,@SEL6     ;SET UP POLL PARAMETER
25 064676 112777 000001 136152                MOVB  #01,@SEL2      ;DO CONTROL IN WRITE TSS/GSS
26 064704 022737 000237 021024                CMP   #237,TSSA
27 064712 001406                BEQ   WRIP1EX        ;EXIT IF DONE
28 064714 005237 021024                INC   TSSA
29 064720 062737 000002 021022                ADD   #2,TSSSE
30 064726 000745                BR    WRIP1          ;GO BACK FOR MORE
31 064730 000207                WRIP1EX: RTS        PC
32
    
```

CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 105  
 WRITE MODEM CONTROL

```

1      .SBTTL          WRITE MODFM CONTROL
2      : **
3      : FUNCTIONAL DESCRIPTION:          WRMCS  WRITE MODEM CONTROL SIGNALS
4      :
5      :           WIRTE MODEM CONTROL SIGNALS FROM TEMP TO DMP
6      :           THIS ROUTINE IS IGNORED BY THE DMV
7      :
8      : INPUTS:
9      :           TEMP  CONTAINS CONTENTS FRO BSEL4
10     :
11     : CALLING SEQUENCE:
12     :           JSR   PC,WRMCS          ;WRITE MODEM CONTROL
13     :
14
15 064732 152777 000200 136112 WRMCS:  BISB   @RQI,@BSELO      ;DO REQUEST IN
16 064740 004737 065114          JSR   PC,TOORIO      ;WAIT TIL PORT IS OURS
17 064744 113777 015756 136106          MOVB  TRIBN,@BSEL3    ;SET UP TRIBN
18 064752 013777 017350 136102          MOV   TEMP,@SEL4
19 064760 012777 000021 136100          MOV   @21,@SEL6      ;DO WRITE MODEM
20 064766 112777 000001 136062          MOVB  @01,@SEL2      ;CONTROL IN
21 064774 000207          RTS   PC          ;THEN RETURN TO CALLER
22
23
24     .SBTTL          HALT TRIB SUBROUTINE
25     : **
26     : FUNCTIONAL DESCRIPTION:
27     :           HLTTRB - HALT TRIB SUBROUTINE HALTS ALL TRIBS THAT
28     :           ARE FOUND IN THE TRIBLSIT
29     :
30     : INPUTS:          TRIBLS - CONTAINS VALID TRIBS
31     :
32     : SUBORDINATE ROUTINES USED:
33     :
34     :           TOORIO  TIME OUT OR INPUT OR OUTPUTN INTERRUPT
35     :
36     : CALLING SEQUENCE:
37     :           JSR   PC,HLTTRB
38     :
39
40 064776 022737 000001 023102 HLTTRB: CMP   @1,DEVPAR      ;IS THIS TRIB OR CONTROL
41 065004 001442          BEQ   HLTREX          ;BRANCH IF TRIB
42 065006 032737 000002 017410          BIT   @DATCKB,PARAM
43 065014 001006          BNE   HLTTR2          ;IF CHECK GO TO 2
44 065016 012737 000002 017462          MOV   @2,TIMERS      ;SET UP FOR 2 SEC TIMER
45 065024 005737 017462          HLTTR3: TST  TIMERS
46 065030 001375          BNE   HLTTR3          ;WAIT FOR TIMER TO BE 0
47 065032
48 065032 012737 177777 015762 HLTTR2: MOV   @1,INDEX      ;MAKE INDEX = 1
49 065040 004737 046462          HLTTR1: JSR   PC,GIVIND    ;GET VALID INDEX
50 065044 022737 000040 015762          CMP   @32,,INDEX     ;DONE
51 065052 001417          BEQ   HLTREX          ;IF SO EXIT
52
53          ;HALT TRIB
54
55 065054 152777 000200 135770          BISB  @RQI,@BSELO      ;DO REQUEST IN
56 065062 004737 065114          JSR   PC,TOORIO      ;WAIT TIL PORT IS OURS
57 065066 113777 015756 135764          MOVB  TRIBN,@BSEL3    ;SET UP TRIB NO.
  
```

H16

SEQ 202

CZCLMCO GMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 105 1  
HALT TRIB SUBROUTINE

58	065074	012777	000005	135764	MOV	#05,@SEL6	;HALT TRIB
59	065102	112777	000001	135746	MOVB	#01,@SEL2	;CLEAR ROI AND DO COMMAND.
60	065110	000753			BR	HLTIR1	;GO BACK AND GET ANOTHER
61	065112	000207			HLTIREX: RTS	PC	;RETURN TO CALLER
62							

```

1      .SBTTL                TIME OUT OR INPUT INT. OR OUTPUT INT.
2
3
4
5      ;**
6      ; FUNCTIONAL DESCRIPTION:
7      ;   TOORIO - TIME OUT OR INPUT INTERRUPT OR OUTPUT INTERRUPT
8      ;   THIS ROUTINE SETS UP A TIMER FOR 100 (OCTAL) TICKS
9      ;   THEN CHECKS FOR TIME OUT, OR INPUT INTERRUPT, OR OUTPUT
10     ;   INTERRUPT. IF TIME OUT OCCURS IT REPORTS ERROR AND
11     ;   RESTARTS TIMER. IF INPUT INTERRUPT OCCURS RETURN TO CALLER
12     ;   IF OUTPUT INTERRUPT OCCURS LOG IT AND CONTINUE WAITING FOR
13     ;   INPUT INTERRUPT.
14     ;
15     ; USE OF FLAGS:
16     ;   "OTINT" - SET BY OUTPUT INT ROUTINE
17     ;   "ININT" - SET BY INPUT INT. ROUTINE
18     ;               CLEARED BY THIS ROUTINE.
19     ;
20     ; SUBORDINATE ROUTINES USED:
21     ;
22     ;   "OUTHDL" - OUTPUT INTERRUPT HANDLER
23     ;
24     ; CALLING SEQUENCE:
25     ;   JSR      PC,TOORIO
26     ;--
27 065114 011637 017374      TOORIO: MOV      (SP),PCADD      ;SAVE ADDR. OF CALLING ROUTINE
28 065120 012737 000100 017456      MOV      #100,TIMER1    ;SET UP TIMER
29 065126 032737 000003 023100      BIT      #3,OPTYP      ;IS THIS DMV
30 065134 001403              BEQ      TOOR3          ;BRANCH IF NOT
31 065136 012737 000400 017456      MOV      #400,TIMER1   ;MAKE TIME OUT GREATER IF DMV
32 065144 005737 017456      TOOR3: TST      TIMER1    ;IS TIME EXPIRED
33 065150 001022              BNE      TOOR1          ;IF NOT CONTINUE
34                                ;IF YES ERROR

```

35	065152	012737	037354	017354	MOV	#DVEM3,TEMP2		
36	065160	017737	135672	017360	MOV	@SEL2,TEMP4		
37	065166	017737	135660	017356	MOV	@SELO,TEMP3		
38	065174	004737	042104		JSR	PC,LGDVE		
39	065200	005237	017310		INC	ERRCNT		
40	065204				ERRSOFT	7,DVEM3,ERR13		
	065204	104457					TRAP	C\$ERSOFT
	065206	000007					.WORD	7
	065210	037354					.WORD	DVEM3
	065212	041460					.WORD	ERR13
41	065214	00J737			BR	TOOR10		
42								
43	065216				TOOR1:	BREAK		
	065216	104422					TRAP	C\$BRK
44	065220	032737	000002	017414	BIT	#OTINT,FLAG		
45								
46	065226	001402			BEG	TOOR2		
47								
48	065230	004737	065254		JSR	PC,OUTHDL		
49	065234	032737	000001	017414	TOOR2:	#ININT,FLAG		
50	065242	001740			BEG	TOOR3		
51	065244	042737	000001	017414	BIC	#ININT,FLAG		
52	065252	000207			RTS	PC		
53								

```

1      .SBTTL                OUTPUT INTERRUPT HANDLER
2
3      ;**
4      ; FUNCTIONAL DESCRIPTION:
5      ;   OUTHDL - OUTPUT INTERRUPT HANDLER
6      ;   THIS ROUTINE IS CALLED WHEN AN OUTPUT INTERRUPT HAS SET
7      ;   THE "OTINT" BIT IN THE "FLAG" WORD. IT CHECKS FOR
8      ;   AN RDO SIGNAL IF NO RDO THEN REPORT ILLEGAL INTERRUPT.
9      ;   THEN IT CHECKS FOR BACC OUT IF NOT BACC OUT REPORT THE
10     ;   TYPE OF OUTPUT ERROR. IF BACC OUT FIND IF RX OR TX
11     ;   IF RX SET CRX BIT AND MOVE ADDR AND BYTE COUNT TO RSEL4
12     ;   AND RSEL6. IF TX SET CTXV BIT AND MOVE ADDR AND BYTE COUNT
13     ;   TO TSEL4 AND TSEL6. CLEAR OTINT FLAG AND RETURN TO CALLER.
14
15     ; USE OF FLAGS:
16     ;   "OTINT" - SET BY OUPUT ROUTINE
17     ;               CLEARED BY THIS ROUTINE
18     ;   "DMRRUN" - SET BY DVINIT ROUTINE IF THIS IS DMR
19     ;               CHECKED AND CLEARED BY THIS ROUTINE.
20     ;   "CTX"    - SET IF TRANSMIT COMPLETED
21     ;   "CRX"    - SET IF RECEIVE COMPLETED
22
23     ; SUBORDINATE ROUTINES USED:
24     ;   "LGDVE" -LOG DEVICE ERRORS TO EVENT LOG
25
26     ; CALLING SEQUENCE
27     ;   JSR      PC,OUTHDL
28
29     ;-
30
31
32
33     OUTHDL: MOV      (SP),PCADD      ;SAVE ADDR. OF CALLING ROUTINE
34     065254 011637 017374
35     065260 042737 000002 017414
36     065266 005737 017320
37     065272 001404
38     065274 142777 000200 135554
39     065302 000207
40     065304
41     065304 017703 135546
42     065310 042703 177770
43     065314 022703 000001
44     065320 001405
45     065322 022703 000002
46     065326 001550
47     065330 000137 066274
48
49     ;CONTROL OUT HANDLER
50     CONOHD:
51     065334
52     065334 005003
53     065336 157703 135524
54
55     ; REV C EC
56     ;; IF MODEM DISCONNECT OR RING IS DETECTED, AND DCLT IS NOT RUNNING.
57     ;; WE INGNORE IT.

```

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22-Mar-84 16:24 Page 107 1  
 OUTPUT INTERRUPT HANDLER

```

58 065342 005737 017416      TST      RUNING      ; DCLT RUNNING?
59 065346 001011              BNE      CON01E     ; YES,BRANCH
60 065350 022703 000304      CMP      #304,R3    ; MODEM DISCONNECT(DSR DROPPED)?
61 065354 001404              BEQ      CON01F     ; YES, EXIT
62 065356 022703 000032      CMP      #32,R3    ; MODEM RING ?
63 065362 001401              BEQ      CON01F     ; YES,BRANCH
64 065364 000402              BR       CON01E     ; GO HANDLE INTERRUPT
65 065366              CON01F:
66 065366 000137 066452      JMP      OUTHEX     ; EXIT
67
68 065372 032703 000100      CON01E: BIT      #BIT6,R3 ; IS THIS ERROR IN THE 100 176 RANGE
69                                ;OR THE 300-376 RANGE
70 065376 001052              BNE      CON01      ;BRANCH IF YES.
71 065400 022703 000024      CMP      #24,R3    ;IS THIS A RUN STATE
72 065404 001011              BNE      CON01B     ;IF NOT GO TO 1B
73 065406 032737 000400 017414  BIT      #RUNST,FLAG ;TEST THE RUN STATE
74 065414 001437              BEQ      CON01A     ;IF NOT SET GO TO 1A
75 065416 012737 177777 017416  MOV      #-1,RUNING ; SET "DCLT RUNNING" FLAG
76 065424 000137 066452      JMP      OUTHEX
77 065430 022703 000002      CON01B: CMP      #2,R3    ;IS IT RX THRESH
78 065434 001004              BNE      CON01C     ;BRANCH IF NOT
79 065436 005237 017302      INC      OPVAR     ;BUMP OPVAR
80 065442 000137 066452      JMP      OUTHEX     ;AND EXIT
81 065446 022703 000004      CON01C: CMP      #4,R3    ;IS IT TX THRESH
82 065452 001004              BNE      CON01D     ;BRANCH IF NOT
83 065454 005237 017304      INC      OPVAR1    ;IN TX COUNT
84 065460 000137 066452      JMP      OUTHEX     ;AND EXIT ROUTINE
85 065464 022703 000006      CON01D: CMP      #6,R3    ;IS IT SELECT
86 065470 001411              BEQ      CON01A     ;BRANCH IF SO
87 065472 022703 000032      CMP      #32,R3    ;IS IT RING D
88 065476 001406              BEQ      CON01A
89 065500 022703 000022      CMP      #22,R3    ;IS IT DEAD TRIB
90 065504 001403              BEQ      CON01A     ;BRANCH IF SO
91 065506 012737 177777 017324  MOV      #-1,FTLFLG ;SET FATAL ERROR FLAG
92 065514 016337 023106 017366  CON01A: MOV      CON0LS(R3),CONOTM
93 065522 000427              BR       CON04      ;THEN GO TO 4
94
95
96 065524 012737 177777 017324  CON01:  MOV      #-1,FTLFLG ;SET FATAL ERROR FLAG
97 065532 032703 000200      BIT      #BIT7,R3  ;IS THIS 300 RANGE
98 065536 001006              BNE      CON03      ;IF SO GO TO 3
99 065540 042703 000100      BIC      #BIT6,R3  ;CLEAR TOP BIT
100 065544 016337 023142 017366  MOV      CON01S(R3),CONOTM
101 065552 000413              BR       CON04      ;LOAD UP MSG AND GO TO 4
102
103 065554 022703 000306      CON03:  CMP      #306,R3 ;IS THIS QUE OVER FLOW
104 065560 001003              BNE      CON03A
105 065562 012737 177777 017330  MOV      #-1,OVRcnt
106 065570 042703 000300      CON03A: BIC      #BIT7!BIT6,R3 ;CLEAR THE TOP BITS
107 065574 016337 023210 017366  MOV      CON03S(R3),CONOTM
108
109 065602 017737 135260 017360  CON04:  MOV      @SEL6,TEMP4
110 065610 017737 135242 017356  MOV      @SEL2,TEMP3
111 065616 012737 037430 017354  MOV      @DVEM4,TEMP2
112 065624 004737 042104      JSR      PC,LGDVE  ;GO LOG ERROR
113 065630 005237 017310      INC      ERRcnt
114 065634              ERRSOF T 7,DVEM4,ERR14

```



```

065634 104457
065636 000007
065640 037430
065642 041512
115 065644 000137 066452 JMP OUTHEX ;EXIT OUTPUT HANDLER
116
117 ;INFORMATION OUT HANDLER
118 ;BA AND CC HANDLER
119
120
121
122 065650 INFOHD:
123 065650 122777 000010 135210 CMPB #10, @SEL6 ;IS THIS A MODEM STATUS
124 065656 001005 BNE INFOH1 ;GO TO INFO 1 IF NOT MODEM STATUS
125 065660 017737 135176 020524 MOV @SEL4, MODS ;PUT IN NEW MOD STATUS
126 065666 000137 066452 INFO1B: JMP OUTHEX
127 065672 032777 000040 135166 INFOH1: BIT @BITS, @SEL6 ;
128 065700 001011 BNE INFOHA ;BRANCH IF RD/TSS
129 065702 022777 000020 135156 CMP #20, @SEL6
130 065710 001766 BEQ INFO1B ;GET OUT IF BUFF RET CMP
131 065712 012737 041212 017366 MOV @INFOM, CONOTM ;SET UP FOR INFO ERROR
132 065720 000137 065602 JMP CONO4 ;AND PRINT IT
133 065724 005037 017326 INFOHA: CLR TSSFLG ;CLEAR FLAG
134 065730 017704 135126 MOV @SEL4, R4
135 065734 017703 135126 MOV @SEL6, R3
136 065740 042703 177740 BIC #177740, R3 ;CLEAR ALL BUT LAST 5 BITS
137 065744 105777 135110 TSTB @SEL3 ;IS THIS GSS
138 065750 001007 BNE INFOH8 ;BRANCH IF NOT
139
140
141 065752 116302 021130 MOVB GSSIND(R3), R2
142 065756 006303 ASL R3 ;USE WORD INDEX
143 065760 016337 021030 017366 MOV GSSLST(R3), CONOTM ;USE GSS LIST
144 065766 000406 BR INFOH2
145
146 065770 116302 020762 INFOH8: MOVB TSSIND(R3), R2
147 065774 006303 ASL R3
148 065776 016337 020662 017366 MOV TSSLST(R3), CONOTM ;IF TSS USE THAT LIST
149
150 066004 000172 066010 INFOH2: JMP @INFOH4(R2)
151
152 066010 066016 INFOH4: .WORD INFOH5 ;WORD ROUTINE
153 066012 066044 .WORD INFOH6 ;BYTE ROUTINE
154 066014 066110 .WORD INFOH7 ;SPECIAL ROUTINE
155 066016 010446 INFOH5: PRINTS CONOTM, R4
066020 013746 017366 MOV R4, (SP)
066024 012746 000002 MOV CONOTM, (SP)
066030 010600 MOV #2, -(SP)
066032 104416 MOV SP, R0
066034 062706 000006 TRAP C#PNTS
156 066040 000137 066452 ADD #6, SP
157 066044 010437 017350 INFOH6: JMP OUTHEX
158 066050 066050 PRINTS CONOTM, <B, TEMP>, <B, TEMP+1>
066050 005046 CLR -(SP)
066052 153716 017351 BISB TEMP+1, (SP)
066056 005046 CLR -(SP)
    
```

066060	153716	01735C						BISB	TEMP,(SP)
066064	013746	017366						MOV	CONOTH,(SP)
066070	012746	000003						MOV	#3,(SP)
066074	010600							MOV	SP,RO
066076	104416							TRAP	C#PNTS
066100	062706	000010						ADD	#10,SP
159	066104	000137	066452						
160	066110	010437	017350	INFOH7:	JMP	OUTHEX			
161	066114	005037	017352		MOV	R4,TEMP			
162	066120	005037	017354		CLR	TEMP1			
163	066124	005037	017356		CLR	TEMP2			
164	066130	005037	017360		CLR	TEMP3			
165	066134	032737	000400	017350	CLR	TEMP4			
166	066142	001402			BIT	#BIT8,TEMP			
167	066144	005237	017352		BEQ	1#			
168	066150	032737	001000	017350	INC	TEMP1			
169	066156	001402		1#:	BIT	#BIT9,TEMP			
170	066160	005237	017354		BEQ	2#			
171	066164	032737	002000	017350	INC	TEMP2			
172	066172	001402		2#:	BIT	#BIT10,TEMP			
173	066174	005237	017356		BEQ	4#			
174	066200	032737	0040C0	017350	INC	TEMP3			
175	066206	001402		4#:	BIT	#BIT11,TEMP			
176	066210	005237	017360		BEQ	3#			
177	066214			3#:	INC	TEMP4			
066214	005046				PRINTS	CONOTH,<B,TEMP>,<B,TEMP1>,<B,TEMP2>,<B,TEMP3>,<B,TEMP4>			
066216	153716	017360			CLR	-(SP)			
066222	005046				BISB	TEMP4,(SP)			
066224	153716	017356			CLR	-(SP)			
066230	005046				BISB	TEMP3,(SP)			
066232	153716	017354			CLR	-(SP)			
066236	005046				BISB	TEMP2,(SP)			
066240	153716	017352			CLR	(SP)			
066244	005046				BISB	TEMP1,(SP)			
066246	153716	017350			CLR	-(SP)			
066252	013746	017366			BISB	TEMP,(SP)			
066256	012746	000006			MOV	CONOTH,(SP)			
066262	010600				MOV	#6,( )			
066264	104416				MOV	SP,RO			
066266	062706	000016			TRAP	C#PNTS			
178	066272	000467			ADD	#16,SP			
179	066274			BACCHD:	BR	OUTHEX			
180	066274	032703	000004		BIT	#BIT2,R3			
181	066300	001035			BNE	BACCTX			
182	066302	022737	000022	015766	CMP	#18.,CRX			
183	066310	001001			BNE	1#			
184	066312				DOCLN				
185	066314	005237	015766	1#:	TRAP	C#DCLN			
186					INC	CRX			
187	066320	013702	015772		MOV	RSPTRE,R2			
188	066324	017722	1345#6		MOV	#SEL6,(R2).			
189	066330	017722	1345#6		MOV	#SEL4,(R2).			
190	066334	017722	134516		MOV	#SEL2,(R2).			
191	066340	022702	0161#6		CMP	#RXSKEN,R2			
192	066344	001002			BNE	2#			
193	066346	012702	016012		MOV	#RXSTAK,R2			

IF BIT IS SET GO DO TX  
IS THIS GOING TO BREAK THE BANK?

LOAD R2 WITH POINTER

CZCLMCO DMP V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Pz 107 4  
 OUTPUT INTERRUPT HANDLER

```

194 066352 010237 015772      28:  MOV      R2,RSPTRE
195 066356 005737 017330      38:  TST      OVRCNT
196 066362 001433              BEQ      OUTHEX
197 066364              ERRSOFT 12,DVEM7,ERR15
      066364 104457              TRAP    C$ERSOFT
      066366 000014              .WORD  12
      066370 037645              .WORD  DVEM7
      066372 041550              .WORD  ERR15

198
199
200 066374 005237 015764      BACCTX: INC      CTX          ;INC TX COMPLETE
201 066400 013702 015774          MOV      TSPTR,R2      ;LOAD R2 WITH POINT
202 066404 017722 134456          MOV      @SEL6,(R2).
203 066410 017722 134446          MOV      @SEL4,(R2).
204 066414 017722 134436          MOV      @SEL2,(R2).
205 066420 010237 015774          MOV      R2,TSPTR
206 066424 022702 016012          CMP      @RXSTAK,R2
207 066430 001001          BNE     18
208 066432              DOCLN          ;BAD NEWS
      066432 104444              TRAP    C$DCLN

209
210 066434 005737 017330      18:  TST      OVRCNT      ;CHECK IF HERE FROM QUE OVER
211 066440 001404          BEQ      OUTHEX
212 066442              ERRSOFT 13,DVEM7,ERR16
      066442 104457              TRAP    C$ERSOFT
      066444 000015              .WORD  13
      066446 037645              .WORD  DVEM7
      066450 041610              .WORD  ERR16

213
214 066452 142777 000200 134376  OUTHEX: BICB     @RDO,@SEL2  ;CLEAR RDO
215 066460 005737 017330          TST     OVRCNT      ;TEST THE OVER FLOW COUNT
216 066464 001427          BEQ     OUTHE3      ;BRANCH IF ZERO
217 066466              OUTHE4: BREAK
      066466 104422              TRAP    C$BRK
218 066470 032737 000002 017414          BIT     @OTINT,FLAG  ;IS OUTPUT INTERRUPT SET
219 066476 001402          BEQ     OUTHE5      ;BRANCH IF NOT
220 066500 000137 065254          JMP     OUTMDL      ;WHEN SET GO BACK FOR NEXT ON QUE
221 066504 032737 000001 017414  OUTHE5: BIT     @ININT,FLAG ;TEST FOR INPUT INT
222 066512 001414          BEQ     OUTHE3      ;BRANCH IF NOT INPUT
223 066514 042737 000001 017414          BIC     @ININT,FLAG
224 066522 105077 134340          CLRB   @SEL6
225 066526 112777 000001 134322          MOVB   @01,@SEL2    ;DO NO REQUEST
226 066534 012737 177777 017322          MOV     @1,RQIFLG   ;SET RQI FLAG
227 066542 000751          BR     OUTHE4

228
229          ; SEE IF FATAL ERROR HAS OCCURRED
230
231
232 066544 005737 017324      OUTHE3: TST     FTLFLG
233 066550 001416          BEQ     OUTHE6      ;BRANCH IF NOT FATAL
234 066552 005037 017324          CLR     FTLFLG      ;CLEAR FATAL FLAG

235
236          ; IF A FATAL ERROR HAS OCCURRED, WE CLEAR ALL MODEM SIGNALS
237
238
239 066556 105037 015756          CLRB   TRIBN        ;PARAMETER FOR SUBROUTINE
240 066562 005037 017350          CLR     TEMP        ;CLEAR ALL MODEM SIGNALS
    
```

```

241 066566 004737 064732      JSR    PC,WRMCS      ; GO CLEAR MODEM SIGNALS
242 066572 005037 017416      CLR    RUNNING      ; INIT "DCLT RUNNING" FLAG
243
244
245          ; RESTORE ORIGINAL STACK POINTER AND GO TO "DCLT>" PROMPT.
246          ;
247
248 066576 013706 017364      MOV    SAVSP,SP     ; RESET STACK
249 066602 000137 052402      J*P    GTRAS        ; GO BACK TO DCLT>
250
251
252 066606 005737 017322      OUTHE6: TST    RQIFLG
253 066612 001405              BEQ    OUTHE2
254 066614 005037 017322      CLR    RQIFLG      ; CLEAR THE RQI FLAG.
255 066620 152777 000200 134224 BISB   #RQI,08SELO
256 066626 005737 017326      OUTHE2: TST    TSSFLG ; TEST THE TSSFLG
257 066632 001315              BNE    OUTHE4      ; IF NOT ZERO WAIT TIL IT IS.
258 066634 005037 017330      CLR    OVRCNT      ; CLEAR THE OVERFLOW FLAG
259 066640 000207              RTS    PC           ; RETURN TO CALLER
260
261 066642 000000              TSEL6: .WORD    0
262 066644 000000              TSEL4: .WORD    0
263 066646 000000              TSEL3: .WORD    0
264 066650 000000              RSEL4: .WORD    0      ; TEMP STORAGE LOCS.
265 066652 000000              RSEL6: .WORD    0
266 066654 000000              RSEL3: .WORD    0
267
    
```

```

9
10
11
21
22 066656          BGNSRV  DVINS          DVINS::
   066656
28 066656 052737 000001 017414  BIS    #ININT,FLAG
29 066664 042777 000200 134160  BIC    #BIT7,#BSELO ;CLEAR RQI
30
31 066672          ENDSRV
   066672          L10021:
   066672 000002          RTI

```

32

42 066674  
066674

BGNSRV DVOUTS

DVOUTS::

43

49 066674 052737 000002 017414  
50 066702  
066702  
066702 000002

BIS #OTINT,FLAG  
ENDSRV

L10022: RTI

51

52

53

12  
13  
14  
15 066704  
066704  
066704 104401  
16  
1  
2

.EVEN  
ENDTST

L10020: TRAP C#ETST

.SBTTL HARDWARE PARAMETER CODING SECTION

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

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12

13 066706  
066706 000034  
066710

BGNHRD

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

14  
24  
25  
26

.SBTTL DEVICE INDEPENDENT SECTION

27 066710  
066710 000130  
066712 067000  
066714 000001

GPRML DPLX,0,1,YES

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

28  
29  
44

.SBTTL DEVICE DEPENDENT SECTION

45  
46  
47

48 066716  
066716 001031  
066720 067031  
066722 160000  
066724 177776

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

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

49 066726  
066726 002031  
066730 067056  
066732 000300  
066734 000776

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

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

50 066736  
066736 003032  
066740 067111  
066742 000340  
066744 000004  
066746 000007

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

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

51 066750  
066750 005032  
066752 067226  
066754 000007  
066756 000000  
066760 000004

GPRMD OPTYPM,12,0,7,0,4,YES

.WORD T\$CODE  
.WORD OPTYPM  
.WORD 7  
.WORD T\$LLOLIM  
.WORD T\$HILIM

52 066762  
066762 004130  
066764 067137  
066766 000001

GPRML PTPMLP,10,1,YES

.WORD T\$CODE  
.WORD PTPMLF  
.WORD 1

53 066770  
066770 004044

XFERF ENDHWL

.WORD T\$CODE



```

54 066772          GPRML  TRIBCO,10.2,YES          .WORD  T$CODE
    066772 004130          .WORD  TRIBCO
    066774 067173          .WORD  2
    066776 000002
55 067000          ENDMWL:
56 067000          ENDMRD
                                L10023: .EVEN
    067000
57
58          .MLIST  BEX
59
60          ;DEVICE INDEPENDENT QUESTIONS
61
62 067000          106      125      114  DPLX:  .ASCIZ  /FULL DUPLEX OPERATION : /
63
64
65          ;DEVICE DEPENDENT QUESTION
66
67 067031          104      105      126  CSRADR: .ASCIZ  /DEVICE CSR ADDRESS: /
68 067056          111      116      124  VECTOR: .ASCIZ  /INTERRUPT VECTOR ADDRESS: /
69 067111          111      116      124  PRIOR:  .ASCIZ  /INTERRUPT PRIORITY : /
70 067137          111      123      040  PTPMLP: .ASCIZ  /IS THIS MULTIPOINT NETWORK:/
71 067173          111      123      040  TRIBCO: .ASCIZ  /IS THIS A CONTROL STATION:/
72 067226          117      120      124  OPTYPM: .ASCII  /OPTION TYPE /<15><12>
73 067244          040      060      075  .ASCII  / 0=DMP/<15><12>/ 1=DMV:/
74
75          .LIST  BEX
76
77          .EVEN
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
22  
23  
24  
25  
26  
33  
34  
35  
36  
37  
38  
39  
40  
47  
48  
49  
50  
51

;.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  
.....

\$PATCH: .BLKW 30

LASTAD

L\$LAST:: ENDMOD

.END

067344 000000  
067346 000000

067350  
067350

000001

.EVEN  
.WORD 0  
.WORD 0

ACT 000003  
 ACTATV 055026  
 ACTBCR 054632  
 ACTCHK 055242  
 ACTCKT 057010  
 ACTCLB 054154  
 ACTCLP 055354  
 ACTCLR 053644  
 ACTCOP 054452  
 ACTCRC 055256  
 ACTCSE 054000  
 ACTCST 054072  
 ACTDLL 055074  
 ACTDME 054400  
 ACTDMQ 054372  
 ACTDMS 054350  
 ACTDMX 054406  
 ACTECH 055152  
 ACTEKE 056622  
 ACTEKT 056404  
 ACTEQO 054574  
 ACTETB 055452  
 ACTEWS 055554  
 ACTEXT 053730  
 ACTEXX 057006  
 ACTESA 056610  
 ACTESB 056604  
 ACTHLP 053664  
 ACTKAL 055472  
 ACTKTB 055462  
 ACTLIS 055064  
 ACTLLP 055364  
 ACTLPX 055402  
 ACTLXX 055444  
 ACTMEX 055020  
 ACTME1 054754  
 ACTMOP 055334  
 ACTMOS 055264  
 ACTMS0 054654  
 ACTMS1 054662  
 ACTMS2 054672  
 ACTMS3 054702  
 ACTMS4 054712  
 ACTMS5 054722  
 ACTMS6 054740  
 ACTM2X 055122  
 ACTNO 055142  
 ACTNUF 053634  
 ACTNUL 053642  
 ACTNUM 054462  
 ACTOPM 054554  
 ACTPAS 055036  
 ACTPRO 055272  
 ACTPRT 053740  
 ACTQFG 055276  
 ACTREC 055056  
 ACTREX 044500

ACTRGS 044524  
 ACTRHL 044434  
 ACTRLG 044510  
 ACTRLP 055374  
 ACTRNF 044424  
 ACTRNL 044432  
 ACTRPS 055324  
 ACTRSE 044546  
 ACTRSF 044672  
 ACTRSO 044706  
 ACTRTN 044600  
 ACTRTS 044554  
 ACTRUN 053754  
 ACTSEX 054764  
 ACTSHO 053654  
 ACTSHW 054214  
 ACTSLS 056150  
 ACTSTE 054414  
 ACTSTS 055250  
 ACTSTT 054424  
 ACTSTX 054432  
 ACTSZE 054442  
 ACTTAL 055114  
 ACTTLP 055344  
 ACTTRA 055104  
 ACTWS1 055732  
 ACTWS2 055736  
 ACTWS3 056036  
 ACTWS5 055666  
 ACTWS7 056002  
 ACTWS9 055570  
 ACTW7A 056020  
 ACTW7B 056026  
 ADDCC 045350  
 ADDC1 045444  
 ADR 000020 G  
 ALCK 060150  
 ALCK1 060234  
 ALCK2 060310  
 ALCK2A 060550  
 ALCK3 060610  
 ALCK3A 060732  
 ALCK3B 060754  
 ALCK3C 060744  
 ALCK3D 060714  
 ALCK4 060776  
 ALCK4A 061010  
 ALCK5 060210  
 ALCK5B 060220  
 ALLTR 060210  
 ASSEMB 000010  
 ATVMOD 000027  
 BABTM 040275  
 BACCHD 066274  
 BACCTX 066374  
 BAD 017371  
 BADCHR 000051

BASM1 030401  
 BASM2 030372  
 BASM3 030363  
 BDCLK 027232  
 BIT0 000001 G  
 BIT00 000001 G  
 BIT01 000002 G  
 BIT02 000004 G  
 BIT03 000010 G  
 BIT04 000020 G  
 BIT05 000040 G  
 BIT06 000100 G  
 BIT07 000200 G  
 BIT08 000400 G  
 BIT09 001000 G  
 BIT1 000002 G  
 BIT10 002000 G  
 BIT11 004000 G  
 BIT12 010000 G  
 BIT13 020000 G  
 BIT14 040000 G  
 BIT15 100000 G  
 BIT2 000004 G  
 BIT3 000010 G  
 BIT4 000020 G  
 BIT5 000040 G  
 BIT6 000100 G  
 BIT7 000200 G  
 BIT8 000400 G  
 BIT9 001000 G  
 BLDBEX 045556  
 BLDBUF 045446  
 BLDB1 045456  
 BLDB2 045520  
 BLDB3 045536  
 BOE 000400 G  
 BSEL0 023052  
 BSEL1 023054  
 BSEL2 023056  
 BSEL3 023060  
 BSEL4 023062  
 BSEL5 023064  
 BSEL6 023066  
 BSEL7 023070  
 BUFEX 027377  
 BUFLIM 001000  
 BUFTSM 041111  
 BYTBIT 017316  
 CABLE 000002  
 CARLOS 041175  
 CBLLOP 000045  
 CCURAD 017246  
 CHECK 000003  
 CLEAR 000001  
 CLIACT 053436  
 CLIALN 000007  
 CLIALP 000006

CLIBCR 023604  
 CLIBDL 023477  
 CLIBIF 000003  
 CLIBR 000002  
 CLIBRX 023457  
 CLIDEC 000011  
 CLIERM 023362  
 CLIERR 000000  
 CLIEXI 000001  
 CLINBG 023435  
 CLINPS 023541  
 CLINUF 023412  
 CLINUM 000005  
 CLIOCT 000010  
 CLIPPE 023702  
 CLIPW 023750  
 CLIRAC 044360  
 CLIRT 044774  
 CLISE0 023651  
 CLISPA 000004  
 CLISTR 000012  
 CLITRE 021170  
 CLI\$PM 023354  
 CLI\$RP 025334  
 CLKBR 017440  
 CLKCSR 017436  
 CLKEN 017446  
 CLKHZ 017444  
 CLKINT 041700 G  
 CLKSET 041654  
 CLKVEC 017442  
 CLNSET 017320  
 CLRPEX 046362  
 CLRPLS 046340  
 CLRPL1 046352  
 CMDBUF 003130  
 CMPBUF 004416  
 CMPNEW 061040  
 CMPPTR 017240  
 CMPSEX 061322  
 CMPSR 061022  
 CMPS1 061202  
 CMPS2 061206  
 CMPS3 061074  
 CMPS5 061304  
 CMPS5A 061310  
 CMPS6 061252  
 CMPS7 061170  
 CMPTOT 017242  
 MSG0 000020  
 MSG1 000021  
 MSG2 000022  
 MSG3 000023  
 MSG4 000024  
 MSG5 000025  
 MSG6 000026  
 CONOHD 065334

CONOLS 023106  
 CONOTM 017366  
 CONO1 065524  
 CONO1A 065514  
 CONO1B 065430  
 CONO1C 065446  
 CONO1D 065464  
 CONO1E 065372  
 CONO1F 065366  
 CONO1S 023142  
 CONO3S 023210  
 CONO4 065602  
 CONO3 065554  
 CONO3A 065570  
 CPTR 017340  
 CPTRLS 015412  
 CPTRR 017336  
 CPTTLS 015512  
 CR 031566  
 CRC 000040  
 CRFB 000020  
 CRX 015766  
 CSHEXP 000006  
 CSHTRN 000007  
 CSRADR 067031  
 CTOTCC 017244  
 CTPP 000064  
 CTS 000004  
 CTX 015764  
 CURADD 017342  
 CURCC 017334  
 C\$AU 000052  
 C\$AUTO 000061  
 C\$BRK 000022  
 C\$BSEG 000004  
 C\$BSUB 000002  
 C\$CEFG 000045  
 C\$CLCK 000062  
 C\$CLEA 000012  
 C\$CLOS 000035  
 C\$CLP1 000006  
 C\$CVEC 000036  
 C\$DCLN 000044  
 C\$DDU 000051  
 C\$DRPT 000024  
 C\$DU 000053  
 C\$EDIT 000003  
 C\$ERDF 000055  
 C\$ERHF 000056  
 C\$ERRO 000060  
 C\$ERSF 000054  
 C\$ERSO 000057  
 C\$ESCA 000010  
 C\$ESEG 000005  
 C\$ESUB 000003  
 C\$ETST 000001  
 C\$EXIT 000032

C\$GETB= 000026	DLLE7 062004	DVEM5 037514	EFM2 027627	EVT3F 030717
C\$GETW= 000027	DLLE7A 062020	DVEM6 037571	EF.CON= 000036 G	EVT4 031124
C\$GMAN= 000043	DLLE8 062124	DVEM7 037645	EF.NEW= 000035 G	EVT4A 031226
C\$GPHR= 000042	DLLGA = 001000	DVEM8 037725	EF.PWR= 000034 G	EVT4B 031016
C\$GPLO= 000030	DLLIND 023222	DVEST 063554	EF.RES= 000037 G	EVT44 031055
C\$GPRI= 000040	DLLMOD= 000033	DVES1 063540	EF.STA= 000040 G	EVT5A 031324
C\$INIT= 000011	DLLM1 002647	DVES1A 063454	EKT8 = 000063	EVT6 030756
C\$INLP= 000020	DLLM1C 002172	DVI = 000012	EMSG0 002221	EVTLOG 017466
C\$MANI= 000050	DLLM1E 002654	DVINEX 063746	EMSG1 002222	EVTLST 020562
C\$MEM = 000031	DLLM2 002654	DVINI? 062752	EMSG2 002223	EVTMIN 020612
C\$MSG = 000023	DLLM2C 002174	DVINS 066656 G	EMSG3 002224	EVTPTR 017464
C\$OPEN= 000034	DLLM2E 003130	DVIN1 063172	EMSG4 002324	EVTSLC 020610
C\$PNTB= 000014	DLLPRI 061602	DVIN11 063326	EMSG5 002416	EVTTC 020614
C\$PNTF= 000017	DLLQ1 025210	DVIN12 063334	EMSG6 002517	EVTTMP 020622
C\$PNTS= 000016	DLM 032014	DVIN13 063246	EMSG8 002647	EXIT = 000066
C\$PNTX= 000015	DLTXRX 061642	DVIN4 063060	ENADD 017314	E\$END = 002100
C\$QIO = 000377	DLVM 032040	DVTM4A 063006	ENDALL 042776	E\$LOAD= 000035
C\$RDBU= 000007	DMCM 032031	DVIN6 063110	ENDEVT 044204	FACSIM 045572
C\$REFG= 000047	DMP = 000000	DVM 032054	ENDHWL 067000	FHDPLX 017406
C\$RESE= 000033	DMPE = 000053	DVMODS 063756	ENDIT 051652	FLAG 017414
C\$REVI= 000003	DMPM 032044	DVOUTS 066674 G	ENDQ0 = 000017	FTLFLG 017324
C\$RFLA= 000021	DMPQ = 000054	DVRCC 017272	EOP = 000024	F\$AU = 000015
C\$RPT = 000025	DMP5 = 000052	DVRCLS 015612	EQUQ 025055	F\$AUTO= 000020
C\$SEFG= 000046	DMP6 = 000004	DVRCT 017274	FQUQ1 025111	F\$BGN = 000040
C\$SPRI= 000041	DMSGAD 002176	DVREX 064066	_QUQ2 025151	F\$CLEA= 000007
C\$SVEC= 000037	DMSGCT 002150	DVRTB 017266	ERRCNT 017310	F\$DU = 000016
C\$TPRI= 000013	DMV = 000001	DVRXA 017270	ERR1 041340 G	F\$END = 000041
DAM 032022	DMVDF1 016202	DVRXQ 064006	ERR10 041430 G	F\$HARD= 000004
DATCKB= 000002	DMVDF2 016204	DVTCC 017252	ERR13 041460 G	F\$HW = 000013
DCD = 000001	DMVDF3 016210	DVTCLS 015652	ERR14 041512 G	F\$INIT= 000006
DCK = 000014	DMVDF4 016212	DVTCT 017256	ERR15 041550 G	F\$JMP = 000050
DCLFLG 017376	DMVDF5 016214	DVTREX 064622	ERR16 041610 G	F\$MOD = 000000
DDF = 000022	DMVM 032105	DVTRX1 064616	ERR2 041402 G	F\$MSG = 000011
DEADTM 040245	DNM 032035	DVTR3 064150	ERX = 000100	F\$PROT= 000021
DER = 000010	DOGLOB 045776	DVTR4 064304	ETRB = 000060	F\$PWR = 000017
DEVPAR 023102	DOGL1 046036	DVTR4A 064370	ETWS = 000065	F\$RPT = 000012
DEV1 020652	DOGL2 046146	DVTR4B 064442	ETX = 000200	F\$SEG = 000003
DEV2 020654	DOGL4 046154	DVTR5 064446	EVL = 000004 G	F\$SOFT= 000005
DEV3 020656	DOW = 000004	DVTR5A 064532	EVMCTS 031530	F\$SRV = 000010
DEV4 020660	DPLX 067000	DVTB 017254	EVMDCD 031540	F\$SUB = 000002
DFPTBL 002130 G	DPM 032006	DVTXA 017250	EVMDSR 031534	F\$SW = 000014
DIAGMC= 000000	DQM 032017	DVTXRX 064070	EVMOCG 031410	F\$TEST= 000001
DISCON 041150	DSR = 000010	DZM 032057	EVMOH0 031433	GARPEX 047006
DIVN15 063274	DTEM 032050	ECHO = 000037	EVMOS1 031513	GARPFL 046744
DLE = 000020	DUM 032011	ECHOB = 000004	EVMRI 031550	GARP1 046760
DLL 061420	DUMEX 045346	EDDCK 030144	EVMRTS 031544	GATCEX 047072
DLLAB 040022	DUMPSR 045212	EDDDE 030257	EVMSQD 031554	GATCFL 047016
DLLCM 027160	DUM1 045304	EDDER 030127	EVMTM 031560	GATC1 047036
DLLEA 061626	DUM2 045326	EDDLE 030222	EVTADD 020616	GETCL 052472
DLLE1 062104	DUM3 045242	EDDVI 030174	EVTBCT 020620	GETIND 047154
DLLE2 061752	DUM4 045216	EDEOP 030312	EVTEND 020522	GETI1 047160
DLLE3 062164	DUPM 032025	EDMOS 030326	EVTFO 030447	GETI2 047200
DLLE4 062154	DVEM0 037116	EDRXC 030101	EVTF1 030545	GETPRM 051316
DLLE5 062274	DVEM1 037177	EDRXQ 030054	EVTF2 030574	GETRCL 042602
DLLE5A 062212	DVEM2 037257	EDTXC 030025	EVTF3 030646	GLBDEF 016206
DLLE5B 062246	DVEM3 037354	EDTXQ 030001	EVTF3C 030660	GLBEND 016220
DLLE6 062036	DVEM4 037430	EFM11 027724	EVTF3D 030675	GLBPLS 017220

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 113 3  
 Symbol table

GNTXPR	047102	G#HILI=	000002	INIT1	050766	LOGEOP	042250	L#LUN	002074	G
GNTX1	047120	G#LOLI=	000001	INOVN	= 002000	LOGEX	042552	L#MREV	002050	G
GNTX2	047112	G#NO =	000000	INTPRI	023076	LOGMSC	042266	L#NAME	002000	G
GOOD	017370	G#OFFS=	000400	INVEC	023072	LOGRXC	042074	L#PRIO	002042	G
GRPTCP	046546	G#OFFSI=	000376	ISR	= 000100	LOGRXQ	042056	L#PROT	050744	G
GSSIND	021130	G#PRMA=	000001	IXE	= 004000	LOGS1	042304	L#PRT	002112	G
GSSLST	021030	G#PRMD=	000002	I\$AU	= 000041	LOGS2	042544	L#REPP	002062	G
GSSOA	035004	G#PRML=	000000	I\$AUTO=	000041	LOGS3	042344	L#REV	002010	G
GSS1A	035045	G#RADA=	000140	I\$CLN =	000041	LOGS3A	042240	L#RPT	050736	G
GSS10A	035425	G#RADB=	000000	I\$DU =	000041	LOGS4	042420	L#SPC	002056	G
GSS11A	035477	G#RADDB=	000000	I\$HRD =	000041	LOGS5	042444	L#SPCP	002020	G
GSS12A	035517	G#RADL=	000120	I\$INIT=	000041	LOGTXC	042040	L#SPTP	002024	G
GSS13A	035555	G#RADO=	000020	I\$MOD =	000041	LOGTXQ	042022	L#STA	002030	G
GSS14A	035616	G#XFER=	000004	I\$MSG =	000041	LOGUNT	017372	L#TEST	002114	G
GSS15A	035637	G#YES =	000010	I\$PROT=	000040	LOOPS	003362	L#TIML	002014	G
GSS16A	035747	HELP =	000000	I\$PTAB=	000041	LOT =	000010	L#UNIT	002012	G
GSS17A	036057	HELPDC=	000000	I\$PWR =	000041	LPO	026765	L10000	002150	
GSS2A	035103	HLP =	000005	I\$RPT =	000041	LP00	026766	L10001	041400	
GSS20A	036141	HLPEND	003304	I\$SEG =	000041	LP1	026775	L10002	041426	
GSS21A	036206	HLPF	024116	I\$SETU=	000041	LP2	027006	L10003	041456	
GSS22A	036253	HLPTAB	003260	I\$SRV =	000041	LP3	027014	L10004	041510	
GSS23A	036320	HLP0	024040	I\$SUB =	000041	LP4	027027	L10005	041546	
GSS24A	036365	HLP1	024123	I\$TST =	000041	L#ACP	002110	L10006	041606	G
GSS25A	036432	HLP2	024136	J\$JMP =	000167	L#APT	002036	L10007	041646	G
GSS26A	036477	HLP2B	024254	KALL =	000062	L#AU	052056	L10010	042020	G
GSS27A	036521	HLP2C	024344	KDPM	032072	L#AUT	002070	L10011	050742	G
GSS3A	035142	HLP3	024417	KDZM	032076	L#AUTO	051770	L10013	051766	G
GSS30A	036543	HLP3A	024504	KEYWD1	003252	L#CCP	002106	L10014	051770	G
GSS31A	036576	HLP4	024531	KLM	032102	L#CLEA	051772	L10015	052046	G
GSS32A	036653	HLP4A	024610	KTRB =	000061	L#CO	002032	L10016	052054	G
GSS33A	036736	HLP5	024666	LCLKEN=	000100	L#DEPO	002011	L10017	052062	G
GSS34A	036761	HLP6	024756	LCPREX	046264	L#DESC	023304	L10020	066704	G
GSS35A	037026	HLTREX	065112	LCPRLS	046234	L#DESP	002076	L10021	066672	G
GSS36A	037050	HLTTRB	064776	LCPRL1	046266	L#DEVP	002060	L10022	066702	G
GSS37A	037071	HLTTR1	065040	LCPR1	046242	L#DISP	002124	L10023	067000	G
GSS4A	035177	HLTTR2	065032	LCPR2	046314	L#DLY	002116	L5060	027121	G
GSS5A	035234	HLTTR3	065024	LCPTEX	046434	L#DTP	002040	MARM	040161	G
GSS6A	035271	HOE =	100000	LCPTLS	046364	L#DTP	002034	MARM	040140	G
GSS7A	035347	IBE =	010000	LCPT1	046406	L#DU	052050	MCLR =	040000	G
GTRA2	052072	IDU =	000040	LDRCLS	046604	L#DUT	002072	MLTYF	017404	G
GTRA3	052134	IEI =	000001	LDRPLS	046506	L#DVTY	023266	MOBITE	020544	G
GTRA4	052142	IER =	020000	LDTCLS	046704	L#EF	002052	MOBITS	020526	G
GTRA5	052402	IEO =	000020	LDTPLS	046644	L#ENVI	002044	MOCHK =	000010	G
GTREX	057576	INDEX	015762	LGDVE	042104	L#ETP	002102	MODE	017420	G
GTRX2	057674	INDW	015760	LIS =	000006	L#EXP1	002046	MODES	003344	G
GTRX2A	057742	INFOHA	065724	LISCK	062530	L#EXP4	002064	MODLOC=	000003	G
GTRX2B	060024	INFOHD	065650	LISCKA	062570	L#EXP5	002066	MODREM=	000004	G
GTRX2C	057702	INFOH1	065672	LISMOD=	000032	L#HARD	066710	MODS	020524	G
GTRX22	057730	INFOH2	066004	LISP	027103	L#HIME	002120	MODTYP	017402	G
GTR9	057056	INFOH4	066010	LMDLOP=	000046	L#HPCP	002016	MOMSGS	020544	G
GTVIND	046462	INFOH5	066016	LNCNT	017300	L#HPTP	002022	MOP =	000043	G
GTVI1	046466	INFOH6	066044	LOE =	040000	L#HW	002130	MOSC =	000056	G
GTXRXB	052072	INFOH7	066110	LOGAQR	047302	L#ICP	002104	MOO	026674	G
G#CNTD=	000200	INFOH8	065770	LOGCMD	042224	L#INIT	050752	MO1	026704	G
G#DELM=	000372	INFOH	041212	LOGCML	042206	L#LADP	002026	MO2	026715	G
G#DISP=	000003	INFO1B	065666	LOGCMP	042170	L#LAST	067350	MO3	026725	G
G#EXCP=	000400	ININT =	000001	LOGDVI	042122	L#LOAD	002100	MO4	026734	G

Symbol table

MOS 026751  
MOJ 026736  
MPLY 017232  
MSC 000016  
MSG 002736  
MSG LIM 000017  
MSGTRN 027415  
MSGTRU 027446  
MSGTYP 017332  
MSGO 002220  
MSGOC 002150  
MSG1 002221  
MSG1C 002152  
MSG2 002222  
MSG2C 002154  
MSG3 002223  
MSG3C 002156  
MSG4 002224  
MSG4C 002160  
MSG5 002324  
MSG5C 002162  
MSG6 002416  
MSG6C 002164  
MSG8 002646  
MSG8C 002170  
MTP 000001  
MTPLEX 046460  
MTPLY 046436  
NEW 051310  
NO 000036  
NOCLK 027256  
NOD0 021170  
NOD1 021174  
NOD10 021252  
NOD100 022102  
NOD101 022106  
NOD102 022124  
NOD103 022130  
NOD104 022144  
NOD105 022150  
NOD106 022164  
NOD107 022170  
NOD11 021254  
NOD110 022204  
NOD111 022210  
NOD112 022224  
NOD113 022230  
NOD114 022234  
NOD115 022250  
NOD116 022254  
NOD117 022272  
NOD12 021266  
NOD120 022276  
NOD121 022312  
NOD122 022316  
NOD123 022332  
NOD124 022336

NOD125 022352  
NOD126 022356  
NOD127 022372  
NOD13 021272  
NOD130 022376  
NOD131 022412  
NOD132 022416  
NOD133 022436  
NOD134 022442  
NOD135 022446  
NOD136 022452  
NOD137 022456  
NOD14 021306  
NOD140 022462  
NOD141 022466  
NOD142 022472  
NOD143 022474  
NOD144 022500  
NOD145 022504  
NOD146 022520  
NOD147 022524  
NOD15 021312  
NOD150 022540  
NOD151 022544  
NOD152 022550  
NOD153 022554  
NOD154 022560  
NOD155 022564  
NOD156 022606  
NOD157 022612  
NOD16 021326  
NOD160 022626  
NOD161 022632  
NOD162 022654  
NOD163 022660  
NOD164 022702  
NOD165 022706  
NOD166 022712  
NOD167 022716  
NOD17 021332  
NOD170 022722  
NOD171 022726  
NOD172 022742  
NOD173 022744  
NOD174 022764  
NOD175 022770  
NOD176 023004  
NOD177 023010  
NOD2 021200  
NOD20 021346  
NOD200 023022  
NOD201 023026  
NOD202 023040  
NOD203 023044  
NOD204 023050  
NOD205 044774  
NOD206 045000

NOD207 045004  
NOD21 021352  
NOD210 045006  
NOD211 045022  
NOD212 045024  
NOD213 045040  
NOD214 045042  
NOD215 045054  
NOD216 045060  
NOD217 045072  
NOD22 021356  
NOD220 045074  
NOD221 045106  
NOD222 045112  
NOD223 045116  
NOD224 045122  
NOD225 045136  
NOD226 045140  
NOD227 045154  
NOD23 021370  
NOD230 045156  
NOD231 045174  
NOD232 045200  
NOD233 045204  
NOD234 045206  
NOD235 045210  
NOD24 021374  
NOD25 021406  
NOD26 021412  
NOD27 021414  
NOD3 021202  
NOD30 021420  
NOD31 021434  
NOD32 021440  
NOD33 021456  
NOD34 021462  
NOD35 021500  
NOD36 021504  
NOD37 021522  
NOD4 021216  
NOD40 021526  
NOD41 021544  
NOD42 021550  
NOD43 021574  
NOD44 021600  
NOD45 021604  
NOD46 021622  
NOD47 021626  
NOD5 021220  
NOD50 021640  
NOD51 021644  
NOD52 021650  
NOD53 021672  
NOD54 021674  
NOD55 021720  
NOD56 021722  
NOD57 021726

NOD6 021234  
NOD60 021746  
NOD61 021752  
NOD62 021774  
NOD63 022000  
NOD64 022004  
NOD65 022010  
NOD66 022014  
NOD67 022020  
NOD7 021236  
NOD70 022024  
NOD71 022030  
NOD72 022034  
NOD73 022040  
NOD74 022044  
NOD75 022050  
NOD76 022062  
NOD77 022066  
NOEXM 041132  
NONE 000000  
NOTNUF 000050  
NULEV 030407  
NULL 000000  
NUM 000014  
N10\$ 021174  
N100\$ 021644  
N102\$ 021650  
N104\$ 021674  
N105\$ 022722  
N106\$ 022726  
N107\$ 022744  
N108\$ 022770  
N110\$ 021722  
N111\$ 021726  
N112\$ 021752  
N114\$ 022044  
N115\$ 022040  
N116\$ 022062  
N117\$ 022106  
N118\$ 022130  
N120\$ 022230  
N121\$ 022474  
N122\$ 022524  
N123\$ 022500  
N125\$ 023050  
N126\$ 022544  
N130\$ 022150  
N131\$ 022210  
N132\$ 022170  
N140\$ 022560  
N141\$ 022564  
N142\$ 022612  
N143\$ 022632  
N144\$ 022660  
N150\$ 022706  
N160\$ 023004  
N161\$ 023010

N162\$ 023022  
N163\$ 023040  
N20\$ 021352  
N25\$ 021374  
N30\$ 021412  
N40\$ 021312  
N42\$ 021202  
N43\$ 021220  
N44\$ 021236  
N45\$ 021254  
N46\$ 021272  
N47\$ 021332  
N50\$ 022000  
N51\$ 022004  
N52\$ 022010  
N60\$ 022234  
N61\$ 022254  
N62\$ 022276  
N63\$ 022316  
N64\$ 022336  
N65\$ 022356  
N66\$ 022376  
N67\$ 022416  
N68\$ 022442  
N70\$ 022446  
N71\$ 022456  
N72\$ 022462  
N73\$ 022472  
N80\$ 021414  
N81\$ 021420  
N82\$ 021462  
N83\$ 021504  
N84\$ 021526  
N85\$ 021550  
N86\$ 021600  
N87\$ 021626  
OFSET 017346  
OPBFPT 002520  
OPBUF 002524  
OPCNT 002166  
OPEND 002646  
OPRMM 027114  
OPRMSG 000015  
OPTYP 023100  
OPTYPM 067226  
OPVAR 017302  
OPVAR1 017304  
OTINT 000002  
OUTHDL 065254  
OUTHEX 066452  
OUTHE2 066626  
OUTHE3 066544  
OUTHE4 066466  
OUTHE5 066504  
OUTHE6 066606  
OUTH1 065304  
OUTVEC 023074

CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 113-5  
Symbol table

OVRcnt 017330  
 O\$APTS- 000000  
 O\$AU - 000001  
 O\$BGNR- 000001  
 O\$BGNS- 000000  
 O\$DU - 000001  
 O\$ERRT- 000000  
 O\$GNSM- 000000  
 O\$POIN- 000001  
 O\$SETU- 000000  
 PARAM 017410  
 PAS - 000002  
 PASC - 000042  
 PASHOD- 000030  
 PASS1 002650  
 PASS2 002651  
 PASS3 002652  
 PASS4 002653  
 PCADD 017374  
 PCK 027056  
 PCLKCT- 001600  
 PCLKEN- 000111  
 PCPM 027672  
 PEC 027066  
 PE100M 040331  
 PE102M 040345  
 PE104M 040367  
 PE106M 040413  
 PE110M 040444  
 PE112M 040466  
 PE114M 040517  
 PE116M 040542  
 PE120M 040576  
 PE122M 040622  
 PE124M 040652  
 PE126M 040701  
 PE130M 040730  
 PE132M 040756  
 PE134M 041002  
 PE136M 041034  
 PE140M 041056  
 PE142M 041075  
 PE144M 041103  
 PLCK 060122  
 PMS 027075  
 PNCK 027054  
 PNEC 027064  
 PNMS 027073  
 PNS 027043  
 PNT - 001000 G  
 PNTTRB 044326  
 POLDEF 016166  
 POLLEN 063664  
 POLLE2 063736  
 POLLIS 016220  
 POLPM 025234  
 POLPM3 025301

PRI - 002000 G  
 PRIOR 067111  
 PRI00 - 000000 G  
 PRI01 - 000040 G  
 PRI02 - 000100 G  
 PRI03 - 000140 G  
 PRI04 - 000200 G  
 PRI05 - 000240 G  
 PRI06 - 000300 G  
 PRI07 - 000340 G  
 PRNT - 000055  
 PROTO - 000041  
 PROTOB- 000040  
 PSCNT 017306  
 PST 027045  
 PTPMLP 067137  
 PTREND 015406  
 PTRTAB 011416  
 PTR13 011512  
 PTR23 011606  
 P\$ACT 003400  
 P\$BUFA 003374  
 P\$CNT 003402  
 P\$EXIT 047764  
 P\$GDBD 003411  
 P\$INUF 003410  
 P\$INUM 003404  
 P\$RADX 003406  
 P\$TREE 003376  
 P\$TRV 047646  
 P\$TR5 047656  
 QCOPY - 000013  
 QRX - 000004  
 QTX - 000010  
 QUALFG 003254  
 QUALVL 003256  
 QUEOM 041162  
 RBFLIM- 004000  
 RDI - 000020  
 RDO - 000200  
 RDTSS 043112  
 RDT52 043030  
 REC - 000000  
 RECMOD- 000031  
 REPLOG 043224  
 REPORT 042554  
 RESFLG 017400  
 RESTRT 051250  
 RMLPEN 003314  
 RMLPTB 003304  
 RMLPO 027341  
 RMLP1 025377  
 RMLP2 025421  
 RMLP3 025436  
 RMLP4 025470  
 RI - 000200  
 RIM 040226

RMDLOP- 000047  
 RNOTNF- 000012  
 RPASS 017412  
 RPEXT - 000002  
 RPGSS - 000004  
 RPHLP - 000001  
 RPLOG - 000003  
 RPSWE - 000007  
 RPSWF - 000010  
 RPSWO - 000011  
 RPT 043272  
 RPTAA 043342  
 RPTDCK 044056  
 RPTDDE 043776  
 RPTDER 043522  
 RPTDLE 044056  
 RPTDSP 020624  
 RPTDVI 043622  
 RPTDOP 043672  
 RPTIV 025544  
 RPTMSB 044214  
 RPTMSC 044136  
 RPTSNT - 000006  
 RPTSS - 000005  
 RPTTSS 043006  
 RPTTXQ 043444  
 RPTO 043336  
 RPT1 043324  
 RQI - 000200  
 RQIFLG 017322  
 RSEL3 066654  
 RSEL4 066650  
 RSEL6 066652  
 RSPTRE 015772  
 RSPTRS 015770  
 RTS - 000040  
 RUN - 000004  
 RUNNING 017416  
 RUNST - 000400  
 RUSH 040257  
 RXBUF 005416  
 RXC - 000006  
 RXMTOT 017276  
 RXM1 041272  
 RXM2 041315  
 RXNC 041252  
 RXONLY 060036  
 RXON3 060054  
 RXPTR 017234  
 RXQ - 000004  
 RXQUAL 046164  
 RXQUEX 046232  
 RXQU1 046172  
 RXSKEN 016166  
 RXSTAK 016012  
 RXTHEM 041075  
 R101 045000

R111 045006  
 R121 045024  
 R1251 045210  
 R131 045042  
 R141 045060  
 R151 045074  
 R201 045116  
 R211 045140  
 R221 045156  
 R301 045206  
 SAVSP 017364  
 SCM 031635  
 SCMD 031670  
 SCML 031657  
 SDVE 031624  
 SDVI 031646  
 SECRM 031724  
 SEL0 023052  
 SEL2 023056  
 SEL4 023062  
 SEL6 023066  
 SEOP 031701  
 SETET - 000067  
 SETEXP- 000010  
 SETTRN- 000011  
 SHFO 027531  
 SHF1 027567  
 SHMSG 025613  
 SHOW - 000002  
 SHTAB 003334  
 SHTAP 026017  
 SHTBR 026605  
 SHTEND 003343  
 SHTFL 026027  
 SHTIV 026543  
 SHTLP 026231  
 SHTLPA 026316  
 SHTLPB 026371  
 SHTLPC 026433  
 SHTLPD 026504  
 SHTNF 026165  
 SHTRE 025727  
 SHTRM 025766  
 SHTUN 026115  
 SHTYP0 025647  
 SHTYP1 025656  
 SHTYP2 025663  
 SHTYP3 025670  
 SHTYP4 025675  
 SHTYP5 025703  
 SHTYP6 025710  
 SHTYP7 025716  
 SHTYTB 003314  
 SHWOP 047344  
 SIZE - 000012  
 SLST - 000057  
 SLTHEM 040076

SMSC 031712  
 SQD - 040000  
 SRXQ 031613  
 STADD 017312  
 START 051046  
 STATB - 000001  
 STATUS- 000016  
 STATYP 023104  
 STRCM 040117  
 STREAM 040312  
 STRMM 040203  
 STXC 031602  
 STXQ 031571  
 SVCGBL- 000000  
 SVCINS- 000001  
 SVCSUB- 000001  
 SVCTAG- 000001  
 SVCTST- 000001  
 S\$LSYM- 010000  
 S1 051034  
 S2 051112  
 S3 051162  
 S4 051230  
 TABEX 027337  
 TAL - 000005  
 TALCK 062276  
 TALMOD- 000035  
 TCURAD 017264  
 TEMP 017350  
 TEMP1 017352  
 TEMP2 017354  
 TEMP3 017356  
 TEMP4 017360  
 TEMP5 017362  
 TIME7/5 017462  
 TIME/R1 017456  
 TIME/R2 017460  
 TIMIN 017450  
 TIMSEC 017452  
 TIMTCK 017454  
 TM - 002000  
 TOINOT 064156  
 TOIN1 064252  
 TOIN2 064254  
 TOORIO 065114  
 TOOR1 065216  
 TOOR2 065234  
 TOOR3 065444  
 TOTCC 017344  
 TRA - 000001  
 TRAMOD- 000034  
 TRBB - 000002  
 TRBTOT 015754  
 TRIBCO 067173  
 TRIBLS 015712  
 TRIBN 015756  
 TRVACT 047766

Symbol table

TRVALN	050560	TSS12A	032675	TSS7A	032545	T#NS0	= 000000	ULRCLS	046624
TRVALP	050514	TSS13A	032772	TS30AA	034242	T#NS1	= 000004	ULRPLS	046526
TRVBIF	050072	TSS14A	033066	TTL	= 000001	T#NS2	= 000010	ULTCLS	046724
TRVBR	050062	TSS15A	033151	TTLLOP	= 000044	T#PTNU	= 000000	ULTPLS	046664
TRVBRC	050006	TSS16A	033235	TTOTCC	017262	T#SAVL	= 177777	UNKM	032062
TRVDEC	050166	TSS17A	033320	TXBUF	003416	T#SEGL	= 177777	UPTABL	060352
TRVERR	050024	TSS2A	032216	TXC	= 000002	T#SUBN	= 000000	UPTA1	060436
TRVEXI	050044	TSS20A	033404	TXMTOT	017260	T#TAGL	= 177777	UPTA4	060402
TRVMA	050206	TSS21A	033476	TXNC	041232	T#TAGN	= 010024	UPTEX	060516
TRVNOB	050016	TSS22A	033565	TXONLY	060064	T#TEMP	= 000000	VALTRB	003414
TRVNUM	050200	TSS23A	033653	TXON2	060072	T#TEST	= 000001	VECTOR	067056
TRVOCT	050200	TSS24A	033740	TXPTR	017236	T#TSTM	= 177777	WRDEFP	047214
TRVSPA	050114	TSS25A	034027	TXQ	= 000000	T#TSTS	= 000001	WRDE5A	047236
TRVSTR	050646	TSS26A	034114	TXSTAK	015776	T#AU	= 010017	WRDE5B	047232
TSEL3	066646	TSS27A	034145	TXTHEM	041075	T#AUT	= 010014	WRDE5D	047262
TSEL4	066644	TSS3A	032252	T#ARGC	= 000006	T#CLE	= 010015	WRFLG	003412
TSEL6	066642	TSS30A	034230	T#CODE	= 004130	T#DU	= 010016	WRIPEX	064730
TSPTR	015774	TSS31A	034302	T#ERRN	= 000015	T#HAR	= 010023	WRIPP	064634
TSSA	021024	TSS32A	034361	T#EXCP	= 000000	T#HM	= 010000	WRIPPG	064624
TSSE	021022	TSS33A	034444	T#FLAG	= 000040	T#INI	= 010013	WRIP1	064642
TSSFLG	017326	TSS34A	034527	T#GMAN	= 000000	T#MSG	= 010007	WRMCS	064732
TSSIND	020762	TSS35A	034606	T#HILI	= 000004	T#PRO	= 010012	X\$	= 000236
TSSKEY	021026	TSS36A	034665	T#LAST	= 000001	T#RPT	= 010011	X\$ALWA	= 000000
TSSLST	020662	TSS37A	034737	T#LOLI	= 000000	T#SRV	= 010022	X\$FALS	= 000040
TSSOA	032112	TSS4A	032325	T#LSYM	= 010000	T#TES	= 010020	X\$OFFS	= 000400
TSS1A	032146	TSS5A	032373	T#LTNO	= 000001	T1	052064 G	X\$TRUE	= 000020
TSS10A	032603	TSS6A	032455	T#NEST	= 177777	UAM	= 000200 G	\$PATCH	067264
TSS11A	032637								

. ABS. 067350 000 (RW,I,GBL,ABS,OVR)  
 000000 001 (RW,I,LCL,REL,CON)

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 359  
 Work file writes: 351  
 Size of work file: 30056 Words ( 118 Pages)  
 Size of core pool: 17408 Words ( 68 Pages)  
 Operating system: RT 11 (Under RSTS/E)

Elapsed time: 00:04:26.28  
 .CZCLMC/C=SVC34R,CZCLMC



CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page 5 1  
Cross reference table (CREF V05.00)

SPATCH	113-380									
ACT	26-180	82 73	86 1	87 49						
ACTATV	83 36	86-10								
ACTBCR	83-54	85 450								
ACTCHK	83-16	87 150								
ACTCKT	83-65	88-1700								
ACTCLB	84-42	84-560								
ACTCLP	83-50	87 430								
ACTCLR	83-14	84-50								
ACTCOP	83 24	85 130								
ACTCRC	83 45	87-210								
ACTCSE	83 19	84-280								
ACTCST	83-20	84-440								
ACTDLL	83 40	86-150								
ACTDME	83 56	84-90	84-930							
ACTDMQ	83-57	84-920								
ACTDMS	83 55	84-870								
ACTDMX	84 940									
ACTESA	88-1330	88-135								
ACTESB	88 1320									
ACTECH	83-44	87-40								
ACTEKE	88-115	88-1380								
ACTEKT	83-64	88-1010								
ACTEQO	83-28	85-340								
ACTETB	83-61	88-10								
ACTEWS	83 66	88-230								
ACTEXT	83 67	84-170								
ACTEXX	88 106	88 113	88-123	88-136	88-146	88-164	88-1670			
ACTHLP	83-18	84-110								
ACTKAL	82-79	83-63	88-70							
ACTKTB	83-62	88-40								
ACTLIS	83 39	86-120								
ACTLLP	83 51	87-450								
ACTLPX	87-40	87-42	87-44	87-46	87-490					
ACTLXX	87-13	87-34	87-37	87-50	87-540					
ACTM2X	86-2	86-10	86-13	86-16	86-19	86-230				
ACTME1	85-49	85-51	85 53	85-55	85-57	85 640				
ACTMEX	85-27	85-43	85-60	85-65	85-72	85-75	85-790			
ACTMOP	83 48	87-390								
ACTMOS	83-59	87-240								
ACTMSO	83-29	85-480								
ACTMS1	83-30	85-500								
ACTMS2	83-31	85-520								
ACTMS3	83-32	85-540								
ACTMS4	83-33	85-560								
ACTMS5	83-34	85-580								
ACTMS6	83-35	85-610								
ACTNO	83-43	87-10								
ACTNUF	83 53	84-20								
ACTNUL	83-13	84-30								
ACTNUM	83-25	85-160								
ACTOPM	83-26	85-290								
ACTPAS	83 37	86-40								
ACTPRO	83-46	87-270								
ACTPRT	83 58	84-190								
ACTQFG	87-16	87 19	87-22	87-25	87 290					

CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 5 2  
 Cross reference table (CREF V05.00)

ACTREC	83-38	86-90		
ACTREX	48 11	49-100		
ACTRGS	48 13	49-150		
ACTRHL	48-10	49-40		
ACTRLG	48-12	49-120		
ACTRLP	83-52	87-470		
ACTRNF	48-19	49 20		
ACTRNL	48-9	49-30		
ACTRPS	83-47	87-360		
ACTRSE	48-16	49-190		
ACTRSF	48-17	49-370		
ACTRSO	48-18	49-400		
ACTRTN	48 15	49-250		
ACTRTS	48-14	49-210		
ACTRUN	83-17	84-230		
ACTJEX	83-68	85-690		
ACTSHO	83-15	84-80		
ACTSHW	84-34	84-48	84-670	84-79
ACTSLS	83-60	88-730		
ACTSTE	83-21	85-30		
ACTSTS	83-27	87-180		
ACTSTT	83-22	85-60		
ACTSTX	85-4	85-70		
ACTSZE	83-23	85-100		
ACTTAL	83-42	86-210		
ACTTLP	83-49	87-410		
ACTTRA	83-41	86-180		
ACTW7A	88-570			
ACTW7B	88-27	88-55	88-590	
ACTWS1	88-48	88-490		
ACTWS2	88-49	88-510		
ACTWS3	88-50	88-660		
ACTWS5	88-37	88-390	88-58	
ACTWS7	88-530	88-71		
ACTWS9	88-280			
ADDC1	52-26	52-360		
ADDC	52-240	82-160	82-191	
ADR	25-00			
ALCK	32-30	93-200		
ALCK1	94-47	94-510	94-102	
ALCK2	94-52	94-630	94-146	
ALCK2A	94-98	94-1040		
ALCK3	94-66	94-108	94-1120	
ALCK3A	94-122	94-1320		
ALCK3B	94-134	94-1370		
ALCK3C	94-130	94-1350	94-143	
ALCK3D	94 125	94-1280		
ALCK4	94-113	94-1410		
ALCK4A	94 140	94-1440		
ALCK5	94-460	94-136		
ALCK5B	94-480			
ALLTR	90-21	91-22	92-23	94 450
ASSEMB	21 12	21-12		
ATVMOO	26-1450	37-48		
BABTM	38 53	41-1290		
BACCHD	107-46	107-1790		











CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22-Mar-84 16:24 Page 5-8  
 Cross reference table (CREF V05.00)

DMVDF2	30-16*	77-141*															
DMVDF3	30-21*	77-142*															
DMVDF4	30-22*	77-143*															
DMVDF5	30-23*	77-144*															
DMVM	38-105	40-230*															
DNM	38-95	40-220*															
DOGL1	55-20*	55-33															
DOGL2	55-32*																
DOGL4	55-30	55-35*															
DOGLO8	55-13*	82-138															
DOW	26-19*	86-15	100-84							100						236	
DPLX	112-27	112-62*															
DPM	38-88	40-213*															
DQM	38-91	40-216*															
DSR	26-208*	34-4															
OTEM	38-98	40-223*															
DUM	38-89	40-214*															
DUM1	51-32	51-36*															
DUM2	51-35	51-37*															
DUM3	51-31*	51-42															
DUM4	51-29*	51-41															
DUMEX	51-38	51-44*															
DUMPSR	51-28*	82-125															
DUPH	38-93	40-213*															
DVEM0	41-90*	100-110	100-115														
DVEM1	41-93*	100-121	100-127														
DVEM2	41-96*	103-91	103-97														
DVEM3	41-99*	106-35	106-40														
DVEM4	41-102*	107-111	107-114														
DVEM5	41-105*	103-114	103-119														
DVEM6	41-108*	103-136	103-141														
DVEM7	41-111*	107-197	107-212														
DVEM8	41-114*	100-91	100-96														
DVES1	100-205*	100-241															
DVES1A	100-174	100-180	100-192*														
DVEST	100-211*																
DVI	26-76*	46-63															
DVIN1	100-105	100-119*															
DVIN11	100-134	100-149*															
DVIN12	100-132	100-156*															
DVIN13	100-120	100-131*															
DVIN4	100-85	100-87	100-98*														
DVIN4A	100-89*	100-90															
DVIN6	100-100	100-104*	100-108														
DVINEX	100-207	100-242*															
DVINIT	89-78	100-75*	100-97							100-117			100-128				
DVINS	77-165	108-22*															
DVM	38-99	40-224*															
DVMODS	46-96	101-40*															
DVOUTS	77-166	108-42*															
DVRCC	31-17*	72-17*	94-68			97-77*	97-114		99-27*	99-38		99-40*	99-41*	102-49		103-146*	
DVRCLS	29-17*	58-29*	65-17*			65-37											
DVRCT	31-18*	65-17	65-37*			94-105*	94-107		95-42*	95-87*							
DVREX	102-36	102-54*															
DVRTB	31-15*	94-69	103-151*														
DVRXA	31-16*	72-15*	94-67			97-75*	97-113		99-25*	99-37		99-40	102-48		103-147*		





CZCLMCO DMP/V-11 DCLT .IACRO V05.00 Thursday 22-Mar-84 16:24 Page 5 10  
Cross reference table (CREF V05.00)

EQUQ2	40-32#	88-69												
ERR1	43-28#	95-75												
ERR10	43-36#	95-61												
ERR13	43-54#	100-96	100-115	100-127	103-97	103-119	103-141	106-40						
ERR14	43-63#	97-67	107-114											
ERR15	43-68#	107-197												
ERR16	43-72#	107-212												
ERR2	43-32#	95-84												
ERRCNT	31-25#	46-94	46-98	89-75*	95-60*	95-74*	95-83*	96-23	100-95*	100-114*	100-126*	103-96*	106-39*	107-113*
ERX	26-89#	90-17	92-21	93-26	94-127	97-48	99-29	103-134						
ETRB	26-170#	37-205	88-1	88-114										
ETWS	26-175#	37-211												
ETX	26-90#	91-19	93-26	94-101	97-87	97-104	98-40	98-52	103-112					
EVL	25-0#													
EVMCTS	34-14	40-186#												
EVMDCD	34-16	40-188#												
EVMDSR	34-15	40-187#												
EVMOCG	40-179#	47-171												
EVMOHQ	40-184#	47-188												
EVMOST	40-185#	47-203												
EVMRI	34-18	40-190#												
EVMRTS	34-17	40-189#												
EVMSQD	34-19	40-191#												
EVMTH	34-20	40-192#												
EVTADD	34-42#	47-111*	47-115	47-138*	47-145	47-150*	47-154	47-158*	47-162					
EVTBCT	34-43#	47-112*	47-115	47-139*	47-145	47-151*	47-154	47-159*	47-162					
EVTEND	33-4#	46-131	47-93	77-87										
EVTFO	40-153#	47-103												
EVTF1	40-154#	47-108												
EVTF2	40-155#	47-115												
EVTF3	40-156#	47-123												
EVTF3C	40-157#	43-55	47-124											
EVTF3D	40-158#	43-64												
EVTF3F	40-159#	43-69	43-73											
EVTF4	40-174#	47-154												
EVTF44	40-173#	47-146												
EVTF4A	40-175#	47-162												
EVTF4B	40-172#	47-145												
EVTF5A	40-177#	43-29												
EVTF6	40-160#	47-209												
EVTLOG	33-2	33-3#	46-135	47-83	47-91	77-84								
EVTLST	34-25#	47-108												
EVTMIN	34-40#	47-107*	47-108											
EVTPTR	33-2#	46-120	46-136*	47-82	47-98	77-85*								
EVTSEC	34-39#	47-106*	47-108											
EVTTC	34-41#	47-105*	47-108											
EVTTMP	34-44#	47-119*	47-123	47-140*	47-146	47-152*	47-154	47-160*	47-162					
EXIT	26-176#	37-26	82-127	84-17										
F\$AU	21-12#	81-9	81-36											
F\$AUTO	21-12#	78-10	78-19											
F\$BGN	21-12#	21-33	43-28	43-32	43-36	43-54	43-63	43-68	43-72	45-33	75-9	76-8	77-8	77-169
	78-10	79-8	79-27	80-8	81-9	82-23	82-95	82-130	108-22	108-42	109-15	112-13	113-49	
F\$CLEA	21-12#	79-8	79-43											
F\$DU	21-12#	80-8	80-35											
F\$END	21-12	21-12	21-12	21-12	21-12	21-12	21-12	21-12	21-12	21-12	21-12	21-12	21-12	21-12
	21-12	21-12	21-12#	21-33	43-30	43-34	43-38	43-56	43-65	43-70	43-74	43-76	45-57	75-41





CZCLMCO (MP/V 11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 5 13  
Cross reference table (CREF V05.00)

GTVI1	62-190	62-21												
GTVIND	56-27	57-28	60-32	62 180	68-19	69-18	70-15	89-34	89-89	95-36	97-37	98-21	99-22	100-205
GTXRXB	105-49													
HELP	82-350													
	2-30	21-3	21-29	21 72	21-100	22-10	23-12	24-32	24-46	24-62	38 108	39-27	39-40	42-16
	42-28	43-12	44-8	75-11	75-26	75-34	76-16	77-10	77-171	78 12	79-10	79-29	80-10	80-21
	81-11	81-22	82-10	82-17	82-25	109-1	109-17	112-15	112-86	113-14	113-27	113-41		
HELPDC	2-80	2-17	21-90	23-30	24-3	26-50	26-92	26-191	26-215	38-1	38-121	39-14	40-162	41 70
	42-2	43-40	77 101	77-157	97-28	100-3	100-39	100 77	101-3	101-34	102 2	102-40	103-3	103-63
	103-77	108-1	108-12	108-23	108-33	108-44	112-30	112-64						
HLP	26-1270	37-20	37-22	84-15										
HLP0	40-150	82-83												
HLP1	28-8	40-170												
HLP2	28-9	40-180												
HLP2B	28-10	40-210												
HLP2C	28-11	40-220												
HLP3	28-12	40-230												
HLP3A	28-13	40-240												
HLP4	28-14	40-250												
HLP4A	28-15	40-260												
HLP5	28-16	40-270												
HLP6	28-17	40-280												
HLPEND	28-180	84-13												
HLPF	40-160	49-5	84-12											
HLPTAB	28-80	84-11												
HLTRES	105-41	105-51	105-610											
HLTTR1	105-490	105-60												
HLTTR2	105-43	105-470												
HLTTR3	105-450	105-46												
HLTTRB	79-20	96-32	105-400											
HOE	25-00													
I\$AU	21-120	81-90	81-360											
I\$AUTO	21-120	78-100	78-190											
I\$CLN	21-120	79-80	79-27	79-430										
I\$DU	21-120	80-80	80-350											
I\$HRD	112-130	112-560												
I\$INIT	21-120	77-80	77-169	77-1850										
I\$MOD	21-120	21-33	21-330	113 49	113-490									
I\$MSG	21-120	43-280	43-300	43-320	43-340	43-360	43-380	43-540	43 560	43 630	43-650	43 680	43 700	43 720
	43-740													
I\$PROT	21-120	76-80												
I\$PTAB	21-120													
I\$PWR	21-120													
I\$RPT	21-120	75-90	75-410											
I\$SEG	21-120	82 23												
I\$SETU	21-120													
I\$SRV	21-120	45-330	45-570	108-220	108-310	108 420	108-500							
I\$SUB	21 120	82-23												
I\$YST	21-120	82-23	82-230	82-95	82-130	109 15	109-150	109 150						
IBE	25-00													
IDU	25-00													
IEO	26-2330													
IEI	26-2320													
IER	25-00													
INDEX	29-240	56-260	56-28	57-270	57-29	58-25	58-28	60-310	60 33	62 18	62 220	64 18	65 10	65 30
	67-18	67-36	68 180	68-20	69-160	69-19	69-21	70-12	70 140	70 16	70 420	88 32	88 1020	89 340



CZCLMCO GMP/V-11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 5-15  
 Cross reference table (CREF V05.00)

L\$HIME	21-98#					
L\$MPCP	21-98#					
L\$MPTP	21-98#					
L\$HW	21-98	23-10	23-10#			
L\$ICP	21-98#					
L\$INIT	21-98	77-8#				
L\$LADP	21-98#					
L\$LAST	21-98	113-48#				
L\$LOAD	21-98#					
L\$LUN	21-98#					
L\$MREV	21-98#					
L\$NAME	21-98#					
L\$PRIO	21-98#					
L\$PROT	21-98	76-8#				
L\$PRT	21-98#					
L\$REPP	21-98#					
L\$REV	21-98#					
L\$RPT	21-98	75-9#				
L\$SPC	21-98#					
L\$SPCP	21-98#					
L\$SPTP	21-98#					
L\$STA	21-98#					
L\$TEST	21-98#					
L\$TIML	21-98#					
L\$UNIT	21-98#	77-93				
L10000	23-10	23-59#				
L10001	43-30#					
L10002	43-34#					
L10003	43-38#					
L10004	43-56#					
L10005	43-65#					
L10006	43-70#					
L10007	43-74#	43-76				
L10010	45-57#					
L10011	75-41#					
L10013	77-169	77-185#				
L10014	78-19#					
L10015	79-27	79-43#				
L10016	80-19	80-35#				
L10017	81-20	81-36#				
L10020	82-95	82-130	109-15#			
L10021	108-31#					
L10022	108-50#					
L10023	112-13	112-56#				
L5060	40-99#	77-76				
LCLKEN	26-35#	77-59				
LCPR1	57-28#	57-32				
LCPR2	58-28#					
LCPREX	57-30	57-33#				
LCPR11	57-31	58-24#	94-80			
LCPR15	57-27#	90-18	92-20	93-25		
LCPT1	60-32#	60-37				
LCPTEX	60-34	60-38#				
LCPTLS	60-29#	91-18	92-19	93-24		
LDRCLS	65-15#	94-106				
LDRPLS	56-33	63-14#	94-50	94-82	94-95	94-111









CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22-Mar 84 16:24 Page 5-19  
 Cross reference table (CREF V05.00)

NOD102	37-108	37-108#	
NOD103	37-109	37-109	37-109#
NOD104	37-110	37-110#	
NOD105	37-124	37-124	37-124#
NOD106	37-125	37-125#	
NOD107	37-127	37-127	37-127#
NOD11	37-28	37-28	37-28#
NOD110	37-128	37-128#	
NOD111	37-130	37-130	37-130#
NOD112	37-131	37-131#	
NOD113	37-134	37-134#	
NOD114	37-137	37-137	37-137#
NOD115	37-138	37-138#	
NOD116	37-139	37-139	37-139#
NOD117	37-140	37-140#	
NOD12	37-29	37-29#	
NOD120	37-141	37-141	37-141#
NOD121	37-142	37-142#	
NOD122	37-143	37-143	37-143#
NOD123	37-144	37-144#	
NOD124	37-145	37-145	37-145#
NOD125	37-146	37-146#	
NOD126	37-147	37-147	37-147#
NOD127	37-148	37-148#	
NOD13	37-30	37-30	37-30#
NOD130	37-149	37-149	37-149#
NOD131	37-150	37-150#	
NOD132	37-151	37-151	37-151#
NOD133	37-152	37-152#	
NOD134	37-155	37-155#	
NOD135	37-156	37-156#	
NOD136	37-157	37-157#	
NOD137	37-158	37-158#	
NOD14	37-31	37-31#	
NOD140	37-159	37-159#	
NOD141	37-160	37-160#	
NOD142	37-161#		
NOD143	37-164	37-164#	
NOD144	37-165	37-165#	
NOD145	37-166	37-166	37-166#
NOD146	37-167	37-167#	
NOD147	37-168	37-168	37-168#
NOD15	37-32	37-32	37-32#
NOD150	37-169	37-169#	
NOD151	37-172	37-172#	
NOD152	37-173	37-173#	
NOD153	37-174	37-174#	
NOD154	37-177	37-177#	
NOD155	37-188	37-188	37-188#
NOD156	37-189	37-189#	
NOD157	37-190	37-190	37-190#
NOD16	37-33	37-33#	
NOD160	37-191	37-191#	
NOD161	37-192	37-192	37-192#
NOD162	37-193	37-193#	
NOD163	37-194	37-194	37-194#

CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar-84 16:24 Page 5 20  
 Cross reference table (CREF V05.00)

NOD164	37-195	37-195#	
NOD165	37-198	37-198#	
NOD166	37-199	37-199#	
NOD167	37-200	37-200#	
NOD17	37-34	37-34	37-34#
NOD170	37-202	37-202#	
NOD171	37-203	37-203	37-203#
NOD172	37-204#		
NOD173	37-205	37-205	37-205#
NOD174	37-206	37-206#	
NOD175	37-207	37-207	37-207#
NOD176	37-208	37-208#	
NOD177	37-209	37-209	37-209#
NOD2	37-21#		
NOD20	37-35	37-35#	
NOD200	37-210	37-210#	
NOD201	37-211	37-211	37-211#
NOD202	37-212	37-212#	
NOD203	37-213	37-213#	
NOD204	37-216#		
NOD205	50-3	50-3#	
NOD206	50-4	50-4#	
NOD207	50-5#		
NOD21	37-36	37-36#	
NOD210	50-6	50-6	50-6#
NOD211	50-7#		
NOD212	50-8	50-8	50-8#
NOD213	50-9#		
NOD214	50-10	50-10	50-10#
NOD215	50-11	50-11#	
NOD216	50-12	50-12	50-12#
NOD217	50-13#		
NOD22	37-37	37-37	37-37#
NOD220	50-14	50-14	50-14#
NOD221	50-15	50-15#	
NOD222	50-16	50-16#	
NOD223	50-17	50-17#	
NOD224	50-18	50-18	50-18#
NOD225	50-19#		
NOD226	50-20	50-20	50-20#
NOD227	50-21#		
NOD23	37-38	37-38#	
NOD230	50-22	50-22	50-22#
NOD231	50-23	50-23#	
NOD232	50-24	50-24#	
NOD233	50-25#		
NOD234	50-26#		
NOD235	50-27#		
NOD24	37-39	37-39	37-39#
NOD25	37-40	37-40#	
NOD26	37-41#		
NOD27	37-45	37-45#	
NOD3	37-22	37-22	37-22#
NOD30	37-46	37-46	37-46#
NOD31	37-47	37-47#	
NOD32	37-48	37-48	37-48#

























	47-146	47-154	47-162	47-171	47-188	47-203	47-209	49-5	49-32	49-43	51-30	51-34	51-36	52-30
	55-17	55-25	55-26	73-33	73-71	74-138	74-176	75-41	77-37	77-40	77-42	77-44	77-47	77-52
	77-56	77-62	77-71	77-76	77-80	77-96	77-153	77-165	77-166	77-168	77-169	77-185	78-19	79-26
	79-27	79-43	80-35	81-36	82-42	82-83	82-91	82-95	82-103	82-111	82-115	82-130	82-144	82-153
	82-173	82-184	84-12	84-76	85-20	85-45	87-9	87-53	88-29	88-51	88-52	88-66	88-67	88-69
	88-81	88-85	88-89	88-93	88-98	88-112	88-122	88-145	88-165	88-174	89-10	89-18	89-23	89-29
	89-37	89-38	89-55	95-61	95-75	95-84	97-41	97-63	97-67	97-138	98-27	99-24	99-42	100-96
	100-106	100-115	100-127	100-139	103-97	103-101	103-119	103-141	106-40	106-43	107-114	107-155	107-158	107-177
	107-184	107-197	107-208	107-212	107-217	109-15								
TSTSTS	21-12*	82-23*												
T1	22-8	82-23*												
TABEX	40-111*	82-153	82-184											
TAL	26-20*	86-21	99-47											
TALCK	32-32	98-20*	98-49	98-51										
TALMOD	26-151*	37-61												
TCURAD	31-14*	82-59*	82-62	82-159	82-164*	84-54*								
TEMP	31-46*	46-40*	46-45*	46-50*	46-54*	46-58*	46-63*	46-72*	46-76*	46-80*	46-86*	46-91*	46-105	46-121
	46-122*	46-123*	46-124	51-33*	51-34	52-32*	52-33*	52-34	53-35*	53-36*	55-42	55-23*	55-25	55-26
	55-27	58-26*	59-13*	59-16*	61-21	64-19*	73-24*	73-33	73-36*	73-39*	73-71	79-23*	84-75*	84-76
	85-30*	85-31*	85-35	85-38	88-31*	88-46*	88-51	88-52	88-53	88-66	88-66	88-67	88-68	88-70*
	88-88*	88-89	88-128*	88-138*	88-142*	88-143	88-148	89-49*	89-82*	89-88*	94-87*	94-88*	94-92	97-130*
	97-136*	97-138	100-196*	100-199*	100-222*	103-123*	103-124	103-148*	103-149	105-18	107-157*	107-158	107-158	107-160*
TEMP1	107-165	107-168	107-171	107-174	107-177	107-240*								
	31-47*	46-39*	46-44*	46-49*	46-57*	46-62*	46-71*	46-75*	46-79*	46-85*	46-90*	46-118	73-31*	73-33
	73-40*	73-43*	73-71	107-161*	107-167*	107-177								
TEMP2	31-48*	46-64*	46-65*	46-127	53-38*	53-40*	53-44	58-24*	58-31	61-21*	64-20*	64-22	72-14*	73-32*
	73-33	73-44*	73-47*	73-71	88-30*	88-45	88-129*	88-131	89-50*	89-53	89-95*	89-98	94-55*	94-67*
	94-75	94-81	94-90*	94-91	94-114*	95-46*	96-22*	97-76*	97-83*	97-97*	97-101*	97-113*	98-36*	98-45*
	99-26*	99-37*	100-91*	100-110*	100-121*	100-223*	100-227	103-91*	103-114*	103-136*	106-35*	107-111*	107-162*	107-170*
TEMP3	107-177													
	31-49*	43-37	43-55	43-64	46-66*	46-128	72-16*	73-27*	73-30*	73-33	88-68*	88-69	88-70	94-57*
	94-68*	94-76	94-88	94-89	94-90	94-115*	95-48*	95-58	96-23*	97-38*	97-41	97-44	97-45	97-46
	97-47	97-78*	97-85*	97-95*	97-102*	97-114*	97-125*	98-37*	98-46*	99-28*	99-38*	100-92*	100-111*	100-122*
TEMP4	103-92*	103-115*	103-137*	106-37*	107-110*	107-163*	107-173*	107-177						
	31-50*	43-33	43-55	43-64	46-67*	46-102*	46-129	95-55*	95-64*	95-69*	95-70	95-81	96-21*	97-96*
TEMP5	100-93*	100-112*	100-123*	103-93*	103-116*	103-139*	106-36*	107-109*	107-164*	107-176*	107-177			
	31-51*	46-81*	46-95*	46-130	47-122*	47-136*	47-141*	47-146	47-175*	47-208*	47-209	68-17*	68-25	69-15*
	69-25	73-66*	73-69*	73-71	96-20*									
TIMER1	32-47*	45-45	45-47*	82-37*	82-38	100-83*	100-89	100-107	106-28*	106-31*	106-32			
TIMER2	32-48*	45-48	45-50*											
TIMERS	32-49*	45-51	45-55*	100-185*	100-186	103-82*	103-89	105-44*	105-45					
TIMMIN	32-43*	45-42*	46-126	77-81*										
TIMSEC	32-44*	45-39*	45-40	45-43*	46-125	77-82*								
TIMTCK	32-45*	45-36*	45-38*	45-53	46-123	77-83*								
TM	26-213*	34-9												
TOIN1	103-90	103-101*												
TOIN2	103-102*													
TOINOT	103-83*	103-103	103-109											
TOOR1	106-33	106-43*												
TOOR2	106-46	106-49*												
TOOR3	106-30	106-32*	106-50											
TOORIO	47-60	47-68	100-158	100-212	100-233	101-41	102-46	103-70	104-21	105-16	105-56	106-27*	106-41	
TOTCC	31-44*	52-24*	52-25	52-31*	52-33	52-35*	82-49*	82-141*	82-142	82-163	82-170*	82-171	82-196	
TRA	26-16*	86-18												
TRAMOD	26-150*	37-59												
TRBB	26-67*	55-29	77-149	88-36	88-54	89-27								





TSS6A	35 8	41-90												
TSS7A	35-9	41-100												
TSSA	35-740	47-43*	47-46*	47-50*	47-54*	47-55*	47-58*	47-63	47-65	104-17*	104-19*	104-24	104-26	104 28*
TSSE	35-730	47-47*	47-51*	47-56*	47-65	100-194*	100-227*	104-23	104-29*					
TSSFLG	31 340	47-61*	107-133*	107-256										
TSSIND	35-400	88-40	107-146											
TSSKEY	35-750	47 48	47-52	47-54	47-56	49-15*	49-21*	49-38*	49-46*	49-47*				
TSSLST	35-20	88-47	107-148											
TTL	26-260	82-96	89-21	89-25										
TILLOP	26-1580	37-188												
TTOTCC	31-130	54-93	82-50*	82-141	82-146	82-163*	84-50*							
TXBUF	29-30	54-61	82-59	84-53	97-96									
TXC	26-720	46-45												
TXMTUT	31-120	54-81	60-29	82-65*	82-148*	82-150	82-165*	84-45	84-49*	89-8	93-23			
TXNC	41-1590	97-94												
TXON2	91-180													
TXONLY	32-28	91-170												
TXPTR	31 30	54-74*	54-76*	54-77	54-86*	54-88	82-52*	82-63	82-149*	82-157*	82-158	82-162*	84-51*	84 52
	89-57*													
TXQ	26-710	46-40												
TXSTAK	29-300	89-70												
TXTHEM	38 44	41-1480												
UAM	25-00													
ULRCLS	65-350	94-104												
ULRPLS	56-30	63-330	68-22	94-48	94-85									
ULTCLS	67-350	70-18	94-118											
ULTPLS	66-340	70-21	94-73											
UNKM	38-101	40-2260	97-130											
UPTA1	94-78	94-850												
UPTA4	94-72	94-770												
UPTABL	94-710													
UPTEX	94-83	94-960												
VALTRB	28-620	88-26	88-102*	88-161*										
VECTOR	112-49	112-770												
WRDESA	71-220	71-24												
WRDESB	71-210	71-26												
WRDESD	71-320	71-34												
WRDEFP	71-150	82-81	88-20											
WRFLG	28-600	55-16*	82-136	88-28*										
WRIP1	104-18	104-200	104-30											
WRIPEX	104-27	104-310												
WRIPP	100-228	104-190												
WRIPPG	100-195	104-170												
WRMCS	79-24	100-202	105-150	107-241										
X1	21-360	37-19	37-19	37-190	37-20	37-20	37-200	37-21	37-21	37-210	37-22	37-22	37-220	37-23
	37-23	37-230	37-24	37-24	37-240	37-25	37-25	37-250	37-26	37-26	37 260	37-27	37-27	37-270
	37-28	37-28	37-280	37-29	37-29	37-290	37-30	37-30	37-300	37 31	37-31	37-310	37-32	37-32
	37-320	37-33	37-33	37-330	37-34	37-34	37-340	37-35	37-35	37-350	37-36	37 36	37-360	37-37
	37 37	37-370	37 38	37-38	37-380	37-39	37-39	37-390	37-40	37-40	37-400	37-41	37-41	37-410
	37-45	37-45	37-450	37-46	37-46	37-460	37-47	37-47	37-470	37-48	37-48	37-480	37-49	37-49
	37-490	37-50	37-50	37-500	37-51	37-51	37-510	37-52	37-52	37-520	37-53	37-53	37-530	37-54
	37-54	37-540	37-55	37-55	37-550	37-56	37-56	37-560	37-57	37-57	37-570	37-58	37-58	37-580
	37-59	37-59	37-590	37-60	37-60	37-600	37-61	37-61	37-610	37-62	37-62	37 620	37-66	37-66
	37-660	37-67	37-67	37-670	37-68	37-68	37-680	37-69	37-69	37-690	37-70	37-70	37-700	37-75
	37-75	37-750	37-76	37-76	37-760	37-77	37-77	37-770	37-78	37-78	37-780	37-79	37-79	37-790
	37-82	37-82	37 820	37-83	37-83	37-830	37-84	37 84	37-840	37-85	37-85	37-850	37-86	37-86

37-86	37-87	37-87	37-87	37-88	37 88	37-88	37-89	37-89	37-89	37-92	37 92	37-92	37-93
37-93	37-93	37-94	37 94	37-94	37-95	37-95	37-95	37 96	37-96	37-96	37-97	37-97	37-97
37 107	37-107	37-107	37-108	37-108	37-108	37-109	37-109	37-109	37-110	37-110	37-110	37 124	37 124
37-124	37-125	37-125	37-125	37-127	37-127	37-127	37-128	37-128	37-128	37-130	37-130	37 130	37-131
37-131	37-131	37-134	37-134	37-134	37-137	37-137	37-137	37-138	37-138	37-138	37-139	37-139	37 139
37-140	37-140	37-140	37-141	37-141	37-141	37-142	37-142	37-142	37-143	37-143	37-143	37-144	37 144
37-144	37-145	37-145	37-145	37-146	37 146	37-146	37-147	37-147	37-147	37-148	37-148	37 148	37-149
37-149	37-149	37-150	37-150	37-150	37-151	37-151	37-151	37-152	37 152	37-152	37-155	37-155	37-155
37-156	37-156	37-156	37-157	37-157	37-157	37-158	37-158	37-158	37-159	37-159	37-159	37-160	37-160
37-160	37-161	37-161	37-161	37-164	37-164	37-164	37-165	37-165	37-165	37-166	37-166	37-166	37-167
37-167	37-167	37-168	37-168	37-168	37-169	37-169	37-169	37-172	37-172	37-172	37-173	37-173	37 173
37-174	37-174	37-174	37-177	37-177	37-177	37-188	37-188	37-188	37-189	37-189	37-189	37-190	37-190
37 190	37-191	37-191	37-191	37-192	37-192	37-192	37-193	37-193	37-193	37-194	37-194	37-194	37-195
37 195	37-195	37-198	37-198	37-198	37-199	37-199	37-199	37-200	37-200	37-200	37-202	37-202	37-202
37-203	37-203	37-203	37-204	37-204	37-204	37-205	37-205	37-205	37-206	37-206	37 206	37-207	37 207
37-207	37-208	37-208	37-208	37-209	37-209	37-209	37-210	37-210	37-210	37-211	37-211	37-211	37-212
37-212	37-212	37-213	37-213	37-213	37-216	37-216	37-216	50-3	50-3	50-3	50-4	50-4	50-4
50-5	50-5	50-5	50-6	50-6	50-6	50-7	50-7	50-7	50-8	50-8	50-8	50-9	50-9
50-9	50-10	50-10	50-10	50-11	50-11	50-11	50-12	50-12	50-12	50 13	50-13	50-13	50 14
50-14	50-14	50-15	50-15	50-15	50-16	50-16	50-16	50-17	50-17	50-17	50-18	50 18	50 18
50-19	50-19	50-19	50-20	50-20	50-20	50-21	50-21	50-21	50-22	50-22	50-22	50-23	50 23
50-23	50-24	50-24	50-24	50-25	50-25	50-25	50-26	50 26	50-26	50 27	50-27	50-27	50-27

X\$ALWA 21-12  
 X\$FALS 21-12 112-53  
 X\$OFFS 21-12 112-53  
 X\$TRUE 21-12



CZCLMCO DMP/V 11 DCLT MACRO V05.00 Thursday 22 Mar 84 16:24 Page M 2  
 Cross reference table (CREF V05.00)

ENDHW	1 465#	21-12#	23-59											
ENDINI	1 475#	21-12#	77-185											
ENDMOD	1-487#	21-12#	113-44											
ENDMSG	1-500#	21-12#	43-30	43-34	43-38	43-56	43 65	43-70	43-74					
ENDPRO	1-512#	21-12#	76-14											
ENDPTA	1-520#	21-12#												
ENDRPT	1-529#	21-12#	75-41											
ENDSEG	1-541#	21-12#												
ENDSET	1-555#	21-12#												
ENDSFT	1-568#	21-12#												
ENDSRV	1-580#	21-12#	45-57	108-31	108-50									
ENDSUB	1-596#	21-12#												
ENDSW	1-614#	21-12#												
ENDTST	1-624#	21-12#	109-15											
EQUALS	1-642#	21-12#	24-77											
ERRDF	1-714#	21-12#												
ERRHRD	1-718#	21-12#												
ERROR	1-722#	21-12#												
ERRSF	1-726#	21-12#												
ERRSOF	1-730#	21-12#	95-61	95-75	95-84	97-67	100-96	100-115	100-127	103-97	103 119	103-141	106 40	107-114
	107-197	107-212												
ERRTBL	1 734#	21-12#												
ESCAPE	1-744#	21-12#												
EXIT	1-771#	21-12#	43-76	77-169	79-27	80-19	81 20	82-95	82 130					
FEQUAL	1-810#	21-12#												
GETBYT	1-824#	21-12#												
GETPRI	1-834#	21-12#												
GETWOR	1-829#	21-12#												
GMANIA	1-839#	21-12#												
GMANID	1-848#	21-12#	47-16	55-26	77-76	82-103	88-52	88 67	88-69	97-41	98-27			
GMANIL	1-859#	21-12#												
GPHARD	1 868#	21-12#	77 96											
GPRMA	1-874#	21-12#	112-48	112-49										
GPRMD	1 903#	21-12#	47-16	47-16#	55-26	55-26#	77-76	77-76#	82-103	82-103#	88-52	88-52#	88 67	88-67#
	88-69	88-69#	97-41	97-41#	98-27	98-27#	112-50	112-51						
GPRML	1-934#	21-12#	112-27	112-52	112-54									
HEADER	1 954#	21-12#	21-98											
INLOOP	1-962#	21-12#												
IOSETU	1-966#	21-12#												
IOSTAR	1-974#	21-12#												
KT11	1 982#	21-12#												
LASTAD	1-;47#	21-12#	113-48											
M\$BYTE	1-000#	21-12#	21-98	21-98	21-98	21-98#								
M\$CHEC	1-E18#	21-12#	43-76	43-76#	77-169	77-169#	79-27	79-27#	80-19	80-19#	81-20	81-20#	82 95	82-95#
	82-130	82-130#												
M\$CNTD	1 E82#	21-12#	47-16	47-16#	55-26	55-26#	77-76	77-76#	82-103	82-103#	88 52	88 52#	88-67	88 67#
	88-69	88-69#	97-41	97-41#	98-27	98-27#	112-27	112-27#	112-48	112-48#	112-49	112-49#	112-50	112-50#
	112-51	112-51#	112-52	112-52#	112-54	112-54#								
M\$COUN	1 066#	21-12#	43-29	43-29	43-29	43-29#	43-33	43-33#	43-37	43-37	43-37#	43 55	43-55	43 55#
	43-64	43-64	43-64	43-64#	43-69	43-69	43-69	43-69#	43-73	43-73	43-73#	43-73#	46-115	46 115#
	46-118	46-118#	47-10	47-10#	47-24	47-24	47-28	47-28#	47-85	47-85#	47-103	47 103#	47-108	47 108#
	47-108	47-108	47-108#	47 115	47-115	47-115#	47-123	47-123#	47-124	47-124	47-124#	47-145	47-145	47-145#
	47-146	47-146	47-146#	47-154	47-154	47-154#	47-154#	47-162	47 162	47-162	47-162#	47 171	47-171#	47-188
	47-188#	47-203	47-203#	47-209	47-209#	49-5	49-5#	49-32	49-32#	49-43	49-43#	51-30	51-30#	51-34
	51-34#	51-36	51-36#	52-30	52-30#	55-17	55-17#	55-25	55-25#	73-33	73 33	73-33	73-33	73 33#
	73 71	73-71	73-71	73-71	73-71#	74-138	74-138#	74-176	74-176#	77 80	77-80#	82-42	82-42#	82-83









CZCLMCO DMP/V-11 DCLT MACRO V05.00 Thursday 22-Mar-84 16:24 Page M-6  
 Cross reference table (CREF V05.00)

M\$GNSU	1 898#	21-12#												
M\$GNTA	1-890#	21-12#	23-59	23-59#	43-30	43-30#	43-34	43-34#	43-38	43-38#	43-56	43-56#	43-65	43-65#
	43-70	43-70#	43-74	43-74#	45-57	45-57#	75-41	75-41#	77-185	77-185#	78-19	78-19#	79-43	79-43#
	80-35	80-35#	81-36	81-36#	108-31	108-31#	108-50	108-50#	109-15	109-15#	112-56	112-56#		
M\$GNTE	1-894#	21-12#	82-23	82-23#										
M\$HAPT	1 A39#	21-12#	21-98	21-98#										
M\$HNAP	1-824#	21-12#	21-98	21-98#										
M\$INCR	1-026#	21-12#	21-33	21-33#	23-10	23-10	23-10#	23-10#	43-28	43-28	43-28#	43-28#	43-29#	43-30#
	43-32	43-32#	43-32#	43-32#	43-33#	43-34#	43-36	43-36#	43-36#	43-36#	43-37#	43-38#	43-54	43-54#
	43-54#	43-54#	43-55#	43-56#	43-63	43-63	43-63#	43-63#	43-64#	43-65#	43-68	43-68	43-68#	43-68#
	43-69#	43-70#	43-72	43-72	43-72#	43-72#	43-73#	43-74#	45-33	45-33	45-33#	45-33#	46-115#	46-118#
	47-10#	47-16	47-16#	47-16#	47-24#	47-28#	47-85#	47-103#	47-108#	47-115#	47-123#	47-123#	47-145#	47-146#
	47-154#	47-162#	47-171#	47-188#	47-203#	47-209#	49-5#	49-32#	49-43#	51-30#	51-34#	51-36#	52-30#	55-17#
	55-25#	55-26	55-26#	55-26#	73-33#	73-71#	74-138#	74-176#	75-9	75-9	75-9#	75-9#	75-41#	76-8
	76-8	76-8#	76-8#	77-8	77-8	77-8#	77-8#	77-37#	77-40#	77-42#	77-44#	77-47#	77-52#	77-56#
	77-62#	77-71#	77-76	77-76#	77-76#	77-80#	77-96#	77-153#	77-165#	77-166#	77-168#	77-169#	77-185#	78-10
	78-10	78-10#	78-10#	78-19#	79-8	79-8	79-8#	79-8#	79-26#	79-27#	79-43#	80-8	80-8	80-8#
	80-8#	80-35#	81-9	81-9	81-9#	81-9#	81-36#	82-23	82-23	82-23	82-23#	82-23#	82-23#	82-42#
	82-83#	82-91#	82-95#	82-103	82-103#	82-103#	82-111#	82-115#	82-130#	82-144#	82-153#	82-173#	82-184#	84-12#
	84-76#	85-20#	85-45#	87-9#	87-53#	88-29#	88-51#	88-52	88-52#	88-52#	88-66#	88-67	88-67#	88-67#
	88-69	88-69#	88-69#	88-81#	88-85#	88-89#	88-93#	88-98#	88-112#	88-122#	88-145#	88-165#	88-174#	89-10#
	89-18#	89-23#	89-29#	89-37#	89-38#	89-55#	95-61#	95-75#	95-84#	97-41	97-41#	97-41#	97-63#	97-67#
	97-138#	98-27	98-27#	98-27#	99-24#	99-42#	100-96#	100-106#	100-115#	100-127#	100-139#	103-97#	103-101#	103-119#
	103-141#	106-40#	106-43#	107-114#	107-155#	107-158#	107-177#	107-184#	107-197#	107-208#	107-212#	107-217#	108-22	108-22
	108-22#	108-22#	108-42	108-42	108-42#	108-42#	109-15#	112-13	112-13	112-13#	112-13#			
M\$IOSE	1-A00#	21-12#												
M\$LDRO	1-C42#	21-12#	77-40	77-40#	77-42	77-42#	77-44	77-44#	77-47	77-47#	77-56	77-56#	77-62	77-62#
	77-96	77-96#	77-168	77-168#	79-26	79-26#								
M\$MASK	1-871#	21-12#												
M\$MCHI	1-4#	21-12	21-12#	21-12#										
M\$MCLO	1-824#	21-12	21-12#	21-12#										
M\$MSK1	1-877#	21-12#												
M\$POP	1-881#	21-12#	23-59	23-59#	43-30	43-30#	43-34	43-34#	43-38	43-38#	43-56	43-56#	43-65	43-65#
	43-70	43-70#	43-74	43-74#	45-57	45-57#	75-41	75-41#	76-14	76-14#	77-185	77-185#	78-19	78-19#
	79-43	79-43#	80-35	80-35#	81-36	81-36#	108-31	108-31#	108-50	108-50#	109-15	109-15#	112-56	112-56#
	113-49	113-49#												
M\$PRIN	1-836#	21-12#	43-29	43-29#	43-33	43-33#	43-37	43-37#	43-55	43-55#	43-64	43-64#	43-69	43-69#
	43-73	43-73#	46-115	46-115#	46-118	46-118#	47-10	47-10#	47-24	47-24#	47-28	47-28#	47-85	47-85#
	47-103	47-103#	47-108	47-108#	47-115	47-115#	47-123	47-123#	47-124	47-124#	47-145	47-145#	47-146	47-146#
	47-154	47-154#	47-162	47-162#	47-171	47-171#	47-188	47-188#	47-203	47-203#	47-209	47-209#	49-5	49-5#
	49-32	49-32#	49-43	49-43#	51-30	51-30#	51-34	51-34#	51-36	51-36#	52-30	52-30#	55-17	55-17#
	55-25	55-25#	73-33	73-33#	73-71	73-71#	74-138	74-138#	74-176	74-176#	77-80	77-80#	82-42	82-42#
	82-83	82-83#	82-111	82-111#	82-115	82-115#	82-144	82-144#	82-153	82-153#	82-173	82-173#	82-184	82-184#
	84-12	84-12#	84-76	84-76#	85-20	85-20#	85-45	85-45#	87-9	87-9#	87-53	87-53#	88-29	88-29#
	88-51	88-51#	88-66	88-66#	88-81	88-81#	88-85	88-85#	88-89	88-89#	88-93	88-93#	88-98	88-98#
	88-112	88-112#	88-122	88-122#	88-145	88-145#	88-165	88-165#	88-174	88-174#	89-10	89-10#	89-18	89-18#
	89-23	89-23#	89-29	89-29#	89-37	89-37#	89-38	89-38#	89-55	89-55#	97-63	97-63#	97-138	97-138#
	99-24	99-24#	99-42	99-42#	107-155	107-155#	107-158	107-158#	107-177	107-177#				
M\$PUSH	1 831#	21-12#	21-33	21-33#	23-10	23-10#	43-28	43-28#	43-32	43-32#	43-36	43-36#	43-54	43-54#
	43-63	43-63#	43-68	43-68#	43-72	43-72#	45-33	45-33#	75-9	75-9#	76-8	76-8#	77-8	77-8#
	78-10	78-10#	79-8	79-8#	80-8	80-8#	81-9	81-9#	82-23	82-23#	108-22	108-22#	108-42	108-42#
	112-13	112-13#												
M\$PUT	1-C72#	21-12#	43-29	43-29	43-29	43-29	43-29	43-29#	43-33	43-33	43-33	43-33#	43-37	43-37
	43-37	43-37	43-37#	43-55	43-55	43-55	43-55	43-55#	43-64	43-64	43-64	43-64#	43-64	43-64#
	43-69	43-69	43-69	43-69	43-69	43-69#	43-73	43-73	43-73	43-73	43-73	43-73#	46-115	46-115
	46-115#	46-118	46-118	46-118#	47-10	47-10	47-10#	47-10#	47-24	47-24	47-24#	47-28	47-28#	47-85

47-85	47-85	47-103	47-103	47-103	47-108	47-108	47-108	47-108	47-108	47-108	47-108	47-115	47-115
47-115	47-115	47-115	47-123	47-123	47-123	47-123	47-124	47-124	47-124	47-124	47-124	47-145	47-145
47-145	47-145	47-145	47-146	47-146	47-146	47-146	47-146	47-146	47-154	47-154	47-154	47-154	47-154
47-162	47-162	47-162	47-162	47-162	47-162	47-162	47-171	47-171	47-171	47-171	47-188	47-188	47-188
47-203	47-209	47-209	47-209	47-209	47-209	47-209	49-5	49-5	49-5	49-5	49-32	49-32	49-32
49-43	49-43	49-43	51-30	51-30	51-30	51-30	51-34	51-34	51-34	51-34	51-34	51-36	51-36
51-36	52-30	52-30	52-30	52-30	55-17	55-17	55-17	55-25	55-25	55-25	55-25	73-33	73-33
73-33	73-33	73-33	73-33	73-33	73-71	73-71	73-71	73-71	73-71	73-71	73-71	73-71	73-71
74-176	74-176	74-176	77-80	77-80	77-80	77-80	77-153	77-153	77-153	77-153	77-153	77-153	77-153
77-165	77-165	77-166	77-166	77-166	77-166	77-166	82-42	82-42	82-42	82-42	82-83	82-83	82-83
82-111	82-111	82-115	82-115	82-115	82-144	82-144	82-144	82-144	82-153	82-153	82-153	82-153	82-153
82-173	82-173	82-173	82-184	82-184	82-184	82-184	84-12	84-12	84-12	84-12	84-12	84-76	84-76
84-76	84-76	85-20	85-20	85-20	85-45	85-45	85-45	85-45	87-9	87-9	87-9	87-53	87-53
88-29	88-29	88-29	88-29	88-29	88-51	88-51	88-51	88-51	88-66	88-66	88-66	88-66	88-66
88-81	88-81	88-85	88-85	88-85	88-85	88-85	88-89	88-89	88-89	88-89	88-93	88-93	88-93
88-98	88-112	88-112	88-112	88-112	88-122	88-122	88-122	88-122	88-122	88-122	88-145	88-145	88-145
88-165	88-165	88-165	88-174	88-174	88-174	88-174	89-10	89-10	89-10	89-10	89-18	89-18	89-18
89-23	89-29	89-29	89-29	89-29	89-37	89-37	89-37	89-37	89-38	89-38	89-38	89-55	89-55
97-63	97-63	97-138	97-138	97-138	97-138	97-138	97-138	97-138	99-24	99-24	99-24	99-42	99-42
107-155	107-155	107-155	107-158	107-158	107-158	107-158	107-158	107-158	107-177	107-177	107-177	107-177	107-177
107-177	107-177	107-177	107-177	107-177	107-177	107-177	107-177	107-177	107-177	107-177	107-177	107-177	107-177
MSPUT1	1-C81	21-12	43-29	43-29	43-29	43-29	43-29	43-29	43-29	43-29	43-29	43-29	43-29
	43-33	43-33	43-33	43-33	43-37	43-37	43-37	43-37	43-37	43-37	43-37	43-37	43-37
	43-55	43-55	43-55	43-55	43-55	43-55	43-55	43-55	43-64	43-64	43-64	43-64	43-64
	43-64	43-64	43-69	43-69	43-69	43-69	43-69	43-69	43-69	43-69	43-69	43-69	43-69
	43-73	43-73	43-73	43-73	43-73	43-73	43-73	43-73	46-115	46-115	46-115	46-115	46-115
	46-118	46-118	47-10	47-10	47-10	47-10	47-24	47-24	47-24	47-24	47-24	47-28	47-28
	47-85	47-85	47-85	47-85	47-103	47-103	47-103	47-103	47-108	47-108	47-108	47-108	47-108
	47-108	47-108	47-108	47-108	47-108	47-108	47-115	47-115	47-115	47-115	47-115	47-115	47-115
	47-123	47-123	47-123	47-123	47-123	47-123	47-124	47-124	47-124	47-124	47-124	47-124	47-124
	47-145	47-145	47-145	47-145	47-145	47-145	47-145	47-145	47-146	47-146	47-146	47-146	47-146
	47-146	47-146	47-154	47-154	47-154	47-154	47-154	47-154	47-154	47-154	47-154	47-154	47-154
	47-162	47-162	47-162	47-162	47-162	47-162	47-162	47-162	47-171	47-171	47-171	47-171	47-171
	47-188	47-188	47-203	47-203	47-203	47-203	47-203	47-203	47-209	47-209	47-209	47-209	47-209
	49-5	49-5	49-5	49-5	49-32	49-32	49-32	49-32	49-32	49-32	49-32	49-43	49-43
	49-43	49-43	51-30	51-30	51-30	51-30	51-30	51-30	51-34	51-34	51-34	51-34	51-34
	51-36	51-36	51-36	51-36	51-36	51-36	52-30	52-30	52-30	52-30	52-30	55-17	55-17
	55-25	55-25	55-25	55-25	55-25	55-25	73-33	73-33	73-33	73-33	73-33	73-33	73-33
	73-33	73-33	73-33	73-33	73-71	73-71	73-71	73-71	73-71	73-71	73-71	73-71	73-71
	73-71	73-71	74-138	74-138	74-138	74-138	74-138	74-138	74-176	74-176	74-176	74-176	74-176
	77-153	77-153	77-153	77-153	77-153	77-153	77-153	77-153	77-153	77-153	77-153	77-153	77-153
	77-165	77-165	77-166	77-166	77-166	77-166	77-166	77-166	77-166	77-166	77-166	77-166	77-166
	82-83	82-83	82-83	82-83	82-111	82-111	82-111	82-111	82-115	82-115	82-115	82-115	82-115
	82-144	82-144	82-144	82-144	82-153	82-153	82-153	82-153	82-153	82-153	82-153	82-153	82-153
	82-173	82-173	82-184	82-184	82-184	82-184	82-184	82-184	84-12	84-12	84-12	84-12	84-12
	84-76	84-76	84-76	84-76	84-76	84-76	84-76	84-76	85-20	85-20	85-20	85-20	85-20
	85-45	85-45	87-9	87-9	87-9	87-9	87-9	87-9	87-53	87-53	87-53	87-53	87-53
	88-29	88-29	88-51	88-51	88-51	88-51	88-51	88-51	88-66	88-66	88-66	88-66	88-66
	88-66	88-66	88-81	88-81	88-81	88-81	88-81	88-81	88-85	88-85	88-85	88-85	88-85
	88-89	88-89	88-93	88-93	88-93	88-93	88-93	88-93	88-98	88-98	88-98	88-98	88-98
	88-112	88-112	88-122	88-122	88-122	88-122	88-122	88-122	88-122	88-122	88-122	88-122	88-122
	88-165	88-165	88-165	88-165	88-165	88-165	88-165	88-165	88-174	88-174	88-174	88-174	88-174
	89-18	89-18	89-18	89-18	89-23	89-23	89-23	89-23	89-29	89-29	89-29	89-29	89-29
	89-37	89-37	89-38	89-38	89-38	89-38	89-38	89-38	89-55	89-55	89-55	89-55	89-55
	97-138	97-138	97-138	97-138	97-138	97-138	97-138	97-138	99-24	99-24	99-24	99-24	99-24
	99-42	99-42	107-155	107-155	107-155	107-155	107-155	107-155	107-158	107-158	107-158	107-158	107-158



