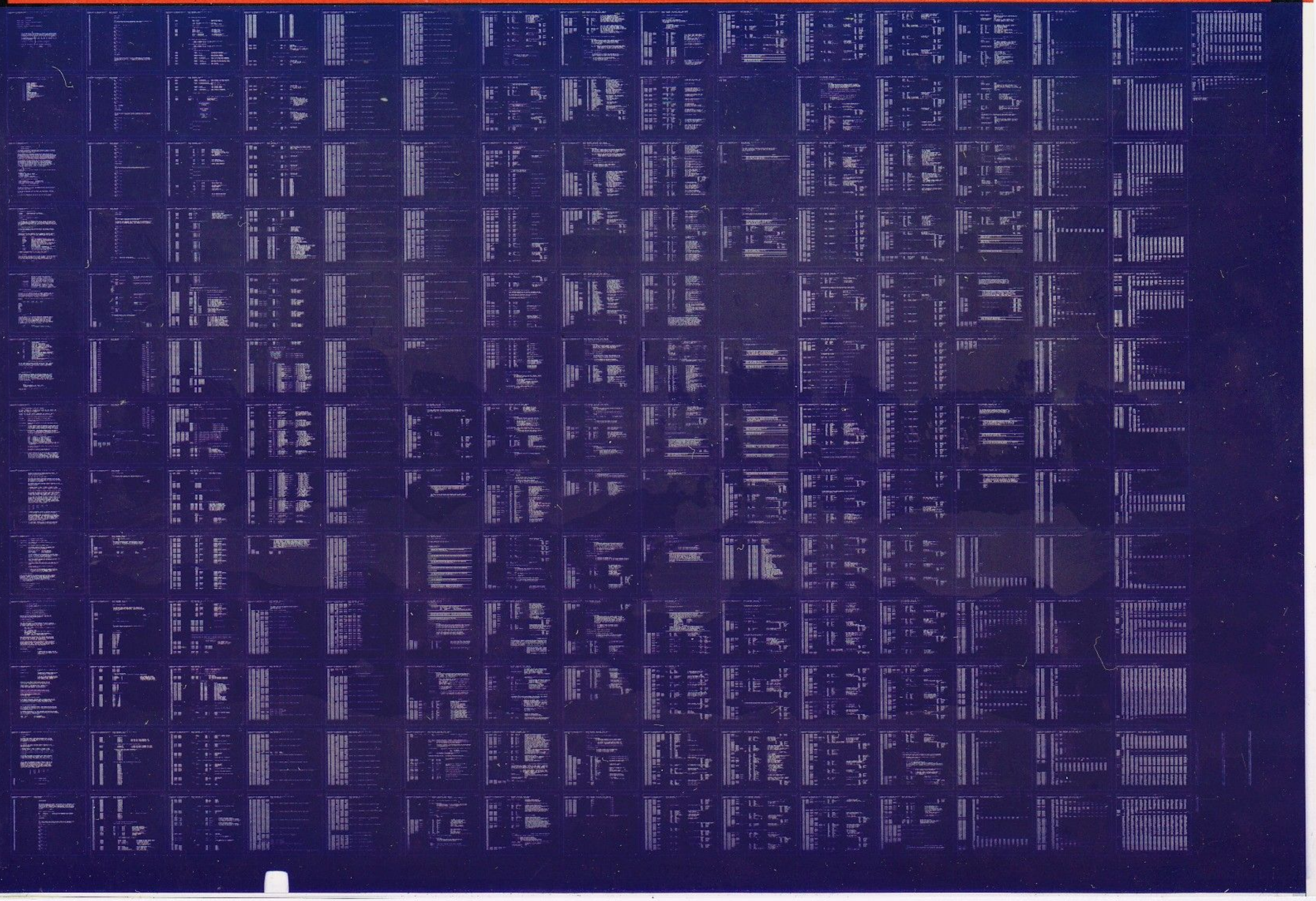
DEUNA NI EXERCISER DIAG AH-T228A-MC
CZUACA0 FICHE 1 OF 1



.REM &

IDENTIFICATION

PRODUCT CODE: AC-T227A-MC

PRODUCT NAME: CZUACA0 NI EXERCISER DIAGNOSTIC

PRODUCT DATE: 6-APR-83

MAINTAINER: MERRIMACK DIAGNOSTIC ENGINEERING

AUTHOR: GARY MCCOY

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THE NETWORK INTERCONNECT EXERCISER (NIE) PROGRAM IS MEANT TO PROVIDE FIELD SERVICE WITH A TOOL FOR DETERMININING THE CONNECTIVITY OF NODES ON THE NETWORK INTERCONNECT (NI).

THE NIE PROGRAM WILL DETERMINE THE ABILITY OF NODES ON THE NI TO COMMUNICATE WITH EACH OTHER AND PROVIDE NODE INSTALLATION VERIFICATION AND PROBLEM ISOLATION. THE NIE USES THE LOW LEVEL MAINTENANCE FEATURES OF THE DEUNA TO PROVIDE TESTING WITHOUT INTERUPTING NORMAL OPERATION OF THE NI. THE VAX VERSION OF THE NIE CAN ALSO BE RUN CONNCURRENTLY ON ANOTHER NODE, WITH EACH VERSION RUNNING INDEPENDENTLY OF EACH OTHER.

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. THERE IS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES IN SECTION 2 OF THIS DOCUMENT.

1.2 SYSTEM REQUIREMENTS

IN ORDER TO RUN THE CZUAC NIE PROGRAM, THE FOLLOWING MINIMUM HARDWARE IS REQUIRED:

- A PDP-11 CPU
- MINUMUM OF 24K WORDS OF MEMORY
- A WORKING, LINE OR REAL-TIME CLOCK
- A CONSOLE TERMINAL
- ANY XXDP+ SUPPORTED LOAD MEDIA
- DEUNA-11 UNIBUS TO ETHERNET ADAPATER

1.3 RELATED DOCUMENTS AND STANDARDS

- DEUNA USER'S GUIDE EK-DEUNA-UG-001
- DEUNA TECHNICAL DESCRIPTION EK-DEUNA-TD-001
- XXDP+ USER'S MANUAL (CHQUS?.SEQ WHERE ? IS THE REV. LEVEL OF THE MANUAL - "C" IS THE CURRENT REV.)

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

THE GOAL OF THE NIE IS TO TEST THE COMMUNICATIONS LINK AND THEREFORE ASSUMS THAT THE CPU'S, CLOCKS, AND DEUNA'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.

IT IS NOT THE INTENTION OF THE NIE TO TEST THE DEVICE (DEUNA), BUT TO TEST THE COMMUNICATIONS LINK TO WHICH IT IS CONNECTED.

THE PREREQUISITE DIAGNOSTICS ARE:

- CZUAAA0 DEUNA REPAIR LEVEL DIAGNOSTIC
- CZUABA0 DEUNA FUNCTIONAL LEVEL DIAGNOSTIC

ALSO AVAILABLE FOR TEST IS:

- CXUACA0 DEUNA DEC/X-11 MODULE

1.5 ASSUMPTIONS

IT IS ASSUMED THAT THE COMMUNICATIONS DEVICE (DEUNA) HAS BEEN TESTED USING THE PREREQUISITE DIAGNOSTICS. THE OPERATOR SHOULD HAVE READ THE USER DOCUMENTATION PROTION OF THE LISTING TO FAMILIARIZE HIMSELF WITH THE COMMANDS AND CAPABILITIES AVAILABLE UNDER THE DIAGNOSTIC SUPERVISOR AND NIE.

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). YOU WILL THEN BE ASKED THE FOLLOWING QUESTIONS FOR EACH UNIT.

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

UNIT 0
DEVICE CSR ADDRESS : (O) 164524 ?<CR>
INTERRUPT VECTOR ADDRESS : (O) 120 ?<CR>
INTERRUPT PRIORITY : (O) 5 ?<CR>
```

WHEN YOU COMPLETE THE ABOVE SEQUENCE YOU WILL BE AT THE NIE> COMMAND LEVEL.

```
NIE> (A) ?
```

2.5 NETWORK INTERRCONNECT EXERCISER COMMANDS

THE "NIE>" COMMAND LEVEL FOLLOWS THE ATTACHING OF THE DEVICE AND ISSUING THE START TO THE SUPERVISOR. THESE COMMANDS CAN BE TYPED WHEN THE "NIE>" PROMPT IS PRINTED.

YOU ONLY HAVE TO TYPE ENOUGH CHARACTERS TO UNIQUELY SPECIFY A COMMAND. THE COMMAND LINE IS INTERPRETED FROM LEFT TO RIGHT.

HELP OR ? PRINTS OUT A BRIEF DESCRIPTION OF NIE COMMANDS.

SHOW NODES PRINTS OUT THE CONTENTS OF THE NODE TABLE.

SHOW MESSAGE PRINTS OUT THE CURRENT MESSAGE PARAMETERS FOR SIZE, TYPE AND COPIES.

SHOW COUNTERS PRINTS OUT THE CONTENTS OF THE HOST NODE DEUNA INTERNAL COUNTERS.

NODE ADR/TYPE THE NODE COMMAND ALLOWS THE OPERATOR TO ENTER NODES INTO THE NODE TABLE. NODES ARE SPECIFIED USING THEIR 12 HEX DIGIT ETHERNET PHYSICAL ADDRESS AND CAN BE SPECIFIED AS EITHER TARGET OR ASSIST (A DEFAULT OF TARGET IS ASSUMED).

MESSAGE/TYPE=/SIZE=N/COPIES=M THE MESSAGE COMMAND ALLOWS THE OPERATOR TO SELECT THE CURRENT MESSAGE PARAMETERS AS FOLLOWS. ANY OR ALL OF THE PARAMETER CAN BE CHANGED WITH THE COMMAND. THE DEFAULT PARAMETERS ARE TYPE=ALPHA,SIZE=512,COPIES=1.

TYPE ONE OF THE FOLLOWING MESSAGE TYPES:

ALPHA -- !"#$%'()*+,-./0123456789;:=?ABCDEFG ETC.
ONES -- MESSAGE OF ALL ONES (11111111....)
ZEROS -- MESSAGE OF ALL ZEROS (0000000....)
1ALT -- ALTERNATING 1'S AND 0'S (10101010...)
0ALT -- ALTERNATING 0'S AND 1'S (01010101...)
CCITT -- "CCITT" PSEUDO-RANDOM TEST PATTERN
OPERATOR SELECTED -- OPERATOR CHOSEN PATTERN OF LESS THAN 72 CHARACTERS USING 0-9, A-Z AND SPACES. (NOT USED IN PATTERN TEST)

SIZE THE SIZE OF THE MESSAGE BUFFER (DATA ONLY)MAY BE BETWEEN 32 AND 1466 BYTES.

COPIES THE NUMBER OF COPIES OF EACH MESSAGE SENT TO EACH NODE DURING A TEST MAY BE BETWEEN 1 AND 255.

RUN TEST/PASS=NN CAUSES EXECUTION OF THE SPECIFIED TEST FOR NN NUMBER OF PASSES. A DEFAULT VALUE OF 1 IS ASSUMED IF /PASS=NN IS NOT INCLUDED IN THE COMMAND LINE. A VALUE OF NN=-1 WILL CAUSE THE TEST TO BE RUN INDEFINATELY. NODE ADDRESSES FOR THE TESTS ARE TAKEN FROM THE NODE TABLE AND SHOULD BE ENTERED PRIOR TO RUNNING THE TEST USING THE NODE COMMAND. IN THE CASE OF THE LOOPPAIR TEST, NODE PAIRS ARE REQUIRED

AND MUST BE SPECIFIED AS TARGET AND ASSIST NODES. THE CURRENTLY SELECTED VALUES FOR MESSAGE TYPE, SIZE AND COPIES ARE USED BY EACH TEST.

THERE ARE FOUR TEST TO CHOSE FROM:

DIRECT — THE DIRECT TEST SENDS A LOOP DIRECT MESSAGE TO ALL OF THE NODES CONTAINED IN THE NODE TABLE AND WAITS FOR A RESPONSE. THE INTEGRITY OF THE RETURNED DATA IS CHECKED AND ANY ERRORS ARE REPORTED TO THE OPERATOR.

LOOPPAIR — THE LOOPPAIR TEST SENDS ASSISTED LOOPBACK MESSAGES TO THE NODE PAIRS CONTAINED IN THE NODE TABLE. THREE TYPE OF ASSISTED MESSAGES ARE SENT:

1) RECEIVE ASSIST -- HOST -> TARGET -> ASSIST -> HOST
2) TRANSMIT ASSIST -- HOST -> ASSIST -> TARGET -> HOST
3) FULL ASSIST -- HOST -> ASSIST -> TARGET -> ASSIST -> HOST

IN EACH CASE A RESPONSE IS WAITED FOR AND THE DATA IS CHECKED.

PATTERN — THE PATTERN TEST SENDS SIX DIFFERENT LOOP DIRECT MESSAGES TO EACH NODE CONTAINED IN THE NODE TABLE. EACH OF THE SIX PATTERN TYPES (ALPHA, ONES, ZEROS, 1ALT, 0ALT, CCITT) IS USED FOR EACH NODE. RETURNED DATA IS CHECKED FOR ERRORS.

ALL — THE ALL NODE TEST PERFORMS THE MOST EXTENSIVE CHECK OF THE NETWORK AND IS COMPOSED OF TWO PARTS. FIRST A LOOP DIRECT MESSAGE IS SENT TO EACH NODE IN THE TABLE. IF THIS IS SUCCESSFUL, THE EXERCISER BUILDS AN ARRAY OF NODE PAIRS FROM THE TABLE AND SENDS A FULL ASSISTED LOOPBACK MESSAGE TO EACH PAIR IN THE ARRAY. A SAMPLE ARRAY OF PAIRS FOR A TABLE WITH 7 NODES IS SHOWN BELOW.

1-2	2-3	3-4	4-5	5-6	6-7
1-3	2-4	3-5	4-6	5-7	
1-4	2-5	3-6	4-7		
1-5	2-6	3-7			
1-6	2-7				
1-7					

IDENTIFY ADR — A REQUEST ID MESSAGE IS SENT TO THE NODE SPECIFIED BY ADR AND THE RESPONDED SYSTEM ID PARAMETERS ARE PRINTED.

BUILD — THE BUILD COMMAND CAUSES THE EXERCISERS TO LISTEN FOR SYSTEM ID MESSAGES WHICH ARE BROADCAST BY ALL DEUNA NODES ONCE EVERY 10 MINUTES. ALL NODES IDENTIFYING THEMSELVES ARE ADDED TO THE NODE TABLE. THE BUILD COMMAND STOPS WHEN NO NEW NODES HAVE BEEN ADDED FOR 10 MINUTES OR WHEN 40 MINUTES HAVE ELAPSED. THE AVERAGE TIME FOR THIS COMMAND SHOULD BE 15-25 MINUTES.

CLEAR NODE/ADR — THE CLEAR NODE COMMAND CLEARS THE SPECIFIED NODE FROM THE NODE TABLE. THE NODE CAN BE SPECIFIED BY EITHER ITS 12 DIGIT PHYSICAL ADDRESS OF ITS LOGICAL NAME

(AS ASSIGNED BY NODE TABLE).

CLEAR NODE/ALL	THIS COMMAND CLEARS THE NODE TABLE.
CLEAR MESSAGE	THIS COMMAND SETS THE MESSAGE PARAMETERS BACK TO THE DEFAULT VALUES.
CLEAR SUMMARY	THIS COMMAND CLEARS THE SUMMARY TABLE.
SUMMARY	THE SUMMARY COMMAND PRINTS OUT THE SUMMARY TABLE. THE NIE MAINTAINS THE FOLLOWING INFORMATION FOR NODES WHO HAVE BEEN SENT MESSAGES: RECEIVES NOT COMPLETE RECEIVES COMPLETE LENGTH ERRORS DATA COMPARE ERRORS BYTES COMPARED BYTES TRANSFERED
SAVE	THE SAVE COMMAND SAVES THE CONTENTS OF THE NODE TABLE. FOR THE VAX VERSION, THE TABLE IS SAVED IN A FILE CALLED NIE.TBL. THE PDP-11 VERSION CANNOT WRITE TO EXTERNAL MEDIA, SO THE CONTENTS ARE SAVE INTERNALLY.
UNSAVE	THE UNSAVE COMMAND RESTORES THE CONTENTS OF THE NODE TABLE. USED. THE PDP-11 VERSION USES THE CONTENTS OF ITS INTERNALLY SAVED TABLE.
UNSAVE/FILE.EXT	THE UNSAVE COMMAND WHEN USED WITH A FILE NAME WILL READ A NODE TABLE CREATED BY USING AN EDITOR FOR AN XXDP+ MEDIA.
EXIT	RETURNS CONTROL TO THE DIAGNOSTIC SUPERVISOR (EITHER VDS OR DRS).

NOTES: 1) ADR IS THE PHYSICAL ADDRESS OF A NODE ON THE NI.
2) PASS COUNT IS A DECIMAL NUMBER BETWEEN 1 AND 65534. A DEFAULT VALUE OF 1 IS ASSUMED. SPECIFYING -1 CAUSES THE TEST TO BE RUN INDEFINATELY.

2.5 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A RESTART OR CONTINUE COMMAND, THE RUNTIME SERVICES WILL ASK FOR SOFTWARE PARAMETERS. THESE PARAMETERS WILL GOVERN SOME DIAGNOSTIC SPECIFIC OPERATION MODES. YOU WILL BE PROMPTED BY "CHANGE SW (L) ?" IF YOU WISH TO CHANGE ANY PARAMETERS, ANSWER BY TYPING "Y". THE SOFTWARE QUESTIONS AND THE DEFAULT VALUES ARE DESCRIBED IN THE NEXT PARAGRAPH(S).

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

7. ANSWER THE "CHANGE SW" QUESTION WITH "N"

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING ONLY THE DEFAULTS FOR FLAGS AND SOFTWARE PARAMETERS. THESE DEFAULTS ARE DESCRIBED IN SECTIONS 2.3 AND 2.5.

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 "IBR" 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", "IBR" OR "IXR" 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

ERROR MESSAGE:	MEANING
?ILL CMD-BAD SYNTAX	A COMMAND WITH AN ILLEGAL CHAR WAS TYPED - RETYPE THE COMMAND. THE VAILD COMMANDS AND THEIR SYNTAX ARE SHOWN IN SECTION 2.5

?INCOMPLETE	A REQUIRED PART OF A COMMAND WAS LEFT OUT.
?NUMBER 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 OCCURED WHEN TYPING A "DUMP" COMMAND WHERE OCTAL ADDRESSES ARE EXPECTED.

EXAMPLE OF A LOST PACKET ERROR DURING LOOPPAIR TESTING

```
CZUAC HRD ERR  00028 ON UNIT 00 TST 001 SUB 000 PC:064442

TIMEOUT OCCURED - LOOP MESSAGE TYPE - RECEIVE ASSIST
FAILING TARGET NODE ADDRESS: AA-00-03-00-00-00
FAILING ASSIST NODE ADDRESS: AA-00-03-00-00-02
```

EXAMPLE OF A LOST PACKET ERROR DURING PATTERN TESTING

```
CZUAC HRD ERR 00028 ON UNIT 00 TST 001 SUB 000 PC:63730

TIMEOUT OCCURED BEFORE LOOPBACK REPLY
FAILING NODE ADDRESS: AA-00-03-00-00-00
DATA PATTERN: ONES
```

4.0 PERFORMANCE AND PROGRESS REPORTS

AT THE END OF EACH PASS, THE PASS COUNT IS GIVEN ALONG WITH THE TOTAL NUMBER OF ERRORS REPORTED SINCE THE DIAGNOSTIC WAS STARTED. THE "EOP" SWITCH CAN BE USED TO CONTROL HOW OFTEN THE END OF PASS MESSAGE IS PRINTED. SECTION 2.2 DESCRIBES SWITCHES.

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. [6]) 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   174510          ;[0] CSR ADDRESS
.WORD   120             ;[2] INTERRUPT VECTOR
.WORD   240             ;[6] INTERRUPT PPIORITY (5)
```

6.0 TEST SUMMARIES

6.1 DIRECT

THE DIRECT TEST SENDS A LOOP DIRECT MESSAGE TO ALL OF THE
NODES CONTAINED IN THE NODE TABLE AND WAITS FOR A RESPONSE.
THE INTEGRITY OF THE RETURNED DATA IS CHECKED AND ANY ERRORS
ARE REPORTED TO THE OPERATOR.

6.2 LOOPPAIR

THE LOOPPAIR TEST SENDS ASSISTED LOOPBACK MESSAGES TO THE
NODE PAIRS CONTAINED IN THE NODE TABLE. THREE TYPE OF ASSISTED
MESSAGES ARE SENT:

1) RECEIVE ASSIST -- HOST -> TARGET -> ASSIST -> HOST
2) TRANSMIT ASSIST -- HOST -> ASSIST -> TARGET -> HOST
3) FULL ASSIST -- HOST -> ASSIST -> TARGET -> ASSIST -> HOST

IN EACH CASE A RESPONSE IS WAITED FOR AND THE DATA IS CHECKED.

6.3 PATTERN

THE PATTERN TEST SENDS SIX DIFFERENT LOOP DIRECT MESSAGES
TO EACH NODE CONTAINED IN THE NODE TABLE. EACH OF THE SIX
PATTERN TYPES (ALPHA, ONES, ZEROS, 1ALT, 0ALT, CCITT)
IS USED FOR EACH NODE. RETURNED DATA IS CHECKED FOR ERRORS.

6.4 ALL

THE ALL NODE TEST PERFORMS THE MOST EXTENSIVE CHECK OF THE
NETWORK AND IS COMPOSED OF TWO PARTS. FIRST A LOOP DIRECT
MESSAGE IS SENT TO EACH NODE IN THE TABLE. IF THIS IS
SUCCESSFUL, THE EXERCISER BUILDS AN ARRAY OF NODE PAIRS
FROM THE TABLE AND SENDS A FULL ASSISTED LOOPBACK MESSAGE
TO EACH PAIR IN THE ARRAY. A SAMPLE ARRAY OF PAIRS FOR A
TABLE WITH 7 NODES IS SHOWN BELOW.

1-2	2-3	3-4	4-5	5-6	6-7
1-3	2-4	3-5	4-6	5-7	
1-4	2-5	3-6	4-7		
1-5	2-6	3-7			
1-6	2-7				
1-7					

&

```
627                             .SBTTL  PROGRAM HEADER
628
629                                     .ENABL  ABS,AMA
630     002000                          .       =       2000
631
632
633                             .SBTTL  PROGRAM MACROS
634
```

```
635
636                                     ;I$STACK MACRO
637                                     ;-------------
638
639                                     ;+++
640                                     ;THE I$STACK MACRO FACILITATES INITIALIZING THE R6 (HARDWARE) STACK
641                                     ;AND THE R5 (PARAMETER) STACK.  R5 IS SET TO THE STACK LOW LIMIT
642                                     ;(STAKLO) AND THE PARAMETER STACK GROWS UPWARD.  R6 IS SET TO THE
643                                     ;STACK HIGH LIMIT (STAKHI) AND THE HARDWARE STACK GROWS DOWNWARD.
644                                     ;IF THERE IS A STACK OVER-RUN, IT WILL BE DETECTED BY THE PREG14
645                                     ;ROUTINE.
646                                     ;---
647
648                                     .MACRO  I$STACK STAKLO,STAKHI
649
650                                     MOV     STAKLO,R5       ;INITIALIZE THE PARAMETER STACK POINTER.
651                                     MOV     STAKHI,SP       ;INITIALIZE THE HARDWARE STACK POINTER.
652
653                                     .ENDM   I$STACK
654
655                                     ;PUSH MACRO
656                                     ;----------
657
658                                     ;+++
659                                     ;THE "PUSH" MACRO FACILITATES PUSHING ITEMS ON THE HARDWARE STACK.
660                                     ;UP TO SEVEN ITEMS MAY BE PLACED ON THE STACK WITH ONE MACRO.
661                                     ;---
662                                     .MACRO  PUSH    A,B,C,D,E,F,G
663
664                                     .IF NB G
665                                     MOV     G,-(SP)
666                                     .ENDC
667
668                                     .IF NB F
669                                     MOV     F,-(SP)
670                                     .ENDC
671
672                                     .IF NB E
673                                     MOV     E,-(SP)
674                                     .ENDC
675
676                                     .IF NB D
677                                     MOV     D,-(SP)
678                                     .ENDC
679
680                                     .IF NB C
681                                     MOV     C,-(SP)
682                                     .ENDC
683
684                                     .IF NB B
685                                     MOV     B,-(SP)
686                                     .ENDC
687
688                                     .IF NB A
689                                     MOV     A,-(SP)
690                                     .ENDC
```

```
691
692                                     .ENDM   PUSH
693
694                                     ;POP MACRO
695                                     ;---------
696
697                                     ;+++
698                                     ;THE "POP" MACRO FACILITATES RETRIEVING ITEMS FROM THE HARDWARE STACK.
699                                     ;UP TO SEVEN ITEMS MAY BE RETRIEVED WITH ONE MACRO.
700                                     ;---
701
702                                     .MACRO  POP     A,B,C,D,E,F,G
703
704                                     .IF NB A
705                                     MOV     (SP)+,A
706                                     .ENDC
707
708                                     .IF NB B
709                                     MOV     (SP)+,B
710                                     .ENDC
711
712                                     .IF NB C
713                                     MOV     (SP)+,C
714                                     .ENDC
715
716                                     .IF NB D
717                                     MOV     (SP)+,D
718                                     .ENDC
719
720                                     .IF NB E
721                                     MOV     (SP)+,E
722                                     .ENDC
723
724                                     .IF NB F
725                                     MOV     (SP)+,F
726                                     .ENDC
727
728                                     .IF NB G
729                                     MOV     (SP)+,G
730                                     .ENDC
731
732                                     .ENDM   POP
733
734                                     ;CALL MACRO
735                                     ;----------
736
737                                     ;+++
738                                     ;THE CALL MACRO FACILITATES CALLING A SUBROUTINE VIA THE REGISTER
739                                     ;PRESERVE ROUTINE (PREG14).  IT PLACES THE PARAMETERS TO BE PASSED ON
740                                     ;THE PARAMETER STACK.  UP TO 7 PARAMETERS MAY BE PASSED USING THIS
741                                     ;MACRO.
742                                     ;---
743
744                                     .MACRO  CALL    S  A,B,C,D,E,F,G
745
746                                     .IF NB G
```

```
                        MOV     G,(R5)+
                        .ENDC

                        .IF NB F
                        MOV     F,(R5)+
                        .ENDC

                        .IF NB E
                        MOV     E,(R5)+
                        .ENDC

                        .IF NB D
                        MOV     D,(R5)+
                        .ENDC

                        .IF NB C
                        MOV     C,(R5)+
                        .ENDC

                        .IF NB B
                        MOV     B,(R5)+
                        .ENDC

                        .IF NB A
                        MOV     A,(R5)+
                        .ENDC

                        JSR     R4,PREG14
                        .WORD   S-ANCHOR

                        .ENDM   CALL

                        ;RETURN MACRO
                        ;------------

                        ;+++
                        ;THE RETURN MACRO FACILITATES PASSING PARAMETERS BACK TO A CALLING
                        ;ROUTINE.  UP TO 7 PARAMETERS MAY BE PASSED BACK ON THE PARAMETER
                        ;STACK.
                        ;---

                        .MACRO  RETURN  A,B,C,D,E,F,G

                        .IF NB G
                        MOV     G,(R5)+
                        .ENDC

                        .IF NB F
                        MOV     F,(R5)+
                        .ENDC

                        .IF NB E
                        MOV     E,(R5)+
                        .ENDC

                        .IF NB D
```

```
                        MOV     D,(R5)+
                        .ENDC

                        .IF NB C
                        MOV     C,(R5)+
                        .ENDC

                        .IF NB B
                        MOV     B,(R5)+
                        .ENDC

                        .IF NB A
                        MOV     A,(R5)+
                        .ENDC

                        RTS     PC

                        .ENDM   RETURN

                        ;P$PUSH MACRO
                        ;------------

                        ;+++
                        ;THE P$PUSH MACRO FACILITATES PUSHING PARAMETERS ON THE PARAMETER
                        ;STACK.  UP TO SEVEN ITEMS MAY BE PUSHED WITH ONE MACRO.
                        ;---

                        .MACRO  P$PUSH  A,B,C,D,E,F,G

                        .IF NB G
                        MOV     G,(R5)+
                        .ENDC

                        .IF NB F
                        MOV     F,(R5)+
                        .ENDC

                        .IF NB E
                        MOV     E,(R5)+
                        .ENDC

                        .IF NB D
                        MOV     D,(R5)+
                        .ENDC

                        .IF NB C
                        MOV     C,(R5)+
                        .ENDC

                        .IF NB B
                        MOV     B,(R5)+
                        .ENDC

                        .IF NB A
                        MOV     A,(R5)+
                        .ENDC
```

```
859
860                                     .ENDM   P$PUSH
861
862                                     ;P$POP MACRO
863                                     ;-----------
864
865
866                                     ;+++
867                                     ;THE P$POP MACRO FACILITATES RETRIEVING PARAMETERS FROM THE PARAMETER
868                                     ;STACK.  UP TO 7 PARAMETERS MAY BE RETRIEVED.
869                                     ;
870                                     ;THE ROUTINE THAT RECEIVES THE PARAMETERS HAS THE RESPONSIBILITY OF
871                                     ;CLEANING UP THE PARAMETER STACK.  THIS MACRO IS AN AID TO MAKING
872                                     ;A LOCAL COPY OF PASSED PARAMETERS AND CLEANING UP THE PARAMETER STACK.
873                                     ;---
874
875                                     .MACRO  P$POP   A,B,C,D,E,F,G
876
877                                     .IF NB A
878                                     MOV     -(R5),A
879                                     .ENDC
880
881                                     .IF NB B
882                                     MOV     -(R5),B
883                                     .ENDC
884
885                                     .IF NB C
886                                     MOV     -(R5),C
887                                     .ENDC
888
889                                     .IF NB D
890                                     MOV     -(R5),D
891                                     .ENDC
892
893                                     .IF NB E
894                                     MOV     -(R5),E
895                                     .ENDC
896
897                                     .IF NB F
898                                     MOV     -(R5),F
899                                     .ENDC
900
901                                     .IF NB G
902                                     MOV     -(R5),G
903                                     .ENDC
904
905                                     .ENDM   P$POP
906
907
908
909                             .MACRO  CLI     CHAR,HITVAL,MISADR,CMPSTR
910                                     NODCL   CHAR,HITVAL,\X$,MISADR,CMPSTR   ;;#### PARSE TREE ####
911                             .ENDM
912
913
914                             .MACRO  NODCL   CHAR,HITVAL,XY,MISADR,CMPSTR
```

```
                                NOD'XY: .BYTE   CHAR,HITVAL             ;SPECIAL CHAR. CODE OR COMPARE CHAR.
                                                                        ; AND ACTION (HIT) VALUE FOR ACTION
                                                                        ; ROUTINES.
                                .IF NB  MISADR
                                        .WORD   MISADR-NOD'XY           ;DISPLACEMENT TO 'MISS' NODE (BYTES)
                                .ENDC
                                .IF NB  CMPSTR
                                        .WORD   1$-NOD'XY               ;DISPLACEMENT TO GET TO NEXT NODE
                                        .ASCIZ  CMPSTR                  ;  (ONLY IF ITS A 'CLISTR' NODE)
                                        .EVEN
                                  .NLIST
                                  1$:
                                  .LIST
                                .ENDC
                                .NLIST
                                X$=X$+1
                                .LIST
                                .ENDM

                                .MACRO  RNGFRM A,B,C                    ; MACRO TO FORM TRANSMIT AND RECIEVE
                                                                        ; DESCRIPTOR RINGS.

                                        .LIST
                                        .WORD   RPKLEN                  ; SEGMENT LENGTH
                                        .NLIST
                                        NEXT    A,\B
                                B=B+1
                                        .LIST
                                        .WORD   C                       ; OWNERSHIP AND STATUS BITS
                                        .WORD   0                       ; STATUS
                                        .WORD   0                       ; SEQUENCE NUMBER

                                        .NLIST
                                .ENDM

                                .MACRO  NEXT A,B
                                        .LIST
                                        .WORD   A'B                     ; SEGMENT BUFFER ADDRESS
                                        .NLIST
                                .ENDM

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

002000                                  POINTER BGNRPT


002000                                  HEADER  CZUAC,A,0,0,1,PRI07
002000                                                                  L$NAME::
002000  103                                                                     .ASCII /C/
002001  132                                                                     .ASCII /Z/
002002  125                                                                     .ASCII /U/
002003  101                                                                     .ASCII /A/
002004  103                                                                     .ASCII /C/
```

```
   971  002005      000                                    .BYTE   0
   972  002006      000                                    .BYTE   0
   973  002007      000                                    .BYTE   0
   974  002010                                      L$REV::
   975  002010      101                                    .ASCII  /A/
   976  002011                                      L$DEPO::
   977  002011      060                                    .ASCII  /0/
   978  002012                                      L$UNIT::
   979  002012  000000                                     .WORD   0
   980  002014                                      L$TIML::
   981  002014  000000                                     .WORD   0
   982  002016                                      L$HPCP::
   983  002016  112752                                     .WORD   L$HARD
   984  002020                                      L$SPCP::
   985  002020  000000                                     .WORD   0
   986  002022                                      L$HPTP::
   987  002022  002170                                     .WORD   L$HW
   988  002024                                      L$SPTP::
   989  002024  000000                                     .WORD   0
   990  002026                                      L$LADP::
   991  002026  113156                                     .WORD   L$LAST
   992  002030                                      L$STA::
   993  002030  000000                                     .WORD   0
   994  002032                                      L$CO::
   995  002032  000000                                     .WORD   0
   996  002034                                      L$DTYP::
   997  002034  000001                                     .WORD   1
   998  002036                                      L$APT::
   999  002036  000000                                     .WORD   0
  1000  002040                                      L$DTP::
  1001  002040  002164                                     .WORD   L$DISPATCH
  1002  002042                                      L$PRIO::
  1003  002042  000340                                     .WORD   PRIO7
  1004  002044                                      L$ENVI::
  1005  002044  000000                                     .WORD   0
  1006  002046                                      L$EXP1::
  1007  002046  000000                                     .WORD   0
  1008  002050                                      L$MREV::
  1009  002050      003                                    .BYTE   C$REVISION
  1010  002051      003                                    .BYTE   C$EDIT
  1011  002052                                      L$EF::
  1012  002052  000000                                     .WORD   0
  1013  002054  000000                                     .WORD   0
  1014  002056                                      L$SPC::
  1015  002056  000000                                     .WORD   0
  1016  002060                                      L$DEVP::
  1017  002060  002122                                     .WORD   L$DVTYP
  1018  002062                                      L$REPP::
  1019  002062  076560                                     .WORD   L$RPT
  1020  002064                                      L$EXP4::
  1021  002064  000000                                     .WORD   0
  1022  002066                                      L$EXP5::
  1023  002066  000000                                     .WORD   0
  1024  002070                                      L$AUT::
  1025  002070  000000                                     .WORD   0
  1026  002072                                      L$DUT::
```

```
002072  000000                                         .WORD   0
002074                                        L$LUN::
002074  000000                                         .WORD   0
002076                                        L$DESP::
002076  002130                                         .WORD   L$DESC
002100                                        L$LOAD::
002100  104035                                         EMT     E$LOAD
002102                                        L$ETP::
002102  000000                                         .WORD   0
002104                                        L$ICP::
002104  076600                                         .WORD   L$INIT
002106                                        L$CCP::
002106  100142                                         .WORD   L$CLEAN
002110                                        L$ACP::
002110  100140                                         .WORD   L$AUTO
002112                                        L$PRT::
002112  076572                                         .WORD   L$PROT
002114                                        L$TEST::
002114  000000                                         .WORD   0
002116                                        L$DLY::
002116  000000                                         .WORD   0
002120                                        L$HIME::
002120  000000                                         .WORD   0


                              ;
                              ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                              ;
002122                               DEVTYP  <DEUNA>
002122                                        L$DVTYP::
002122  042504  047125  000101                         .ASCIZ  /DEUNA/
                                                       .EVEN



                              ; TEST DESCRIPTION
                              ;
002130                               DESCRIPT        <CZUAC DEUNA NI EXERCISER>
002130                                        L$DESC::
002130  055103  040525  020103                         .ASCIZ  /CZUAC DEUNA NI
002136  042504  047125  020101
002144  044516  042440  042530
002152  041522  051511  051105
002160     000
        002162                                         .EVEN
                                     .EVEN


                              ;
                              ; FORMAT STATEMENTS USED IN PRINT CALLS
                              ;



```

```
  1080                                  .SBTTL  DISPATCH TABLE
  1081
  1082                                  ;++
  1083                                  ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
  1084                                  ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
  1085                                  ;--
  1086
  1087  002162                                  DISPATCH 1
  1088  002162  000001                                                                 .WORD   1
  1089  002164                                                                LSDISPATCH::
  1090  002164  100274                                                                 .WORD   T1
  1091
```

```
1092                                    .SBTTL  DEFAULT HARDWARE P-TABLE
1093
1094                                    ;++
1095                                    ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
1096                                    ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
1097                                    ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
1098                                    ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
1099                                    ;--
1100
1101  002166                                    BGNHW   DFPTBL
1102  002166  000003                                                                   .WORD   L10000-L$HW/2
1103  002170                                                                  L$HW::
1104  002170                                                                  DFPTBL::
1105
1106  002170  174510                           .WORD   174510                ; CSR
1107  002172  000120                           .WORD   120                   ; VECTOR
1108  002174  000240                           .WORD   PRI05                 ; PRIORITY
1109
1110
1111  002176                                    ENDHW
1112  002176                                                                  L10000:
```

```
                                .SBTTL  SOFTWARE P-TABLE

                                ;++
                                ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
                                ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
                                ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
                                ; AT RUN TIME.
                                ;--

002176                                  BGNSW   SFPTBL
002176  000000                                                         .WORD   L10001-L$SW/2
002200                                                          L$SW::
002200                                                          SFPTBL::

002200                                  ENDSW
002200                                                          L10001:

                                .SBTTL  GLOBAL EQUATES SECTION



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

002200                                  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
```

```
1169          000010                  BIT3==  BIT03
1170          000004                  BIT2==  BIT02
1171          000002                  BIT1==  BIT01
1172          000001                  BIT0==  BIT00
1173                                  ;
1174                                  ; EVENT FLAG DEFINITIONS
1175                                  ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
1176                                  ;
1177          000040                  EF.START==      32.                     ; START COMMAND WAS ISSUED
1178          000037                  EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
1179          000036                  EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
1180          000035                  EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
1181          000034                  EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
1182                                  ;
1183                                  ;
1184                                  ; PRIORITY LEVEL DEFINITIONS
1185                                  ;
1186          000340                  PRI07== 340
1187          000300                  PRI06== 300
1188          000240                  PRI05== 240
1189          000200                  PRI04== 200
1190          000140                  PRI03== 140
1191          000100                  PRI02== 100
1192          000040                  PRI01== 40
1193          000000                  PRI00== 0
1194                                  ;
1195                                  ;OPERATOR FLAG BITS
1196                                  ;
1197          000004                  EVL==           4
1198          000010                  LOT==          10
1199          000020                  ADR==          20
1200          000040                  IDU==          40
1201          000100                  ISR==         100
1202          000200                  UAM==         200
1203          000400                  BOE==         400
1204          001000                  PNT==        1000
1205          002000                  PRI==        2000
1206          004000                  IXE==        4000
1207          010000                  IBE==       10000
1208          020000                  IER==       20000
1209          040000                  LOE==       40000
1210          100000                  HOE==      100000
```

```
1211
1212                                    ;;;EQUATES FOR FLAG WORD;;;;;
1213
1214            000000                          CTARGT==0
1215            000001                          CASIST==1
1216            000002                          CSHCTR==2               ;ARG TYPE FOR 'SHOW COUNTERS' CMD
1217            000004                          CCLNAD==4               ;ARG TYPE FOR 'CLEAR NODE/ADR' CMD
1218            000010                          CCLNAL==8.              ;ARG TYPE FOR 'CLEAR NODE/ALL' CMD
1219            000020                          CEXIT==16.
1220
1221                                    ;;;CLOCK ENABLE VALUES TO BE LOADED IN CLK'S CSR;;;
1222
1223            000100                          LCLKEN==100             ; L-CLOCK CSR VALUE TO ENABLE THE CLOCK
1224            000111                          PCLKEN==111             ; P-CLOCK CSR VALUE TO ENABLE THE CLOCK
1225            001600                          PCLKCT==1600            ; P-CLOCK COUNT SET REGISTER FOR COUNTER
1226
1227                                    ; SPECIAL CLI CODES FOR "CHAR" ARGUEMENT IN CLI CALLS
1228                                    ;   (COMMAND LINE INTERPRETER DEFINITIONS)
1229            000000                          CLIERR= 0
1230            000001                          CLIEXI= 1
1231            000002                          CLIBR = 2
1232            000003                          CLIBIF= 3
1233            000004                          CLISPA= 4
1234            000005                          CLINUM= 5
1235            000006                          CLIALP= 6
1236            000007                          CLIALN= 7
1237            000010                          CLIOCT= 8.
1238            000011                          CLIDEC= 9.
1239            000012                          CLISTR= 10.
1240
1241                                    ;DEFS FOR COMMAND LINE INTERPRETATION ACTION VALUES
1242
1243            000000                          NULL=0
1244            000001                          HELP=1
1245            000002                          NODE=2
1246            000003                          BUILD=3
1247            000004                          CRUN=4
1248            000005                          CPATRN=5
1249            000006                          CSAVE=6
1250            000007                          SUMMRY=7
1251            000010                          IDENT=10
1252            000011                          EXIT=11
1253            000012                          NOTNUF=12
1254            000013                          CEXADR=13
1255            000014                          CSAVR4=14
1256            000015                          CNODE=15
1257            000016                          CALPHA=16
1258            000017                          CONES=17
1259            000020                          CZEROS=20
1260            000021                          C1ALT=21
1261            000022                          COALT=22
1262            000023                          CCCITT=23
1263            000024                          COPRSL=24
1264            000025                          CTYPE=25
1265            000026                          CSIZE=26
1266            000027                          CCPYS=27
```

```
1267            000030                          CNDADR=30
1268            000031                          CNODAL=31
1269            000032                          CRNALL=32
1270            000033                          CLUPPR=33
1271            000034                          CSHMSG=34
1272            000035                          CCLMSG=35
1273            000036                          CCNTR=36
1274            000037                          CNDLOG=37
1275            000040                          CFUNCT=40
1276            000041                          CUNSAV=41
1277            000042                          CCLSUM=42
1278            000043                          CDIR=43
1279            000044                          CDEFLT=44
1280            000045                          CUNSVF=45
1281
1282            000000                          ALPHA==0                        ;MESSAGE TYPE VALUES
1283            000001                          ONES==1
1284            000002                          ZEROS==2
1285            000003                          ONEALT==3
1286            000004                          ZROALT==4
1287            000005                          CCITT==5
1288            000006                          OPRSEL==6
1289
1290                                    ;
1291                                    ;       GLOBAL EQUATES FOR THE DEUNA DRIVER
1292                                    ;
1293                                    ;PORT CONTROL AND STATUS REGISTER 0
1294
1295
1296            100000          SERI    ==      BIT15           ; STATUS ERROR INTERRUPT
1297            040000          PCEI    ==      BIT14           ; PORT COMMAND ERROR INTERRUPT
1298            020000          RXI     ==      BIT13           ; RECEIVE RING INTERRUPT
1299            010000          TXI     ==      BIT12           ; TRANSMIT RING INTERRUPT
1300            004000          DNI     ==      BIT11           ; DONE INTERRUPT
1301            002000          RCBI    ==      BIT10           ; RECEIVE BUFFER UNAVAILABLE
1302                            ;
1303            000400          FATI    ==      BIT08           ; FATAL ERROR INTERERUPT
1304            000200          INTR    ==      BIT07           ; INTERRUPT SUMMARY <15:08>
1305            000100          INTE    ==      BIT06           ; INTERRUPT ENABLE
1306            000040          RSET    ==      BIT05           ; UNA RESET
1307
1308                                    ; PORT COMMANDS IN BIT 3 TO BIT 0
1309                                    ; -------------------------------
1310
1311            000001                  GETPCB == BIT00                 ; GET ADDRESS OF PORT CONTROL BLOCK
1312            000002                  GETFNT == BIT01                 ; GET COMMAND IN PORT CONTROL BLOCK
1313            000003                  PNOP == BIT00!BIT01             ; NO OPERATION PERFORMED
1314            000004                  STRT  == BIT02                  ;  ENABLE XMIT AND RCVR
1315            000005                  BOOT == BIT02!BIT00             ;  BOOT , -> PRIM LOAD STATE,
1316                                                                    ;   INITATE DOWNLINE LOAD
1317
1318            000010                  PDMD == BIT03                   ; POLLING DEMAND/WAKE UP BIT
1319            000011                  TMRO == BIT03!BIT00             ; SANITY TIMER ENABLE ( =1 ITS ON)
1320            000012                  TMRF == BIT03!BIT01             ; SANITY TIMER OFF
1321            000015                  RSTT == BIT03!BIT02!BIT00       ; RESET SANITY TIMER
1322            000017                  STOP == BIT03!BIT02!BIT01!BIT00 ; SUSPEND UNA OPERATION
```

```
1323
1324
1325
1326                            ;PORT CONTROL AND STATUS REGISTER 1
1327
1328
1329            100000                  XPWR == BIT15                   ; TRANSCEIVER POWER OK
1330            040000                  ICAB == BIT14                   ; PORT TO LINK CABLE OK
1331
1332                                    ; SELF TEST ERROR CODE IN BIT 13 TO BIT 08
1333            000200                  PCTO  == BIT07                  ; PORT COMMAND TIMEOUT
1334
1335            000010                  RMTC == BIT03                   ; REMOTE CONSOLE RESERVED (=1)
1336
1337                                    ; PORT STATE IN BIT 2 TO BIT 0
1338
1339            000000                  RESET == 0                      ;  000 RESET STATE
1340            000001                  PRIMLD== BIT00                  ;  001 PRIMARY LOAD STATE
1341            000002                  READY== BIT01                   ;  010 READY STATE
1342            000003                  RUN ==  BIT01!BIT00             ;  011 RUNNING STATE
1343
1344            000005                  UNIHLT == BIT02!BIT00           ;  101 UNIBUS HALTED STATE
1345            000006                  NIHLT  == BIT02!BIT01           ;  110 NI HALTED STATE
1346            000007                  NIUNI  == BIT02!BIT01!BIT00     ;  111 NI AND UNIBUS HALTED STATE
1347
1348
1349
1350                            ;PORT CONTROL AND STATUS REGISTER 2
1351
1352                                    ; LOWER 16 ADDRESS BITS OF THE PORT CONTROL BLOCK BASE
1353                                    ; ADDRESS POINTER  IN BIT 15 TO BIT 0
1354
1355                            ;PORT CONTROL AND STATUS REGISTER 3
1356
1357                                    ; UPPER 2 ADDRESS BITS OF THE PORT CONTROL BLOCK BASE
1358                                    ; ADDRESS POINTER  IN BIT 1 TO BIT 0
1359
1360                            ;PORT FUNCTIONS
1361
1362                                    ; FUNCTION CODES ARE AS FOLLOWS
1363
1364            000000                     PFNOP == 0                   ; NO OPERATION PERFORMED
1365            000002                     RDDEFA == BIT01              ; READ DEFAULT PHYSICAL ADDRESS
1366
1367            000004                     RDPHYA == BIT02              ; READ PHYSICAL ADDRESS
1368            000005                     WDPHYA == BIT02!BIT00        ; WRITE PHYSICAL ADDRESS
1369
1370            000006                     RDMULA == BIT02!BIT01        ; READ LIST OF MULTICAST ADDRESSES
1371            000007                     WDMULA == BIT02!BIT01!BIT00  ; WRITE LIST OF MULTICAST ADDRESSES
1372
1373            000010                     RDRNGS  == BIT03             ; READ BOTH THE RCVR AND XMIT RINGS
1374            000011                     WDRNGS  == BIT03!BIT00       ; WRITE BOTH THE RCVR AND XMIT RINGS
1375
1376            000012                     RDCNTS  == BIT03!BIT01       ; READ COUNTERS
1377            000013                     CLRCNTS == BIT03!BIT01!BIT00 ; READ AND CLEAR COUNTERS
1378
```

```
1379          000014                        RDMODE == BIT03!BIT02          ; READ INTERNAL LINK MODE REGISTER
1380          000015                        WDMODE == BIT03!BIT02!BIT00    ; WRITE INTERNAL LINK MODE REGISTER
1381
1382          000016                        RDSTA  == BIT03!BIT02!BIT01    ; READ PORT STATUS
1383          000017                        CLRSTA == BIT03!BIT02!BIT01!BIT00
1384                                                                       ; READ AND CLEAR PORT STATUS
1385
1386          000020                        DMPMEM == BIT04                ; DUMP INTERNAL MEMORY
1387          000021                        LDMEM  == BIT04!BIT00          ; LOAD INTERNAL MEMORY
1388
1389          000022                        RDSYS  == BIT04!BIT01          ; READ SYSTEM ID PARAMETERS
1390          000023                        WDSYS  == BIT04!BIT01!BIT00    ; WRITE SYSTEM ID PARAMETERS
1391
1392                                ;
1393                                ;       ETHERNET PACKET  OFFSETS
1394                                ;
1395
1396
1397          000016                HEADER  ==      14.             ; OFFSET (SIZE) TO END OF  HEADER IN BYTES
1398
1399          000000                DESTIN  ==      0              ; DESTINATION ADDRESS
1400          000006                SOURCC  ==      6              ; SOURCE ADDRESS
1401          000014                PROTOT  ==      12.            ; PROTOCOL TYPE FIELD
1402
1403                                ;               ------------------------
1404                                ;               !  DESTINATION ADDRESS  !
1405                                ;               ------------------------
1406                                ;               !       (6 BYTES)      !
1407                                ;               ------------------------
1408                                ;               !                      !
1409                                ;               ------------------------
1410                                ;       +6      !    SOURCE  ADDRESS   !
1411                                ;               ------------------------
1412                                ;               !       (6 BYTES)      !
1413                                ;               ------------------------
1414                                ;               !                      !
1415                                ;               ------------------------
1416                                ;       +12.    !    PROTOCALL TYPE    !
1417                                ;               ------------------------
1418                                ;       +14.    !         DATA         !
1419                                ;               ------------------------
1420                                ;               !       MORE DATA      !
1421                                ;
1422
1423                                ;+
1424                                ;       XMIT RING DESCRIPTOR DEFINITIONS
1425                                ;-
1426
1427                                ; TDRB+0
1428                                ;
1429                                ;       NOTHING NEEDED
1430
1431                                ; TDRB+2
1432                                ;
1433                                ;       NOTHING NEEDED
1434
```

```
1435
1436                                    ; TDRB+4
1437                                    ;
1438
1439            000400                          ENP     ==      BIT08           ; END OF PACKET FLAG
1440            001000                          STP     ==      BIT09           ; STOP OF PACKET FLAG
1441            002000                          DEF     ==      BIT10           ; DEFFERRING PACKET FLAG
1442            004000                          ONE     ==      BIT11           ; XMIT SUCCESSFUL AFTER ONE RETRY
1443            010000                          MORE    ==      BIT12           ; XMIT SUCCESSFUL AFTER MORE THAN
1444                                                                            ;       ONE RETRY
1445            040000                          ERRS    ==      BIT14           ; ERROR SUMMARY BIT
1446            100000                          OWN     ==      BIT15           ; OWNERSHIP BIT (=1 UNA, =0 HOST)
1447
1448                                    ; TDRB+6
1449
1450            002000                          RTRY    ==      BIT10           ; RETRY ERROR BIT
1451            004000                          LCAR    ==      BIT11           ; LOST CARRIER ERROR BIT
1452            010000                          LCOL    ==      BIT12           ; LATE COLLISION ERROR BIT
1453
1454            040000                          UBTO    ==      BIT14           ; UNIBUS TIMEOUT ERROR BIT
1455            100000                          BUFL    ==      BIT15           ; BUFFER LENGTH ERROR BIT
1456
1457                                    ;+
1458                                    ;       RCVR RING DESCRIPTOR DEFINTIONS
1459                                    ;-
1460
1461                                    ; RDRB+0
1462                                    ;
1463                                    ;       NOTHING NEEDED
1464
1465                                    ; RDRB+2
1466                                    ;
1467                                    ;       NOTHING NEEDED
1468
1469
1470                                    ; RDRB+4
1471                                    ;
1472
1473                                            ; <- INDICATES SAME AS DEFINED FOR XMIT RING
1474
1475                                            ;ENP    ==      BIT08           ; END OF PACKET FLAG
1476                                            ;STP    ==      BIT09           ; STOP OF PACKET FLAG
1477
1478            004000                          CRC     ==      BIT11           ; CRC ERROR IN RECEIVED PACKET
1479            010000                          OFLO    ==      BIT12           ; MESSAGE OVERFLOW
1480            020000                          FRAM    ==      BIT13           ; FRAMING ERROR
1481
1482                                            ;ERRS   ==      BIT14           ; ERROR SUMMARY BIT
1483                                            ;OWN    ==      BIT15           ; OWNERSHIP BIT (=1 UNA, =0 HOST)
1484
1485                                    ; RDRB+6
1486
1487            020000                          NCHN    ==      BIT13           ; SET TO INDICATE UNA IN NO
1488                                                                            ; BUFFER CHAIN ON RCVR MODE
1489
1490                                            ;UBTO   ==      BIT14           ; UNIBUS TIMEOUT ERROR BIT
```

```
1491                                    ;BUFL   ==      BIT15           ; BUFFER LENGTH ERROR BIT
1492
1493            002756                  XPKLEN  == 1518.                ; TRANSMIT PACKET LENGTH
1494            002756                  RPKLEN  == 1518.                ; RECIEVE PACKET LENGHT
1495            000006                  NO.NTR  == 6                    ; NUMBER OF ENTRIES IN DESCRIPTOR RINGS
1496            000050                  TBLLEN  == 40.                  ; NODE TABLE LENGTH (CHANGE STBLEN IF
1497                                                                    ;  THIS IS CHANGED, OR ELSE!)
1498            000132                  STBLEN  == 90.                  ; SUMMARY TABLE LENGTH (= 2.25 X TBLLEN)
1499            000004                  FRDADR  == 4                    ; OFFSET FOR MESSAGE HEADERS
1500
1501                                    ;
1502                                    ; SYSTEM ID REPLY MESSAGE OFFSETS
1503                                    ;
1504            000022                          SIRCPT == 22
1505            000027                          SIVERS == 27
1506            000030                          SIECO  == 30
1507            000031                          SIUECO == 31
1508            000035                          SIFNCT == 35
1509            000042                          SIADDR == 42
1510            000053                          SIDEV  == 53
1511
1512                                    ;
1513                                    ; LOOP DIRECT OFFSETS
1514                                    ;
1515            000016                          LDSKIP == 16
1516            000020                          LDFCT1 == 20
1517            000022                          LDADR1 == 22
1518            000030                          LDFCT2 == 30
1519            000032                          LDADR2 == 32
1520
1521                                    ;
1522                                    ; FULL ASSIST OFFSETS
1523                                    ;
1524            000016                          FASKIP == 16
1525            000020                          FAFCT1 == 20
1526            000022                          FAADR1 == 22
1527            000030                          FAFCT2 == 30
1528            000032                          FAADR2 == 32
1529            000040                          FAFCT3 == 40
1530            000042                          FAADR3 == 42
1531            000050                          FAFCT4 == 50
1532            000052                          FAADR4 == 52
1533                                    ;
1534                                    ; COUNTER OFFSETS
1535                                    ;
1536            000002                          C.SECS == 2
1537            000004                          C.PREC == 4
1538            000010                          C.MREC == 10
1539            000014                          C.RERB == 14
1540            000016                          C.RERR == 16
1541            000020                          C.RDAT == 20
1542            000024                          C.RMDB == 24
1543            000030                          C.RLIN == 30
1544            000032                          C.RLEX == 32
1545            000034                          C.PXMT == 34
1546            000040                          C.MXMT == 40
```

```
1547            000044                  C.PXM3 == 44
1548            000050                  C.PXM2 == 50
1549            000054                  C.PXMD == 54
1550            000060                  C.XDAT == 60
1551            000064                  C.XMDB == 64
1552            000066                  C.XABB == 66
1553            000070                  C.XABT == 70
1554            000074                  C.COLL == 74
1555
1556                                    .SBTTL  GLOBAL DATA SECTION
1557
1558                                    ;++
1559                                    ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1560                                    ; IN MORE THAN ONE TEST.
1561                                    ;--
1562                                    ;COMMAND LINE BUFFER, DATA LOCATIONS AND MESSAGES FOR ACTION ROUTINES
1563
1564   002200   000110                  CMDBUF: .BLKB   72.             ;BUFFER FOR OPERATOR COMMANDS
1565   002310   000000                  KEYWD1: .WORD   0               ;
1566   002312   000000                  KEYWD2: .WORD   0
1567   002314   000000                  ADRBUF: .WORD   0               ;BUFFER FOR NODE ADDRESS
1568   002316   000000                          .WORD   0
1569   002320   000000                          .WORD   0
1570   002322   000022                  STRBUF: .BLKB   18.             ;BUFFER FOR ALPHANUM. ADDRESS STRING
1571   002344   000022                  STRBU1: .BLKB   18.
1572   002366   000000                  CBOADR: .WORD   0               ;POINTER FOR BEGINING OF ADDRESS STRING
1573   002370   000000                  P$TYPE: .WORD   0               ;LOC. TO HOLD MESSAGE TYPE
1574   002372   000000                  P$SIZE: .WORD   0               ;LOC. TO HOLD MESSAGE SIZE
1575   002374   000000                  P$CPYS: .WORD   0               ;LOC. TO HOLD NO. OF MESSAGE COPIES
1576   002376   000000                  P$PASS: .WORD   0               ;LOC. TO HOLD NO. OF PASSES
1577   002400   000000                  NODTY:  .WORD   0               ;LOC. TO HOLD NODE TYPE FOR NODE TABLE SETUP
1578   002402   000000                  SLOT::  .WORD   0               ;USED BY NODE TABLE SUBROUTINES
1579   002404   000050                  NODTBL: .BLKW   TBLLEN          ; SPACE FOR NODE TABLE
1580   002524   177777                          .WORD   177777          ;FILL LAST FOUR BYTES OF TABLE WITH ONES
1581   002526   177777                          .WORD   177777
1582   002530   000050                  SAVTBL: .BLKW   TBLLEN          ;SPACE FOR SAVE TABLE
1583   002650   177777                  ILLADR: .WORD   177777          ;ILLEGAL ADDRESS FOR COMPARISON
1584   002652   177777                          .WORD   177777          ; (MUST NOT BE PHYSICALLY SEPARATED FROM
1585   002654   177777                          .WORD   177777          ; END OF SAVTBL)
1586   002656   000132                  STATBL: .BLKW   STBLEN          ;SPACE FOR SUMMARY TABLE
1587   003142   177777                          .WORD   177777
1588
1589                                    ;COMMAND LINE TRAVERSE LOCATIONS (USED BY 'P$TRV')
1590
1591   003144   000000                  P$BUFA: .WORD   0               ;LOC. TO HOLD ADDR. OF CMD LINE BUFFER
1592   003146   000000                  P$TREE: .WORD   0               ;LOC. TO HOLD ADDR. OF PARSING TREE
1593   003150   000000                  P$ACT:  .WORD   0               ;LOC. TO HOLD ADDR. OF ACTION ROUTINE
1594   003152   000000                  P$CNT:  .WORD   0               ;LOC. TO BE A COUNTER LOCATION
1595   003154   000000                  P$NUM:  .WORD   0               ;LOC. TO HOLD NUMERIC VALUE FROM PARSE
1596   003156   000000                  P$RADX: .WORD   0               ;LOC. TO HOLD RADIX(LO) & +/-(HI BYTE)
1597   003160      000                  P$NNUF: .BYTE   0               ;RETURN =0 IF ENOUGH OF COMMAND FOUND
1598   003161      000                  P$GDBD: .BYTE   0               ;RETURN CODE 0 IF NO ERROR FOUND
1599   003162      000                  P$AERR: .BYTE   0               ;RETURN 0 IF 12 DIGIT ADDRESS ENTERED
1600   003163      000                  P$MERR: .BYTE   0               ;RETURN -1 IF ERROR IN OPERATOR SELECTED
1601                                                                    ;MESSAGE INPUT OCCURED, 0 FOR GOOD INPUT
1602   003164   056060                  HLPTAB: .WORD   HELP1
```

```
1603  003166  056161                          .WORD   HELP2
1604  003170  056254                          .WORD   HELP3
1605  003172  056325                          .WORD   HELP4
1606  003174  056376                          .WORD   HELP5
1607  003176  056476                          .WORD   HELP6
1608  003200  056611                          .WORD   HELP7
1609  003202  056722                          .WORD   HELP8
1610  003204  057012                          .WORD   HELP9
1611  003206  057101                          .WORD   HELP10
1612  003210  057172                          .WORD   HELP11
1613  003212  057270                          .WORD   HELP12
1614  003214  057375                          .WORD   HELP13
1615  003216  057474                          .WORD   HELP14
1616  003220  057573                          .WORD   HELP15
1617  003222  057676                          .WORD   HELP16
1618  003224  057765                          .WORD   HELP17
1619  003226  060070                          .WORD   HELP18
1620  003230  060140                          .WORD   HELP19
1621  003232  060243                          .WORD   HELP20
1622  003234  060321                          .WORD   HELP21
1623  003236  060404                          .WORD   HELP22
1624  003240  060505                          .WORD   HELP23
1625  003242  060605                          .WORD   HELP24
1626  003244  060735                          .WORD   HELP25
1627  003246  061021                          .WORD   HELP26
1628  003250  061125                          .WORD   HELP27
1629  003252  061227                          .WORD   HELP28
1630  003254  061346                          .WORD   HELP29
1631  003256  061416                          .WORD   HELP30
1632  003260  000000                  HLPEND: .WORD   0
1633  003262  062372                  MSGTAB: .WORD   MSGTY0                  ;MESSAGE TYPE ASCII ADDRESS TABLE
1634  003264  062400                          .WORD   MSGTY1
1635  003266  062405                          .WORD   MSGTY2
1636  003270  062413                          .WORD   MSGTY3
1637  003272  062420                          .WORD   MSGTY4
1638  003274  062425                          .WORD   MSGTY5
1639  003276  062433                          .WORD   MSGTY6
1640
1641                                  ; THIS SECTION DEFINES THE DATA PATTERNS USED BY THE EXERCISER
1642
1643  003300                          MSGCNT::
1644  003300  000130                  MSG0C:  .WORD   EMSG0-MSG00             ; THE NUMBER OF BYTES IN EACH MESSAGE
1645  003302  000001                  MSG1C:  .WORD   EMSG1-MSG01
1646  003304  000001                  MSG2C:  .WORD   EMSG2-MSG02
1647  003306  000001                  MSG3C:  .WORD   EMSG3-MSG03
1648  003310  000001                  MSG4C:  .WORD   EMSG4-MSG04
1649  003312  000100                  MSG5C:  .WORD   EMSG5-MSG05
1650  003314  000000                  MSG6C:  .WORD   0
1651
1652  003316                          MSGAD::
1653  003316  003334                          .WORD   MSG00
1654  003320  003464                          .WORD   MSG01
1655  003322  003465                          .WORD   MSG02
1656  003324  003466                          .WORD   MSG03
1657  003326  003467                          .WORD   MSG04
1658  003330  003470                          .WORD   MSG05
```

```
 1659  003332  003570                          .WORD   OPSLBF
 1660
 1661  003334  020440  021442  022444  MSG00:: .ASCII  \ !''#$%&'()*+,-/0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ\
 1662  003342  023446  024450  025452
 1663  003350  026454  030057  031061
 1664  003356  032063  033065  034067
 1665  003364  035071  036073  037075
 1666  003372  040077  041101  042103
 1667  003400  043105  044107  045111
 1668  003406  046113  047115  050117
 1669  003414  051121  052123  053125
 1670  003422  054127  055131
 1671  003426  056533  026536  041101          .ASCII  \[]^-ABCDEFGHIJKLMNOPQRSTUVWXYZ\        ; ALPHANUMERIC
 1672  003434  042103  043105  044107
 1673  003442  045111  046113  047115
 1674  003450  050117  051121  052123
 1675  003456  053125  054127  055131
 1676  003464                          EMSG0::
 1677  003464     377                  MSG01:: .BYTE   377                     ; MESSAGE OF ALL ONES
 1678  003465                          EMSG1::
 1679  003465     000                  MSG02:: .BYTE   0                       ; MESSAGE OF ALL ZEROS
 1680  003466                          EMSG2::
 1681  003466     252                  MSG03:: .BYTE   252                     ; MESSAGE OF ALTERNATING ONES
 1682  003467                          EMSG3::
 1683  003467     125                  MSG04:: .BYTE   125                     ; MESSAGE OF ALTERNATING ZEROS
 1684  003470                          EMSG4::
 1685  003470                          MSG05::                                 ; CCITT 511 BIT TEST PATTERN
 1686  003470  177603  157427  031011          .WORD   177603,157427,031011,047321,163715,105221
 1687  003476  047321  163715  105221
 1688  003504  143325  142304  040041          .WORD   143325,142304,040041,104116,052606,172334
 1689  003512  104116  052606  172334
 1690  003520  105025  123754  111337          .WORD   105025,123754,111337,111523,030030,145064
 1691  003526  111523  030030  145064
 1692  003534  137642  143531  063617          .WORD   137642,143531,063617,135075,066730,026575
 1693  003542  135075  066730  026575
 1694  003550  052012  053627  070071          .WORD   052012,053627,070071,151172,165044,031605
 1695  003556  151172  165044  031605
 1696  003564  166632  016147                  .WORD   166632,016147
 1697  003570                          EMSG5::
 1698  003570  000102                  OPSLBF: .BLKB   66.              ;BUFFER FOR OPERATOR SELECTED MESSAGE TYPE
 1699
 1700
 1701  003672  000000                  CFLAG:  .WORD   0                       ;ACTION ROUTINE CMD ARGUMENT FLAG
 1702
 1703                                  ;;CLOCK TABLES, EVENT LOG AND POINTERS
 1704  003674  000000                  CLKCSR: .WORD   0               ; CLOCK CSR ADDRESS
 1705  003676  000000                  CLKBR:  .WORD   0               ; CLOCK INTERRUPT LEVEL
 1706  003700  000C00                  CLKVEC: .WORD   0               ; CLOCK INTERRUPT VECTOR
 1707  003702  000074                  CLKHZ:  .WORD   60.              ; CLOCK'S FREQUENCY IN HERTZ
 1708  003704  000000                  CLKEN:  .WORD   0               ; CLOCK'S CSR VALUE TO INTRPT. ENABLE IT
 1709
 1710  003706  000000                  TIMMIN: .WORD   0               ; PLACE TO KEEP TIME-SINCE-START
 1711  003710  000000                  TIMSEC: .WORD   0
 1712  003712  000000                  TIMTCK: .WORD   0               ; PLACE TO KEEP NO. OF TICKS/SEC.
 1713
 1714  003714  000000                  TIMER1: .WORD   0               ; EVENT TIMER #1 (TICKS)
```

```
1715  003716  000000                  TIMER2: .WORD   0               ; EVENT TIMER #2 (TICKS)
1716  003720  000000                  TIMERS: .WORD   0               ; EVENT TIMER #3 (SECONDS)
1717                                          .EVEN
1718
1719
1720                                  ;
1721                                  ;       TABLE OF  START ADDRESS OF RECIEVE RING BUFFERS
1722                                  ;
1723
1724  003722                          RRNGTB::
1725
1726  003722  026006                          .WORD RRG001
1727  003724  030764                          .WORD RRG002
1728  003726  033742                          .WORD RRG003
1729  003730  036720                          .WORD RRG004
1730  003732  041676                          .WORD RRG005
1731  003734  044654                          .WORD RRG006
1732
1733                                  ;
1734                                  ;       TABLE OF START ADDRESS OF TRANSMIT RING BUFFERS
1735                                  ;
1736
1737  003736                          XRNGTB::
1738
1739  003736  004162                          .WORD   XRG001
1740  003740  007140                          .WORD   XRG002
1741  003742  012116                          .WORD   XRG003
1742  003744  015074                          .WORD   XRG004
1743  003746  020052                          .WORD   XRG005
1744  003750  023030                          .WORD   XRG006
1745
1746                                  ;               POINTERS TO DESCRIPTOR RING ENTRIES
1747
1748  003752  003772                  XRGSRT::.WORD   XRING            ; FIRST ENTRY IN TRANSMIT RING
1749  003754  004066                  RRGSRT::.WORD   RRING            ; FIRST ENTRY IN RECIEVE RING
1750  003756  003772                  XRGCUR::.WORD   XRING            ; CURRENT ENTRY IN TRANSMIT RING
1751  003760  004066                  RRGCUR::.WORD   RRING            ; CURRENT ENTRY IN RECIEVE RING
1752  003762  003772                  XRGNXT::.WORD   XRING            ; NEXT ENTRY IN TRANSMIT RING
1753  003764  004066                  RRGNXT::.WORD   RRING            ; NEXT ENTRY IN RECIEVE RING
1754  003766  004054                  XRGLST::.WORD   XRING+50.        ; LAST ENTRY IN TRANSMIT RING
1755  003770  004150                  RRGLST::.WORD   RRING+50.        ; LAST ENTRY IN RECEIVE RING
1756                                                                   ;  VALUE = NO.NTR X 10.
1757
1758
1759  003772  000006                  XRING:: .REPT 6
1760                                          .NLIST
1761                                          RNGFRM XRG00,B,0
1762                                          .LIST
1763                                          .ENDR
1764  003772  002756                          .WORD   RPKLEN                  ; SEGMENT LENGTH
1765  003774  004162                          .WORD   XRG001                  ; SEGMENT BUFFER ADDRESS
1766  003776  000000                          .WORD   0                       ; OWNERSHIP AND STATUS BITS
1767  004000  000000                          .WORD   0                       ; STATUS
1768  004002  000000                          .WORD   0                       ; SEQUENCE NUMBER
1769
1770  004004  002756                          .WORD   RPKLEN                  ; SEGMENT LENGTH
```

```
1771  004006  007140                    .WORD   XRG002          ; SEGMENT BUFFER ADDRESS
1772  004010  000000                    .WORD   0               ; OWNERSHIP AND STATUS BITS
1773  004012  000000                    .WORD   0               ; STATUS
1774  004014  000000                    .WORD   0               ; SEQUENCE NUMBER
1775
1776  004016  002756                    .WORD   RPKLEN          ; SEGMENT LENGTH
1777  004020  012116                    .WORD   XRG003          ; SEGMENT BUFFER ADDRESS
1778  004022  000000                    .WORD   0               ; OWNERSHIP AND STATUS BITS
1779  004024  000000                    .WORD   0               ; STATUS
1780  004026  000000                    .WORD   0               ; SEQUENCE NUMBER
1781
1782  004030  002756                    .WORD   RPKLEN          ; SEGMENT LENGTH
1783  004032  015074                    .WORD   XRG004          ; SEGMENT BUFFER ADDRESS
1784  004034  000000                    .WORD   0               ; OWNERSHIP AND STATUS BITS
1785  004036  000000                    .WORD   0               ; STATUS
1786  004040  000000                    .WORD   0               ; SEQUENCE NUMBER
1787
1788  004042  002756                    .WORD   RPKLEN          ; SEGMENT LENGTH
1789  004044  020052                    .WORD   XRG005          ; SEGMENT BUFFER ADDRESS
1790  004046  000000                    .WORD   0               ; OWNERSHIP AND STATUS BITS
1791  004050  000000                    .WORD   0               ; STATUS
1792  004052  000000                    .WORD   0               ; SEQUENCE NUMBER
1793
1794  004054  002756                    .WORD   RPKLEN          ; SEGMENT LENGTH
1795  004056  023030                    .WORD   XRG006          ; SEGMENT BUFFER ADDRESS
1796  004060  000000                    .WORD   0               ; OWNERSHIP AND STATUS BITS
1797  004062  000000                    .WORD   0               ; STATUS
1798  004064  000000                    .WORD   0               ; SEQUENCE NUMBER
1799
1800
1801
1802  004066  000006          RRING:: .REPT 6
1803                                  .NLIST
1804                                  RNGFRM RRG00,B,100000
1805                                  .LIST
1806                                  .ENDR
1807  004066  002756                    .WORD   RPKLEN          ; SEGMENT LENGTH
1808  004070  026006                    .WORD   RRG001          ; SEGMENT BUFFER ADDRESS
1809  004072  100000                    .WORD   100000          ; OWNERSHIP AND STATUS BITS
1810  004074  000000                    .WORD   0               ; STATUS
1811  004076  000000                    .WORD   0               ; SEQUENCE NUMBER
1812
1813  004100  002756                    .WORD   RPKLEN          ; SEGMENT LENGTH
1814  004102  030764                    .WORD   RRG002          ; SEGMENT BUFFER ADDRESS
1815  004104  100000                    .WORD   100000          ; OWNERSHIP AND STATUS BITS
1816  004106  000000                    .WORD   0               ; STATUS
1817  004110  000000                    .WORD   0               ; SEQUENCE NUMBER
1818
1819  004112  002756                    .WORD   RPKLEN          ; SEGMENT LENGTH
1820  004114  033742                    .WORD   RRG003          ; SEGMENT BUFFER ADDRESS
1821  004116  100000                    .WORD   100000          ; OWNERSHIP AND STATUS BITS
1822  004120  000000                    .WORD   0               ; STATUS
1823  004122  000000                    .WORD   0               ; SEQUENCE NUMBER
1824
1825  004124  002756                    .WORD   RPKLEN          ; SEGMENT LENGTH
1826  004126  036720                    .WORD   RRG004          ; SEGMENT BUFFER ADDRESS
```

```
 1827  004130  100000                  .WORD   100000          ; OWNERSHIP AND STATUS BITS
 1828  004132  000000                  .WORD   0               ; STATUS
 1829  004134  000000                  .WORD   0               ; SEQUENCE NUMBER
 1830
 1831  004136  002756                  .WORD   RPKLEN          ; SEGMENT LENGTH
 1832  004140  041676                  .WORD   RRG005          ; SEGMENT BUFFER ADDRESS
 1833  004142  100000                  .WORD   100000          ; OWNERSHIP AND STATUS BITS
 1834  004144  000000                  .WORD   0               ; STATUS
 1835  004146  000000                  .WORD   0               ; SEQUENCE NUMBER
 1836
 1837  004150  002756                  .WORD   RPKLEN          ; SEGMENT LENGTH
 1838  004152  044654                  .WORD   RRG006          ; SEGMENT BUFFER ADDRESS
 1839  004154  100000                  .WORD   100000          ; OWNERSHIP AND STATUS BITS
 1840  004156  000000                  .WORD   0               ; STATUS
 1841  004160  000000                  .WORD   0               ; SEQUENCE NUMBER
 1842
 1843
 1844
 1845  004162  002756  XRG001::.BLKB   XPKLEN                  ; XMIT RING BUFFERS
 1846  007140  002756  XRG002::.BLKB   XPKLEN
 1847  012116  002756  XRG003::.BLKB   XPKLEN
 1848  015074  002756  XRG004::.BLKB   XPKLEN
 1849  020052  002756  XRG005::.BLKB   XPKLEN
 1850  023030  002756  XRG006::.BLKB   XPKLEN
 1851
 1852  026006  002756  RRG001::.BLKB   RPKLEN                  ; RECIEVE RING BUFFERS
 1853  030764  002756  RRG002::.BLKB   RPKLEN
 1854  033742  002756  RRG003::.BLKB   RPKLEN
 1855  036720  002756  RRG004::.BLKB   RPKLEN
 1856  041676  002756  RRG005::.BLKB   RPKLEN
 1857  044654  002756  RRG006::.BLKB   RPKLEN
 1858
 1859
 1860
 1861                   ;*****************************************************************************8
 1862                   ;
 1863                   ;INFORMATION ABOUT THE CURRENT UNIT AS OBTAINED FROM THE HARDWARE P-TABLE
 1864                   ;
 1865                   ;*****************************************************************************
 1866
 1867                                                  ;PCSRS OF CURRENT SLOT
 1868  047632  000000  PCSR0:: .WORD                   ;  ADDRESS OF PCSR0     (PORT COMMAND FIELD
 1869  047634  000000  PCSR1:: .WORD                   ;                 1     (STATE & SELF TEST FIELDS
 1870  047636  000000  PCSR2:: .WORD                   ;                 2     (PCB ADDRESS LO 15 BITS
 1871  047640  000000  PCSR3:: .WORD                   ;                 3     (PCB ADDRESS HI 2 BITS
 1872
 1873  047642  000000  PCSR0C::.WORD   0               ;PCSR0 CONTENTS
 1874  047644  000000  PCSR1C::.WORD   0               ;PCSR1 CONTENTS
 1875  047646  000000  PCSR2C::.WORD   0               ;PCSR2 CONTENTS
 1876  047650  000000  PCSR3C::.WORD   0               ;PCSR3 CONTENTS
 1877
 1878
 1879  047652  000000  UNACSR::.WORD   0               ;CSR
 1880  047654  000000  UNAVEC::.WORD   0               ;VECTOR
 1881  047656  000000  UNAPRI::.WORD   0               ;PRIORITY
 1882
```

```
1883  047660  000000                  FRESIZ::.WORD   0       ;POINTER TO WORD CONTAINING SIZE OF FREE MEMORY
1884  047662  000000                  FREMEM::.WORD   0       ;POINTER TO FREE MEMORY SPACE
1885
1886  047664  000000                  UNIT::  .WORD   0       ;CURRENT UNIT NUMBER BEING TESTED
1887
1888                                  ;       PORT CONTROL BLOCK FUNCTION STRUCTURES
1889
1890                                                          ;PORT CONTROL BLOCK
1891  047666  000000                  PCBB0:: .WORD   0       ;       PORT FUNCTION
1892  047670  000000                  PCBB2:: .WORD   0       ;       PORT FUNCTION DEPENDENT PARAMETERS
1893  047672  000000                  PCBB4:: .WORD   0       ;       PORT FUNCTION DEPENDENT PARAMETERS
1894  047674  000000                  PCBB6:: .WORD   0       ;       PORT FUNCTION DEPENDENT PARAMETERS
1895
1896                                  ;               FUNCTION TABLE
1897
1898  047676  047746                  FUNTAB::        .WORD   $PNOP           ; NO OP
1899  047700  000000                                  .WORD   0               ; FILL IN THE HOLE
1900  047702  047750                                  .WORD   $RDDE           ; READ DEFAULT PHYSICAL ADDRESS
1901  047704  000000                                  .WORD   0               ; FILL IN ANOTHER HOLE
1902  047706  047760                                  .WORD   $RDPH           ; READ PHYSICAL ADDRESS
1903  047710  047770                                  .WORD   $WDPH           ; WRITE PHYSICAL ADDRESS
1904  047712  050000                                  .WORD   $RDMC           ; READ MULTICAST ADDRESS LIST
1905  047714  050040                                  .WORD   $WDMC           ; WRITE MULTICAST ADDRESS LIST
1906  047716  050100                                  .WORD   $RDRN           ; READ DESCRIPTOR RINGS
1907  047720  050124                                  .WORD   $WDRN           ; WRITE DESCRIPTOR RINGS
1908  047722  050150                                  .WORD   $RDCN           ; READ COUNTERS
1909  047724  050260                                  .WORD   $CLRC           ; READ AND CLEAR COUNTERS
1910  047726  050270                                  .WORD   $RDMO           ; READ MODE
1911  047730  050300                                  .WORD   $WDMO           ; WRITE MODE
1912  047732  050310                                  .WORD   $RDST           ; READ STATUS
1913  047734  050320                                  .WORD   $CLRS           ; READ AND CLEAR STATUS
1914  047736  050330                                  .WORD   $DMEM           ; DUMP INTERNAL MEMORY
1915  047740  050350                                  .WORD   $LMEM           ; LOAD INTERNAL MEMORY
1916  047742  050360                                  .WORD   $RDSY           ; READ SYS ID PARAMETERS
1917  047744  050370                                  .WORD   $WTSY           ; WRITE SYS ID PARAMETERS
1918
1919                                  ;=
1920                                  ;       PNOP == 0                       ; PORT NO-OPERATION
1921                                  ;-
1922                                  .EVEN
1923  047746  000000                          $PNOP:  .WORD 0                 ; NO-OP
1924
1925                                  ;+
1926                                  ;       RDDEFA == BIT01                 ; READ DEFAULT PHYSICAL ADDRESS
1927                                  ;-
1928                                  .EVEN
1929
1930  047750  000002                          $RDDE::         .WORD   2       ; PCBB+0 FUNCTION READ DEFAULT
1931  047752  000000                          DEPADR::        .WORD   0       ; PCBB+2      PHYSICAL ADDRESS
1932  047754  000000                                          .WORD   0       ; PCBB+4
1933  047756  000000                                          .WORD   0       ; PCBB+6
1934
1935                                  ;+
1936                                  ;       RDPHYA == BIT02                 ; READ PHYSICAL ADDRESS
1937                                  ;-
1938                                  .EVEN
```

```
1939
1940  047760  000004                  $RDPH::         .WORD   4       ; PCBB+0 READ CURRENT (ACTIVE)
1941  047762  000000                  PHYADR::        .WORD   0       ; PCBB+2      PHYSICAL ADDRESS
1942  047764  000000                                  .WORD   0       ; PCBB+4
1943  047766  000000                                  .WORD   0       ; PCBB+6
1944
1945                                  ;+
1946                                  ;       WDPHYA == BIT02!BIT00   ; WRITE PHYSICAL ADDRESS
1947                                  ;-
1948                                  .EVEN
1949  047770  000005                  $WDPH::         .WORD   5       ; PCBB+0 WRITE PHYSICAL ADDRESS
1950  047772  000000                                  .WORD   0       ; PCBB+2
1951  047774  000000                                  .WORD   0       ; PCBB+4
1952  047776  000000                                  .WORD   0       ; PCBB+6
1953
1954                                  ;+
1955                                  ;       RDMULA == BIT02!BIT01   ; READ MULTICAST ADDRESS LIST
1956                                  ;-
1957
1958                                  .EVEN
1959  050000  000006                  $RDMC::         .WORD   6       ; FUNCTION CODE
1960  050002  050010                                  .WORD   UCB6    ; UCBB ADDRESS
1961  050004  000000                                  .WORD   0       ; PCBB+4
1962  050006  000000                                  .WORD   0       ; PCBB+6
1963
1964  050010  000014                  UCB6::          .BLKW   12.     ; ENOUGH ROOM FOR 4 ADDRESSES
1965
1966                                  ;+
1967                                  ;       WDMULA == BIT02!BIT01!BIT00 ; WRITE MULTICAST ADDRESS LIST
1968                                  ;-
1969
1970                                  .EVEN
1971  050040  000007                  $WDMC::         .WORD   7       ; FUNCTION CODE
1972  050042  050050                                  .WORD   UCB7    ; UCBB ADDRESS
1973  050044  000400                                  .WORD   400     ; LENGTH OF LIST = 1
1974  050046  000000                                  .WORD   0       ; PCBB+6
1975
1976  050050  000253                  UCB7::          .WORD   253     ; MULTICAST ADDRESS FOR LOOPBACK
1977  050052  001000                                  .WORD   1000
1978  050054  000000                                  .WORD   0
1979  050056  000011                                  .BLKW   9.      ; ROOM FOR THREE MORE ADDRESSES
1980
1981                                  ;+
1982                                  ;       RDRNGS == BIT03         ; READ BOTH THE RCVR AND XMIT RINGS
1983                                  ;-
1984
1985                                  .EVEN
1986  050100  000010                  $RDRN::         .WORD   10      ; FUNCTION CODE
1987  050102  050110                                  .WORD UCB10     ; UCBB ADDRESS
1988  050104  000000                                  .WORD   0       ; NULL
1989  050106  000000                                  .WORD   0       ; NULL
1990
1991                                  .EVEN
1992
1993  050110  003772                  UCB10::         .WORD XRING     ; UCBB
1994  050112  002000                                  .WORD 2000      ; UCBB+2
```

```
1995  050114  000000                                  .WORD   0       ; UCBB+4
1996  050116  004066                                  .WORD RRING     ; UCBB+6
1997  050120  002000                                  .WORD 2000      ; UCBB+10
1998  050122  000000                                  .WORD   0       ; UCBB+12
1999
2000
2001                                          ;+
2002                                          ;       WDRNGS  == BIT03!BIT00      ; WRITE BOTH THE RCVR AND XMIT RINGS
2003                                          ;-
2004
2005                                          .EVEN
2006
2007  050124  000011                                $WDRN::         .WORD   11      ; FUNCTION CODE
2008  050126  050134                                                .WORD  UCB11    ; UCBB ADDRESS
2009  050130  000000                                                .WORD   0       ; NULL
2010  050132  000000                                                .WORD   0       ; NULL
2011
2012                                          .EVEN
2013
2014  050134                                        UCB11::
2015  050134  003772                                                .WORD   XRING   ;       TRANSMIT RING BASE ADDRESS
2016  050136     000                                                .BYTE   0       ;       HI BITS OF TRANSMIT RING BASE ADDRESS
2017  050137     005                                                .BYTE   5       ;       FIVE WORDS PER RING ENTRY (1 FOR PORT DR
2018  050140  000006                                                .WORD   NO.NTR  ;       EIGHT TRANSMIT DESCRIPTORS IN THE RING
2019
2020  050142  004066                                                .WORD   RRING   ;       RECEIVE RING BASE ADDRESS
2021  050144     000                                                .BYTE   0       ;       HI BITS OF RECEIVE RING BASE ADDRESS
2022  050145     005                                                .BYTE   5       ;       FIVE WORDS PER RING ENTRY (1 FOR PORT DR
2023  050146  000006                                                .WORD   NO.NTR  ;       EIGHT RECEIVE DESCRIPTORS IN THE RING
2024
2025
2026
2027                                          ;+
2028                                          ;       RDCNTS  == BIT03!BIT01      ; READ COUNTERS
2029                                          ;-
2030
2031                                          .EVEN
2032  050150  000012                                $RDCN::         .WORD   12      ; FUNCTION
2033  050152  050160                                                .WORD  UCB12    ; UCBB ADDRESS
2034
2035                                                                              ; DEFAULT COUNT OF COUNTER LIST
2036                                                                                      ;       40 (OCTAL)
2037  050154  000000                                                .WORD   0       ; NULL
2038
2039  050156  000100                                                .WORD   100     ;  (# OF WORDS IN LIST = UPPER BYTE)
2040                                                                              ; MAX NUMBER VALUE = 32 (DECIMAL) =
2041
2042                                          .EVEN
2043
2044  050160                                        UCB13::
2045  050160  000000                               UCB12::         .WORD   0       ; UCBB
2046  050162  000000                                                .WORD   0       ; UCBB+2
2047  050164  000000                                                .WORD   0       ; UCBB+4
2048  050166  000000                                                .WORD   0       ; UCBB+6
2049  050170  000000                                                .WORD   0       ; UCBB+10
2050  050172  000000                                                .WORD   0       ; UCBB+12
```

```
2051  050174  000000                                    .WORD   0       ; UCBB+14
2052  050176  000000                                    .WORD   0       ; UCBB+16
2053  050200  000000                                    .WORD   0       ; UCBB+20
2054  050202  000000                                    .WORD   0       ; UCBB+22
2055  050204  000000                                    .WORD   0       ; UCBB+24
2056  050206  000000                                    .WORD   0       ; UCBB+26
2057  050210  000000                                    .WORD   0       ; UCBB+30
2058  050212  000000                                    .WORD   0       ; UCBB+32
2059  050214  000000                                    .WORD   0       ; UCBB+34
2060  050216  000000                                    .WORD   0       ; UCBB+36
2061  050220  000000                                    .WORD   0       ; UCBB+40
2062  050222  000000                                    .WORD   0       ; UCBB+42
2063  050224  000000                                    .WORD   0       ; UCBB+44
2064  050226  000000                                    .WORD   0       ; UCBB+46
2065  050230  000000                                    .WORD   0       ; UCBB+50
2066  050232  000000                                    .WORD   0       ; UCBB+52
2067  050234  000000                                    .WORD   0       ; UCBB+54
2068  050236  000000                                    .WORD   0       ; UCBB+56
2069  050240  000000                                    .WORD   0       ; UCBB+60
2070  050242  000000                                    .WORD   0       ; UCBB+62
2071  050244  000000                                    .WORD   0       ; UCBB+64
2072  050246  000000                                    .WORD   0       ; UCBB+66
2073  050250  000000                                    .WORD   0       ; UCBB+70
2074  050252  000000                                    .WORD   0       ; UCBB+72
2075  050254  000000                                    .WORD   0       ; UCBB+74
2076  050256  000000                                    .WORD   0       ; UCBB+76
2077
2078                                          ;+
2079                                          ;       CLRCNTS == BIT03!BIT01!BIT00 ; READ AND CLEAR COUNTERS
2080                                          ;-
2081
2082                                          .EVEN
2083
2084  050260  000013                  $CLRC::         .WORD   13      ; FUNCTION
2085  050262  050160                                  .WORD UCB13     ; UCBB ADDRESS
2086                                                                  ; DEFAULT COUNT OF COUNTER LIST
2087  050264  000000                                  .WORD   0       ; NULL
2088  050266  000040                                  .WORD   40      ;  (# OF WORDS IN LIST = UPPER BYTE)
2089                                                                  ; MAX NUMBER VALUE = 32 (DECIMAL) =
2090                                                                  ;       40 (OCTAL)
2091
2092
2093                                  ;( FOR  UCB13::  SEE UCB 12 ABOVE)
2094
2095
2096                                          ;+
2097                                          ;       RDMODE == BIT03!BIT02           ; READ INTERNAL LINK MODE REGISTER
2098                                          ;-
2099
2100                                          .EVEN
2101  050270  000014                  $RDMO::         .WORD   14      ; FUNCTION CODE
2102  050272  000000                                  .WORD   0       ; A 16 BIT COPY OF THE
2103                                                                  ; BITS TO READ THE UNA INTERNAL
2104                                                                  ;       MODE REGISTER
2105  050274  000000                                  .WORD   0       ; NULL
2106  050276  000000                                  .WORD   0       ; NULL
```

```
2107
2108                                    ;+
2109                                    ;       WDMODE == BIT03!BIT02!BIT00  ; WRITE INTERNAL LINK MODE REGISTER
2110                                    ;-
2111
2112                                    .EVEN
2113  050300  000015                    $WDMO::         .WORD   15      ; FUNCTION CODE
2114  050302  000000                            .WORD   0       ; A 16 BIT COPY OF THE
2115                                                            ; BITS TO WRITE THE UNA INTERNAL
2116                                                            ;       MODE REGISTER
2117  050304  000000                            .WORD   0       ; NULL
2118  050306  000000                            .WORD   0       ; NULL
2119
2120
2121                                    ;+
2122                                    ;       RDSTA  == BIT03!BIT02!BIT01  ; READ PORT STATUS
2123                                    ;-
2124
2125                                    .EVEN
2126  050310  000016                    $RDST::         .WORD   16      ; FUNCTION CODE
2127  050312  000000                    STATUS::        .WORD   0       ; A LIST OF ERRORS AND STATUS
2128  050314  000000                            .WORD   0       ; LOWER BYTE = # OF MULTICAST ADRS
2129                                                            ;       MAXIMUM SUPPORTED BY UNA
2130                                                            ; UPPER BYTE = # OF MULTICAST ADRS
2131                                                            ;       CURRENTLY SUPPORTED BY UNA
2132  050316  000000                            .WORD   0       ; WORD = MAXIMUM # OF WORDS IN
2133                                                            ;       UCB FOR COUNTERS
2134                                                            ;       AS CURRENTLY PERCEIVED
2135                                                            ;       BY THE UNA
2136
2137                                    ;+
2138                                    ;       CLRSTA  == BIT03!BIT02!BIT01!BIT0
2139                                    ;-                              ; READ AND CLEAR WRITE PORT STATUS
2140
2141                                    .EVEN
2142  050320  000017                    $CLRS::         .WORD   17      ; FUNCTION CODE
2143  050322  000000                            .WORD   0       ; A LIST OF ERRORS AND STATUS
2144  050324  000000                            .WORD   0       ; LOWER BYTE = # OF MULTICAST ADRS
2145                                                            ;       MAXIMUM SUPPORTED BY UNA
2146                                                            ; UPPER BYTE = # OF MULTICAST ADRS
2147                                                            ;       CURRENTLY SUPPORTED BY UNA
2148  050326  000000                            .WORD   0       ; WORD = MAXIMUM # OF WORDS IN
2149                                                            ;       UCB FOR COUNTERS
2150                                                            ;       AS CURRENTLY PERCEIVED
2151                                                            ;       BY THE UNA
2152
2153                                    ;+
2154                                    ;       DMPMEM == BIT04             ; DUMP INTERNAL MEMORY
2155                                    ;-
2156
2157                                    .EVEN
2158  050330  000020                    $DMEM::         .WORD   20      ; FUNCTION CODE
2159  050332  050340                            .WORD   UCB20   ; UCBB ADDRESS
2160  050334  000000                            .WORD   0
2161  050336  000000                            .WORD   0
2162
```

```
                                        UCB20::
                                        UCB21::         .WORD   0       ; FUNCTION LENGTH (NO OF WORDS TO XFER)
                                                        .WORD   0       ; HDBB - HOST MEMORY DATA BLOCK ADDRESS
                                                        .WORD   0       ; MUST BE ZERO
                                                        .WORD   0       ; IDBB - INTERNAL DATA BLOCK BASE ADDRESS

                                ;+
                                ;       LDMEM  == BIT04!BIT00           ; LOAD UNA INTERNAL MEMORY
                                ;-

                                .EVEN
                                        $LMEM::         .WORD   21      ; FUNCTION CODE
                                                        .WORD   UCB21   ; UCBB ADDRESS
                                                        .WORD   0
                                                        .WORD   0

                                ;+
                                ;       RDSYS  == BIT04!BIT01           ; READ SYSTEM ID
                                ;-

                                .EVEN
                                        $RDSY::         .WORD   22      ; FUNCTION CODE
                                                        .WORD   UCB22   ; UCBB ADDRESS
                                                        .WORD   0
                                                        .WORD   27.     ; LENGTH OF ID MESSAGE
                                ;+
                                ;       WTSYS  == BIT04!BIT01!BIT00     ; WRITE SYSTEM ID
                                ;-

                                .EVEN
                                        $WTSY::         .WORD   23      ; FUNCTION CODE
                                                        .WORD   UCB23   ; UCBB ADDRESS
                                                        .WORD   0
                                                        .WORD   27.     ; LENGTH OF ID MESSAGE

                                        UCB22:
                                        UCB23:          .WORD   0       ;UDBB+0
                                                        .WORD   0       ;UDBB+2
                                                        .WORD   0       ;UDBB+4
                                                        .WORD   0       ;UDBB+6
                                                        .WORD   0       ;UDBB+10
                                                        .WORD   0       ;UDBB+12
                                                        .WORD   0       ;UDBB+14
                                                        .WORD   0       ;UDBB+16
                                                        .WORD   0       ;UDBB+20
                                                        .WORD   0       ;UDBB+22
                                                        .WORD   0       ;UDBB+24
                                                        .WORD   0       ;UDBB+26
                                                        .WORD   0       ;UDBB+30
                                                        .WORD   0       ;UDBB+32
                                                        .WORD   0       ;UDBB+34
                                                        .WORD   0       ;UDBB+36
                                                        .WORD   0       ;UDBB+40
                                                        .WORD   0       ;UDBB+42
                                                        .WORD   0       ;UDBB+44
                                                        .WORD   0       ;UDBB+46
```

```
2219  050450  000000                                  .WORD   0       ;UDBB+50
2220  050452  000000                                  .WORD   0       ;UDBB+52
2221  050454  000000                                  .WORD   0       ;UDBB+54
2222  050456  000000                                  .WORD   0       ;UDBB+56
2223  050460  000000                                  .WORD   0       ;UDBB+60
2224  050462  000000                                  .WORD   0       ;UDBB+62
2225  050464  000000                                  .WORD   0       ;UDBB+64
2226
2227  050466  000000                  UDBB::  .WORD   0       ;UNIBUS DATA BLOCK BASE
2228  050470  000000                          .WORD   0       ;+2
2229  050472  000000                          .WORD   0       ;+4
2230  050474  000000                          .WORD   0       ;+6
2231
2232                                  ;
2233                                  ;       SUMMARY DATA COUNTERS
2234                                  ;
2235
2236  050476  000000                  S.REC:: .WORD   0                       ; MESSAGES RECEIVED
2237  050500  000000                  S.NREC::.WORD   0                       ; MESSAGES NOT RECEIVED
2238  050502  000000                  S.LEN:: .WORD   0                       ; LENGTH ERRORS
2239  050504  000000                  S.COMP::.WORD   0                       ; COMPARE ERRORS
2240  050506  000000                  S.BYTE::.WORD   0                       ; BYTES COMPARED
2241  050510  000000                  S.XFER::.WORD   0                       ; BYTES TRANSFERED
2242
2243                                  ;
2244                                  ;       DEUNA DRIVER AND ASSOCIATED SUBROUTINES DATA
2245                                  ;
2246
2247  050512  000000                  FATFLG::.WORD   0                       ; FATAL ERROR FLAG
2248  050514  000000                  PCEFLG::.WORD   0                       ; PORT COMMAND ERROR FLAG
2249  050516  000000                  NIRCNT::.WORD   0                       ; UNA RECIEVE MESSAGE COUNTER
2250  050520  000000                  XFLAG:: .WORD   0                       ; FRAME TRANSMITTED FLAG
2251  050522  000000                  DNIFLG::.WORD   0                       ; DONE INTERRUPT FLAG
2252  050524  000000                  RBFCNT::.WORD   0                       ; RECIEVE BUFFERS LOST COUNTER
2253  050526  000000                  BCOUNT::.WORD   0                       ; UNEXPLAINED INTERRUPTS COUNTER
2254  050530  000000                  ERRFLG::.WORD   0                       ; ERROR FLAG
2255  050532  000000                  TIMOUT::.WORD   0                       ; TIME OUT COUNTER
2256  050534  000000                  RETRYS::.WORD   0                       ; COUNTER FOR FRAMES FAILING DUE TO RTRY ERROR
2257  050536  000000                  RCVERR::.WORD   0                       ; COUNTS NO. OF BUFFERS RECEIVED WITH  ERRORS
2258  050540  000000                  RCVBUF::.WORD   0                       ; COUNTS NO. OF GOOD BUFFERS RECEIVED
2259  050542  000000                  COUNT:: .WORD   0                       ; USED IN BLDBUF SUBROUTINE AS COUNTER
2260  050544  000220                  PROT00::.WORD   000220                  ; PROTOCALL TYPE FOR LOOPBACK MESSAGES
2261  050546  001140                  PROT02::.WORD   001140                  ; PROTOCALL TYPE FOR REMOTE CONSOLE
2262  050550  000000                  TEMP::  .WORD   0                       ; USED IN XMIT AS TEMPORARY STORAGE
2263  050552  000000                  TEMP1:: .WORD   0                       ; USED FOR TEMPORARY STORAGE
2264  050554  000000                  TEMP2:: .WORD   0                       ; USED FOR TEMPORARY STORAGE
2265  050556  000000                  TEMP3:: .WORD   0                       ; USED FOR TEMPORARY STORAGE
2266  050560  000000                  XFER::  .WORD   0                       ; STORES 'BYTES TRANSFERED'
2267  050562  000000                  CPYCNT::.WORD   0                       ; 'NO. OF COPIES' COUNTER FOR LOOPING
2268  050564  000000                  PCCALL::.WORD   0                       ; STORES PC OF CALLING ROUTINE FOR ERROR REPORTS
2269  050566  000000                  BUFLEN::.WORD   0                       ; STORES TRANSMIT BUFFER LENGTH
2270  050570  000000                  CMPBUF::.WORD   0                       ; STORES LOCATION OF DATA BUFFER TO BE COMPARED
2271  050572  000050                  PATCH:: .BLKW   40.                     ; 40 WORDS FOR PROGRAM PATCH
2272
2273                                  ;
2274                                  ; REQUEST ID MESSAGE FORMAT
```

```
2275                                    ;
2276
2277  050712                            REQID::
2278  050712  000003                            .WORD   3                       ; BYTE COUNT (=3 FOR REQUEST ID)
2279  050714  000005                            .WORD   5                       ; FUNCTION CODE FOR REQUEST ID
2280  050716  051115                            .WORD   'MR                     ; RECIEPT NUMBER
2281
2282                                    ;
2283                                    ; LOOP DIRECT MESSAGE
2284                                    ;
2285
2286                                    .EVEN
2287
2288  050720                            LOPDIR::
2289  050720  000000                            .WORD   0                       ; SKIP COUNT
2290  050722  000002                            .WORD   2                       ; FUNCTION = FORWARD DATA
2291  050724  000000  000000  000000            .WORD   0,0,0                   ; LOCAL NODE ADDRESS
2292  050732  000001                            .WORD   1                       ; FUNCTION = REPLY
2293  050734  000000  000000  000000            .WORD   0,0,0                   ; LOCAL NODE ADDRESS
2294
2295                                    ;
2296                                    ; TRANSMIT ASSIST MESSAGE
2297                                    ;
2298
2299  050742                            TASIST::
2300  050742  000000                            .WORD   0                       ; SKIP COUNT
2301  050744  000002                            .WORD   2                       ; FUNCTION = FORWARD DATA
2302  050746  000000  000000  000000            .WORD   0,0,0                   ; TRANSMIT ASSIST ADDRESS
2303  050754  000002                            .WORD   2                       ; FUNCTION = FORWARD DATA
2304  050756  000000  000000  000000            .WORD   0,0,0                   ; LOCAL NODE ADDRESS
2305  050764  000001                            .WORD   1                       ; FUNCTION = REPLY
2306  050766  000000  000000  000000            .WORD   0,0,0                   ; LOCAL NODE ADDRESS
2307
2308                                    ;
2309                                    ; RECIEVE ASSIST MESSAGE
2310                                    ;
2311
2312  050774                            RASIST::
2313  050774  000000                            .WORD   0                       ; SKIP COUNT
2314  050776  000002                            .WORD   2                       ; FUNCTION = FORWARD DATA
2315  051000  000000  000000  000000            .WORD   0,0,0                   ; TRANSMIT ASSIST ADDRESS
2316  051006  000002                            .WORD   2                       ; FUNCTION = FORWARD DATA
2317  051010  000000  000000  000000            .WORD   0,0,0                   ; LOCAL NODE ADDRESS
2318  051016  000001                            .WORD   1                       ; FUNCTION = REPLY
2319  051020  000000  000000  000000            .WORD   0,0,0                   ; LOCAL NODE ADDRESS
2320
2321                                    ;
2322                                    ; FULL ASSIST MESSAGE
2323                                    ;
2324
2325  051026                            FASIST::
2326  051026  000000                            .WORD   0                       ; SKIP COUNT
2327  051030  000002                            .WORD   2                       ; FUNCTION = FORWARD DATA
2328  051032  000000  000000  000000            .WORD   0,0,0                   ; TARGET NODE ADDRESS
2329  051040  000002                            .WORD   2                       ; FUNCTION = FORWARD DATA
2330  051042  000000  000000  000000            .WORD   0,0,0                   ; ASSIST NODE ADDRESS
```

```
2331  051050  000002                          .WORD   2                       ; FUNCTION = FORWARD DATA
2332  051052  000000  000000  000000          .WORD   0,0,0                   ; LOCAL NODE ADDRESS
2333  051060  000001                          .WORD   1                       ; FUNCTION = REPLY
2334  051062  000000  000000  000000          .WORD   0,0,0                   ; LOCAL NODE ADDRESS
2335
2336
2337                                  .SBTTL  COMMAND LINE ACTION TREE
2338
2339                                  ;SAMPLE CLI TREE NODE  (ALWAYS AT LEAST 1 WORD)
2340
2341                                  ; ----------------------
2342                                  ; ! ACTION  ! CHAR CODE !
2343                                  ; ----------------------
2344                                  ; !  MISS DISPLACEMENT  !       ONLY IF "MISS" ARGUMENT DEFINED
2345                                  ; ----------------------
2346                                  ; !  NEXT MODE DISPLMNT !       ONLY IF "ASCII" ARGUMENT DEFINED
2347                                  ; ----------------------
2348                                  ; !  ASCIZ MATCH STRING !       ONLY IF "ASCII" ARGUMENT DEFINED
2349                                  ; !        (.EVEN)      !
2350                                  ; ----------------------
2351
2352  051070                          CLITRE:
2353
2354                                  ;FIRST KEYWORD
2355  051070                                CLI     CLISPA,0,N10$                   ;SKIP ANY LEADING SPACES
2356  051074                          N10$: CLI     <'?>,HELP,N12$                  ;IS THE FIRST NON-SP CHAR. A "?"
2357  051100                                CLI     CLIEXI,0                        ;  IF YES DO "HLP" AND EXIT
2358  051102                          N12$: CLI     CLISTR,HELP,N14$,<'HELP'>       ;ELSE IS FIRST WORD A "HELP"
2359  051116                                CLI     CLIEXI,0                        ;  IF YES DO "HLP" AND EXIT
2360  051120                          N14$: CLI     CLISTR,NOTNUF,N16$,<'NODE'>     ;ELSE IS FIRST WORD A "NODE"
2361  051134                                CLI     CLIBR,0,N80$                    ;  IF YES, BR N80$
2362  051140                          N16$: CLI     CLISTR,BUILD,N18$,<'BUILD'>     ;ELSE IS FIRST WORD A "BUILD"
2363  051154                                CLI     CLIEXI,0                        ;  IF YES DO "BUILD" AND EXIT
2364  051156                          N18$: CLI     CLISTR,NOTNUF,N20$,<'RUN'>      ;ELSE IS FIRST WORD A "RUN"
2365  051170                                CLI     CLIBR,0,N180$                   ;  IF YES, BR N180$
2366  051174                          N20$: CLI     <'S>,NOTNUF,N25$                ;ELSE IS FIRST CHAR. A "S"
2367  051200                                CLI     CLISTR,0,N22$,<'HOW'>           ;  IF YES IS REST OF WORD "HOW"
2368  051212                                CLI     CLIBR,0,N100$                   ;    IF YES,BR N100$
2369  051216                          N22$: CLI     CLISTR,SUMMRY,N23$,<'UMMARY'>   ;  ELSE IS REST OF WORD "UMMARY"
2370  051234                                CLI     CLIEXI,0                        ;    IF YES, DO "SUMM" AND EXIT
2371  051236                          N23$: CLI     CLISTR,CSAVE,N24$,<'AVE'>       ;  ELSE IS REST OF WORD "AVE"
2372  051250                                CLI     CLIEXI,0                        ;    IF YES, DO "SAVE" AND EXIT
2373  051252                          N24$: CLI     CLIERR,0                        ;  ELSE "ILL COMMAND"
2374  051254                                CLI     CLIEXI,0                        ;  EXIT
2375  051256                          N25$: CLI     CLISTR,NOTNUF,N26$,<'CLEAR'>    ;ELSE IS FIRST WORD A "CLEAR"
2376  051272                                CLI     CLIBR,0,N120$                   ;  IF YES, BR N120$
2377  051276                          N26$: CLI     CLISTR,NOTNUF,N28$,<'IDENTIFY'> ;ELSE IS FIRST WORD "IDENTIFY"
2378  051316                                CLI     CLIBR,0,N140$                   ;  IF YES, GET ADDRS, BR N140$
2379  051322                          N28$: CLI     CLISTR,NOTNUF,N29$,<'MESSAGE'>  ;ELSE IS FIRST WORD "MESSAGE"
2380  051340                                CLI     CLIBR,0,N160$                   ;  IF YES, BR N160$
2381  051344                          N29$: CLI     CLISTR,CUNSAV,N30$,<'UNSAVE'>   ;ELSE IS FIRST WORD "UNSAVE"
2382  051362                                CLI     CLIBR,0,N210$                   ;  IF YES, BR TO N210$
2383  051366                          N30$: CLI     CLISTR,EXIT,N31$,<'EXIT'>       ;ELSE IS FIRST WORD "EXIT"
2384  051402                                CLI     CLIEXI,0                        ;  IF YES EXIT
2385  051404                          N31$: CLI     CLISTR,NOTNUF,N32$,<'FUNCTION'> ;ELSE IS FIRST WORD "FUNCTION"
2386  051424                                CLI     CLIBR,0,N200$                   ;  IF YES, BR N200$
```

```
2387  051430                  N32$:   CLI     CLIERR,0                        ;OTHERWISE ''ILL CMD'',
2388  051432                          CLI     CLIEXI,0                        ;  EXIT
2389
2390                          ;SECOND KEYWORD (ADR/TYPE) FOR NODE COMMAND
2391
2392  051434                  N80$:   CLI     CLISPA,0,N81$                   ;SKIP ANY LEADING SPACES
2393  051440                  N81$:   CLI     CLIBR,CSAVR4,N82$               ;SAVE STRING POINTER LOCATION
2394  051444                  N82$:   CLI     CLIBR,NODE,N90$                 ;PARSE THROUGH ADDRESS,CHECK
2395                                                                          ;FOR TARGET OR ASSIST, DO NODE
2396  051450                  N90$:   CLI     CLIBIF,0,N32$                   ;TAKE ERROR BRANCH IF ERROR EXISTS
2397  051454                  N95$:   CLI     CLIEXI,0                        ;EXIT
2398
2399                          ;SECOND KEYWORD FOR SHOW COMMAND
2400
2401  051456                  N100$:  CLI     CLISPA,0,N101$                  ;SKIP LEADING SPACES
2402  051462                  N101$:  CLI     CLISTR,CNODE,N102$,<'NODES'>    ;IS NEXT WORD 'NODES''
2403  051476                          CLI     CLIBR,0,N110$                   ;  IF YES, SET FLAG, BR N110$
2404  051502                  N102$:  CLI     CLISTR,CSHMSG,N104$,<'MESSAGE'> ;ELSE IS NEXT WORD 'MESSAGE''
2405  051520                          CLI     CLIBR,0,N110$                   ;  IF YES ,SET FLAG, BR N110$
2406  051524                  N104$:  CLI     CLISTR,CCNTR,N106$,<'COUNTERS'> ;ELSE IS NEXT WORD ''COUNTERS''
2407  051544                          CLI     CLIBR,0,N110$                   ; GO TO COUNTERS ROUTINE
2408  051550                  N106$:  CLI     CLIBR,0,N32$                    ;ELSE ''ILL COMMAND''
2409  051554                  N110$:  CLI     CLIEXI,0                        ;EXIT
2410
2411                          ;SECOND KEYWORD FOR CLEAR COMMAND
2412
2413  051556                  N120$:  CLI     CLISPA,0,N121$                  ;SKIP LEADING SPACES
2414  051562                  N121$:  CLI     CLISTR,0,N130$,<'NODE'>         ;IS NEXT WORD 'NODE''
2415  051576                          CLI     CLISPA,0,N122$                  ;  IF YES SKIP SPACES
2416  051602                  N122$:  CLI     <'/>,CSAVR4,N32$                ;  LOOK FOR DELIMETER, ELSE ''ILL COM''
2417  051606                          CLI     <'A>,0,N123$                    ;  IS NEXT CHAR. AN ''A''
2418  051612                          CLI     CLISTR,CNODAL,N124$,<'LL'>      ;    IF YES, IS WORD ''ALL''
2419  051624                          CLI     CLIBR,0,N135$                   ;      IF YES, SET FLAG,BR N135$
2420  051630                  N123$:  CLI     <'N>,0,N124$                    ;  ELSE IS NEXT CHAR. AN 'N''
2421  051634                          CLI     CLIDEC,0,N32$                   ;    IF YES, STORE NODE LOGICAL NAME
2422  051640                          CLI     CLIBR,CNDLOG,N135$              ;    BR TO CLR. NODE LOGICAL ROUTINE
2423  051644                  N124$:  CLI     CLIBR,CEXADR,N126$              ;  ELSE, EXTRACT ADDRESS
2424  051650                  N126$:  CLI     CLIBR,CNDADR,N135$              ;  SET FLAG, BR N135$
2425  051654                  N130$:  CLI     CLISTR,CCLMSG,N132$,<'MESSAGE'> ;ELSE IS NEXT WORD 'MESSAGE''
2426  051672                          CLI     CLIBR,0,N135$                   ;  IF YES, SET FLAG, BR N135$
2427  051676                  N132$:  CLI     CLISTR,CCLSUM,N134$,<'SUMMARY'> ;ELSE IS NEXT WORD 'SUMMARY''
2428  051714                          CLI     CLIBR,0,N135$                   ;  IF YES, CLEAR TABLE AND EXIT
2429  051720                  N134$:  CLI     CLIERR,0                        ;ELSE, ''ILL COMMAND'',
2430  051722                  N135$:  CLI     CLIEXI,0                        ;EXIT
2431
2432                          ;ADDRESS FOR IDENTIFY COMMAND
2433
2434  051724                  N140$:  CLI     CLISPA,0,N141$                  ;SKIP LEADING SPACES
2435  051730                  N141$:  CLI     CLIBR,CSAVR4,N142$              ;SAVE POINTER TO FIRST CHAR. OF ADDRESS
2436  051734                  N142$:  CLI     CLIALN,0,N32$                   ;CHECK THAT ADDRESS HAS LEGAL CHAR.S
2437  051740                          CLI     CLIBR,CEXADR,N143$              ;GET ADDRESS
2438  051744                  N143$:  CLI     CLIEXI,IDENT                    ;DO ''IDENTIFY'', EXIT
2439
2440                          ;REMAINING COMMAND LINE FOR MESSAGE COMMAND
2441
2442  051746                  N160$:  CLI     CLISPA,0,N161$                  ;SKIP LEADING SPACES
```

```
 2443  051752                 N161$:  CLI     <'/>,0,N178$                    ;IF CHAR. "/", CONT., ELSE BR N178$
 2444  051756                         CLI     CLISTR,0,N170$,<'TYPE'>         ;IS NEXT WORD "TYPE"
 2445  051772                         CLI     <'=>,0,N32$                     ;  IF YES, FOLLOWED BY "="?
 2446  051776                         CLI     CLISTR,CALPHA,N162$,<'ALPHA'>   ;    IF "ALPHA", SET FLAG
 2447  052012                         CLI     CLIBR,0,N168$                   ;    CONTINUE AT N168$
 2448  052016                 N162$:  CLI     CLISTR,CONES,N163$,<'ONES'>     ;    IF "ONES", SET FLAG
 2449  052032                         CLI     CLIBR,0,N168$                   ;    CONTINUE AT N168$
 2450  052036                 N163$:  CLI     CLISTR,CZEROS,N164$,<'ZEROS'>   ;    IF "ZEROS", SET FLAG
 2451  052052                         CLI     CLIBR,0,N168$                   ;    CONTINUE AT N168$
 2452  052056                 N164$:  CLI     CLISTR,C1ALT,N165$,<'1ALT'>     ;    IF "1ALT", SET FLAG
 2453  052072                         CLI     CLIBR,0,N168$                   ;    CONTINUE AT N168$
 2454  052076                 N165$:  CLI     CLISTR,COALT,N166$,<'0ALT'>     ;    IF "0ALT", SET FLAG
 2455  052112                         CLI     CLIBR,0,N168$                   ;    CONTINUE AT N168$
 2456  052116                 N166$:  CLI     CLISTR,CCCITT,N167$,<'CCITT'>   ;    IF "CCITT", SET FLAG
 2457  052132                         CLI     CLIBR,0,N168$                   ;    CONTINUE AT N168$
 2458  052136                 N167$:  CLI     <''>,CSAVR4,N32$                ;    IF "OPERATOR", SET FLAG
 2459  052142                         CLI     CLIBR,COPRSL,N168$              ;    AND INPUT SPECIFIED STRING
 2460  052146                 N168$:  CLI     CLIBR,CTYPE,N160$               ;  DO "TYPE", CHECK FOR MORE INPUT
 2461  052152                 N170$:  CLI     CLISTR,0,N175$,<'SIZE'>         ;ELSE IS WORD "SIZE"
 2462  052166                         CLI     <'=>,0,N32$                     ;  IF YES, FOLLOWED BY "="?
 2463  052172                         CLI     CLIDEC,CSIZE,N32$               ;    STORE NUMBER IN MSSIZE
 2464  052176                         CLI     CLIBR,0,N160$                   ;  CHECK FOR MORE INFO
 2465  052202                 N175$:  CLI     CLISTR,0,N32$,<'COPIES'>        ;ELSE IS WORD "COPIES"
 2466  052220                         CLI     <'=>,0,N32$                     ;  IF YES, FOLLOWED BY "="?
 2467  052224                         CLI     CLIDEC,CCPYS,N32$               ;    STORE NUMBER IN MSCPYS
 2468  052230                         CLI     CLIBR,0,N160$                   ;  CHECK FOR MORE INFO
 2469  052234                 N178$:  CLI     CLIBR,0,N32$                    ;ELSE "ILL COMMAND"
 2470
 2471                                 ;SECOND KEYWORD FOR RUN COMMAND
 2472
 2473  052240                 N180$:  CLI     CLISPA,0,N181$                  ;SKIP LEADING SPACES
 2474  052244                 N181$:  CLI     CLISTR,CLUPPR,N182$,<'LOOPPAIR'>        ;IS NEXT WORD "LOOPPAIR"
 2475  052264                         CLI     CLIBR,0,N185$                   ;  IF YES, SET "LOOPPAIR" FLAG
 2476  052270                 N182$:  CLI     CLISTR,CRNALL,N183$,<'ALL'>     ;ELSE IS NEXT WORD "ALL"
 2477  052302                         CLI     CLIBR,0,N185$                   ;  IF YES, SET "ALL" FLAG
 2478  052306                 N183$:  CLI     CLISTR,CDIR,N184$,<'DIRECT'>    ;ELSE IS NEXT WORD "DIRECT"
 2479  052324                         CLI     CLIBR,0,N185$                   ;  IF YES, SET "DIRECT" FLAG
 2480  052330                 N184$:  CLI     CLISTR,CPATRN,N32$,<'PATTERN'>  ;ELSE IS NEXT WORD "PATTERN"
 2481  052346                 N185$:  CLI     CLIBR,CDEFLT,N186$              ;SEE IF DEFAULT OF 1 PASS
 2482  052352                 N186$:  CLI     CLISTR,0,N32$,<'/PASS'>         ;PARSE THROUGH SWITCH
 2483  052366                         CLI     <'=>,0,N32$                     ;PARSE THROUGH "="
 2484  052372                         CLI     CLIDEC,0,N32$                   ;GET PASS COUNT
 2485  052376                 N190$:  CLI     CLIEXI,CRUN                     ;RUN TEST AND EXIT
 2486
 2487                                 ;REMAINING COMMAND LINE FOR FUNCTION COMMAND
 2488
 2489  052400                 N200$:  CLI     CLISPA,0,N201$                  ; SKIP SPACES
 2490  052404                 N201$:  CLI     CLIOCT,CFUNCT,N32$              ; GET OCTAL NUMBER AND DO FUNCT
 2491  052410                         CLI     CLIEXI,0                        ; EXIT
 2492
 2493                                 ;REMAINING COMMAND LINE FOR UNSAVE COMMAND
 2494
 2495  052412                 N210$:  CLI     CLISPA,0,N212$                  ;SKIP SPACES
 2496  052416                 N212$:  CLI     <'/>,0,N215$                    ;PARSE THROUGH '/'
 2497  052422                         CLI     CLIEXI,CUNSVF                   ;SAVE POINTER TO FILE NAME
 2498  052424                 N215$:  CLI     CLIEXI,0
```

```

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;        THE ERRTBL MACRO IS REQUIRED IF YOU INTEND TO REPORT ERRORS USING
                                ;        THE "ERROR" MACRO.  THE ERRTBL MACRO EXPANDS INTO FOUR WORDS THAT
                                ;        ARE USED BY THE RUNTIME SERVICES DURING AN ERROR CALL: ERROR TYPE,
                                ;        ERROR NUMBER, ADDRESS OF ERROR MESSAGE AND ADDRESS OF MESSAGE
                                ;        BLOCK.  THERE MUST BE ONLY ONE ERRTBL IN ANY PROGRAM.  THIS SECTION
                                ;        IS OPTIONAL.  REMOVE IF IT IF YOU ARE NOT GOING TO USE THE ERROR
                                ;        MACRO.  CHANGE THE POINTER MACRO TO REFLECT THIS SECTION'S DEL-
                                ;        ETION IF YOU REMOVE IT.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 052426                                  ERRTBL
 052426                                                                L$ERRTBL::
 052426  000000                 ERRTYP::        .WORD   0
 052430  000000                 ERRNBR::        .WORD   0
 052432  000000                 ERRMSG::        .WORD   0
 052434  000000                 ERRBLK::        .WORD   0
```

```
                                        .SBTTL  GLOBAL TEXT SECTION

                                        ;++
                                        ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
                                        ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
                                        ; MORE THAN ONE TEST.
                                        ;--
052436  006412  044516  037105  CLI$PM: .ASCIZ  <12><15>/NIE>/                    ;NIE PROMPT
052444     000
052445     045  022516  037501  CLIERM: .ASCIZ  /%N%A?ILL CMD-BAD SYNTAX?/
052452  046111  020114  046503
052460  026504  040502  020104
052466  054523  052116  054101
052474  000077
052476  047045  040445  044477  CLINUF: .ASCIZ  /%N%A?INCOMPLETE COMMAND?/
052504  041516  046517  046120
052512  052105  020105  047503
052520  046515  047101  037504
052526     000
052527     045  022516  037501  CLINBG: .ASCIZ  /%N%A?NUMBER TOO BIG?/
052534  052516  041115  051105
052542  052040  047517  041040
052550  043511  000077
052554  047045  040445  041077  CLIBRX: .ASCIZ  /%N%A?BAD RADIX?/
052562  042101  051040  042101
052570  054111  000077
052574  047045  040445  047516  LDRESP: .ASCIZ  /%N%ANODE %T%A HAS RESPONDED./
052602  042504  022440  022524
052610  020101  040510  020123
052616  042522  050123  047117
052624  042504  027104     000
052631     045  022516  050101  RECERR: .ASCIZ  /%N%APACKET RECEIVED WITH UNA ERROR./
052636  041501  042513  020124
052644  042522  042503  053111
052652  042105  053440  052111
052660  020110  047125  020101
052666  051105  047522  027122
052674     000
052675     045  022516  052101  RTRYER: .ASCIZ  /%N%ATRANSMISSION ABORTED -- EXCESSIVE COLLISIONS./
052702  040522  051516  044515
052710  051523  047511  020116
052716  041101  051117  042524
052724  020104  026455  042440
052732  041530  051505  044523
052740  042526  041440  046117
052746  044514  044523  047117
052754  027123     000
052757     045  022516  031104  BLDMSG: .ASCIZ  /%N%D2%A NODE ADDRESSES ADDED, ELAPSED TIME: %D2%A MINUTE./
052764  040445  047040  042117
052772  020105  042101  051104
053000  051505  042523  020123
053006  042101  042504  026104
053014  042440  040514  051520
053022  042105  052040  046511
053030  035105  022440  031104
053036  040445  046440  047111
```

```
2573  053044  052125  027105     000
2574  053051     045  022516  041501  ILADMS: .ASCII  /%N%ACANNOT USE BROADCAST ADDRESS (FF-FF-FF-FF-FF-FF)/
2575  053056  047101  047516  020124
2576  053064  051525  020105  051102
2577  053072  040517  041504  051501
2578  053100  020124  042101  051104
2579  053106  051505  020123  043050
2580  053114  026506  043106  043055
2581  053122  026506  043106  043055
2582  053130  026506  043106     051
2583  053135     045  022516  043101  ILADM1: .ASCIZ  /%N%AFOR LOOP TESTING. ADDRESS WAS NOT ADDED TO NODE TABLE.%N/
2584  053142  051117  046040  047517
2585  053150  020120  042524  052123
2586  053156  047111  027107  040440
2587  053164  042104  042522  051523
2588  053172  053440  051501  047040
2589  053200  052117  040440  042104
2590  053206  042105  052040  020117
2591  053214  047516  042504  052040
2592  053222  041101  042514  022456
2593  053230  000116
2594  053232  047045  040445  046120  CADRER: .ASCIZ  /%N%APLEASE ENTER TWELVE HEXADECIMAL DIGITS./
2595  053240  040505  042523  042440
2596  053246  052116  051105  052040
2597  053254  042527  053114  020105
2598  053262  042510  040530  042504
2599  053270  044503  040515  020114
2600  053276  044504  044507  051524
2601  053304  000056
2602  053306  047045  040445  053524  CADERR: .ASCIZ  /%N%ATWELVE HEX-DIGITS REQUIRED FOR ADDRESS./
2603  053314  046105  042526  044040
2604  053322  054105  042055  043511
2605  053330  052111  020123  042522
2606  053336  052521  051111  042105
2607  053344  043040  051117  040440
2608  053352  042104  042522  051523
2609  053360  000056
2610  053362  047045  040445  020101  NULSTR: .ASCIZ  /%N%AA ZERO LENGTH STRING WAS ENTERED./
2611  053370  042532  047522  046040
2612  053376  047105  052107  020110
2613  053404  052123  044522  043516
2614  053412  053440  051501  042440
2615  053420  052116  051105  042105
2616  053426  000056
2617  053430  047045  051445  022462  NODADR: .ASCIZ  /%N%S2%T/
2618  053436  000124
2619  053440  051445  032061  040445  LOGNAM: .ASCIZ  /%S14%AN%D2/
2620  053446  022516  031104     000
2621  053453     045  030523  022464  NODTYP: .ASCIZ  /%S14%T/
2622  053460  000124
2623  053462  047045  040445  047516  NTBHDR: .ASCIZ  \%N%ANODE PHYSICAL ADDRESS     NODE LOGICAL NAME     TYPE(A/T)\
2624  053470  042504  050040  054510
2625  053476  044523  040503  020114
2626  053504  042101  051104  051505
2627  053512  020123  020040  020040
2628  053520  047516  042504  046040
```

```
2629  053526  043517  041511  046101
2630  053534  047040  046501  020105
2631  053542  020040  020040  054524
2632  053550  042520  040450  052057
2633  053556  000051
2634  053560  047045  040445  044124  TABFUL: .ASCIZ  /%N%ATHE %T%A TABLE IS FILLED TO CAPACITY!/
2635  053566  020105  052045  040445
2636  053574  052040  041101  042514
2637  053602  044440  020123  044506
2638  053610  046114  042105  052040
2639  053616  020117  040503  040520
2640  053624  044503  054524  000041
2641  053632  047045  040445  044124  TABEMT: .ASCIZ  /%N%ATHE %T%A TABLE IS CURRENTLY EMPTY!/
2642  053640  020105  052045  040445
2643  053646  052040  041101  042514
2644  053654  044440  020123  052503
2645  053662  051122  047105  046124
2646  053670  020131  046505  052120
2647  053676  020531     000
2648  053701     116  042117  000105  NOD:    .ASCIZ  /NODE/
2649  053706  052523  046515  051101  SUMM:   .ASCIZ  /SUMMARY/
2650  053714  000131
2651  053716  047045  040445  044124  CLRMSG: .ASCIZ  /%N%ATHE MESSAGE PARAMETERS HAVE BEEN RESET TO:/
2652  053724  020105  042515  051523
2653  053732  043501  020105  040520
2654  053740  040522  042515  042524
2655  053746  051522  044040  053101
2656  053754  020105  042502  047105
2657  053762  051040  051505  052105
2658  053770  052040  035117     000
2659  053775     045  022516  052101  CPYLMT: .ASCIZ  /%N%ATHE NUMBER OF COPIES MUST BE BETWEEN 1 AND 255./
2660  054002  042510  047040  046525
2661  054010  042502  020122  043117
2662  054016  041440  050117  042511
2663  054024  020123  052515  052123
2664  054032  041040  020105  042502
2665  054040  053524  042505  020116
2666  054046  020061  047101  020104
2667  054054  032462  027065     000
2668  054061     045  022516  052101  SIZLMT: .ASCIZ  /%N%ATHE MESSAGE SIZE [DATA] MUST BE BETWEEN 32 AND 1466 BYTES./
2669  054066  042510  046440  051505
2670  054074  040523  042507  051440
2671  054102  055111  020105  042133
2672  054110  052101  056501  046440
2673  054116  051525  020124  042502
2674  054124  041040  052105  042527
2675  054132  047105  051440  020062
2676  054140  047101  020104  032061
2677  054146  033066  041040  052131
2678  054154  051505  000056
2679  054160  047045  040445  044124  NOCMPR: .ASCIZ  /%N%ATHE ADDRESS MARKED FOR DELETION WAS NOT IN THE TABLE./
2680  054166  020105  042101  051104
2681  054174  051505  020123  040515
2682  054202  045522  042105  043040
2683  054210  051117  042040  046105
2684  054216  052105  047511  020116
```

```
 2685  054224  040527  020123  047516
 2686  054232  020124  047111  052040
 2687  054240  042510  052040  041101
 2688  054246  042514  000056
 2689  054252  047045  040445  047101  UNBOND: .ASCIZ  /%N%AAN UNBOUNDED ''OPERATOR INPUT'' STRING WAS ENTERED./
 2690  054260  052440  041116  052517
 2691  054266  042116  042105  021040
 2692  054274  050117  051105  052101
 2693  054302  051117  044440  050116
 2694  054310  052125  020042  052123
 2695  054316  044522  043516  053440
 2696  054324  051501  042440  052116
 2697  054332  051105  042105  000056
 2698  054340  047045  040445  044124  ADRDEL: .ASCIZ  /%N%ATHE ADDRESS HAS BEEN DELETED FROM THE NODE TABLE./
 2699  054346  020105  042101  051104
 2700  054354  051505  020123  040510
 2701  054362  020123  042502  047105
 2702  054370  042040  046105  052105
 2703  054376  042105  043040  047522
 2704  054404  020115  044124  020105
 2705  054412  047516  042504  052040
 2706  054420  041101  042514  000056
 2707  054426  047045  040445  047516  LOGDEL: .ASCIZ  /%N%ANODE N%D1%A HAS BEEN DELETED FROM THE NODE TABLE./
 2708  054434  042504  047040  042045
 2709  054442  022461  020101  040510
 2710  054450  020123  042502  047105
 2711  054456  042040  046105  052105
 2712  054464  042105  043040  047522
 2713  054472  020115  044124  020105
 2714  054500  047516  042504  052040
 2715  054506  041101  042514  000056
 2716  054514  047045  040445  044124  TABCLR: .ASCIZ  /%N%ATHE %T%A TABLE HAS BEEN CLEARED./
 2717  054522  020105  052045  040445
 2718  054530  052040  041101  042514
 2719  054536  044040  051501  041040
 2720  054544  042505  020116  046103
 2721  054552  040505  042522  027104
 2722  054560     000
 2723  054561     045  022516  052101  UNSMSG: .ASCIZ  /%N%ATHE NODE TABLE HAS BEEN %T/
 2724  054566  042510  047040  042117
 2725  054574  020105  040524  046102
 2726  054602  020105  040510  020123
 2727  054610  042502  047105  022440
 2728  054616  000124
 2729  054620  040523  042526  027104  SAVED:  .ASCIZ  /SAVED./
 2730  054626     000
 2731  054627     122  051505  047524  RESTOR: .ASCIZ  /RESTORED./
 2732  054634  042522  027104     000
 2733  054641     045  022516  052101  MSGPRM: .ASCIZ  /%N%ATHE CURRENT MESSAGE PARAMETERS ARE:/
 2734  054646  042510  041440  051125
 2735  054654  042522  052116  046440
 2736  054662  051505  040523  042507
 2737  054670  050040  051101  046501
 2738  054676  052105  051105  020123
 2739  054704  051101  035105     000
 2740  054711     045  022516  052101  MSG1:   .ASCIZ  /%N%ATHE COLLECTION OF ALL NODE ADDRESSES COULD TAKE AS LONG AS 40 MINUT
```

```
 2741  054716  042510  041440  046117
 2742  054724  042514  052103  047511
 2743  054732  020116  043117  040440
 2744  054740  046114  047040  042117
 2745  054746  020105  042101  051104
 2746  054754  051505  042523  020123
 2747  054762  047503  046125  020104
 2748  054770  040524  042513  040440
 2749  054776  020123  047514  043516
 2750  055004  040440  020123  030064
 2751  055012  046440  047111  052125
 2752  055020  051505  000054
 2753  055024  047045  040445  047510  MSG11:  .ASCIZ  /%N%AHOWEVER, IF NO NEW NODES ARE ADDED TO THE TABLE FOR A 10 MINUTE PER
 2754  055032  042527  042526  026122
 2755  055040  044440  020106  047516
 2756  055046  047040  053505  047040
 2757  055054  042117  051505  040440
 2758  055062  042522  040440  042104
 2759  055070  042105  052040  020117
 2760  055076  044124  020105  040524
 2761  055104  046102  020105  047506
 2762  055112  020122  020101  030061
 2763  055120  046440  047111  052125
 2764  055126  020105  042520  044522
 2765  055134  042117     000
 2766  055137     045  022516  052101  MSG12:  .ASCIZ  /%N%ATHE COLLECTION WILL STOP.%N/
 2767  055144  042510  041440  046117
 2768  055152  042514  052103  047511
 2769  055160  020116  044527  046114
 2770  055166  051440  047524  027120
 2771  055174  047045     000
 2772  055177     045  022516  054501  MSG2:   .ASCIZ  /%N%AYOU ENTERED %T%A NODE: %T/
 2773  055204  052517  042440  052116
 2774  055212  051105  042105  022440
 2775  055220  022524  020101  047516
 2776  055226  042504  020072  052045
 2777  055234     000
 2778  055235     045  022516  052101  MSG3:   .ASCIZ  /%N%ATHE SPECIFIED ADDRESS IS: %T/
 2779  055242  042510  051440  042520
 2780  055250  044503  044506  042105
 2781  055256  040440  042104  042522
 2782  055264  051523  044440  035123
 2783  055272  022440  000124
 2784  055276  047045  040445  054524  MSG4:   .ASCIZ  /%N%ATYPE=%T%A,SIZE=%D4%A,COPIES=%D3/
 2785  055304  042520  022475  022524
 2786  055312  026101  044523  042532
 2787  055320  022475  032104  040445
 2788  055326  041454  050117  042511
 2789  055334  036523  042045  000063
 2790                          .EVEN
 2791  055342  047045  040445  042440  HDMSG1: .ASCIZ  /%N%A ETHERNET DEFAULT ADDRESS (HEX):  %T/
 2792  055350  044124  051105  042516
 2793  055356  020124  042504  040506
 2794  055364  046125  020124  042101
 2795  055372  051104  051505  020123
 2796  055400  044050  054105  035051
```

```
 2797  055406  020040  052045     000
 2798  055413     045  031116  040445  HDMSG2: .ASCIZ  /%N2%A ROM MICROCODE VERSION (DECIMAL): %D3/
 2799  055420  051040  046517  046440
 2800  055426  041511  047522  047503
 2801  055434  042504  053040  051105
 2802  055442  044523  047117  024040
 2803  055450  042504  044503  040515
 2804  055456  024514  020072  042045
 2805  055464  000063
 2806  055466  047045  022462  020101  HDMSG3: .ASCIZ  /%N2%A SWITCH PACK SET FOR :/
 2807  055474  053523  052111  044103
 2808  055502  050040  041501  020113
 2809  055510  042523  020124  047506
 2810  055516  020122  000072
 2811  055522  047045  040445  020040  HDMSG4: .ASCIZ  /%N%A        REMOTE AND POWER UP BOOT ENABLED/
 2812  055530  020040  020040  020040
 2813  055536  042522  047515  042524
 2814  055544  040440  042116  050040
 2815  055552  053517  051105  052440
 2816  055560  020120  047502  052117
 2817  055566  042440  040516  046102
 2818  055574  042105     000
 2819  055577     045  022516  020101  HDMSG5: .ASCIZ  /%N%A        REMOTE BOOT ENABLED WITH ROM/
 2820  055604  020040  020040  020040
 2821  055612  051040  046505  052117
 2822  055620  020105  047502  052117
 2823  055626  042440  040516  046102
 2824  055634  042105  053440  052111
 2825  055642  020110  047522  000115
 2826  055650  047045  040445  020040  HDMSG6: .ASCIZ  /%N%A        REMOTE BOOT ENABLED/
 2827  055656  020040  020040  020040
 2828  055664  042522  047515  042524
 2829  055672  041040  047517  020124
 2830  055700  047105  041101  042514
 2831  055706  000104
 2832  055710  047045  040445  020040  HDMSG7: .ASCIZ  /%N%A        REMOTE BOOT DISABLED/
 2833  055716  020040  020040  020040
 2834  055724  042522  047515  042524
 2835  055732  041040  047517  020124
 2836  055740  044504  040523  046102
 2837  055746  042105     000
 2838  055751     045  022516  020101  HDMSG8: .ASCIZ  /%N%A        SELF TEST LOOP ENABLED/
 2839  055756  020040  020040  020040
 2840  055764  051440  046105  020106
 2841  055772  042524  052123  046040
 2842  056000  047517  020120  047105
 2843  056006  041101  042514  000104
 2844  056014  047045  040445  020040  HDMSG9: .ASCIZ  /%N%A        SELF TEST LOOP DISABLED/
 2845  056022  020040  020040  020040
 2846  056030  042523  043114  052040
 2847  056036  051505  020124  047514
 2848  056044  050117  042040  051511
 2849  056052  041101  042514  000104
 2850                          .EVEN
 2851  056060  047045  040445  047503  HELP1:  .ASCIZ  \%N%ACOMMAND SUMMARY FOR THE NETWORK INTERCONNECT EXERCISER (NIE)\
 2852  056066  046515  047101  020104
```

```
 2853  056074  052523  046515  051101
 2854  056102  020131  047506  020122
 2855  056110  044124  020105  042516
 2856  056116  053524  051117  020113
 2857  056124  047111  042524  041522
 2858  056132  047117  042516  052103
 2859  056140  042440  042530  041522
 2860  056146  051511  051105  024040
 2861  056154  044516  024505     000
 2862  056161     045  022516  024101  HELP2:  .ASCIZ  \%N%A(IT IS ONLY NECESSARY TO TYPE THE LETTERS IN BRACKETS)\
 2863  056166  052111  044440  020123
 2864  056174  047117  054514  047040
 2865  056202  041505  051505  040523
 2866  056210  054522  052040  020117
 2867  056216  054524  042520  052040
 2868  056224  042510  046040  052105
 2869  056232  042524  051522  044440
 2870  056240  020116  051102  041501
 2871  056246  042513  051524  000051
 2872  056254  047045  022462  055501  HELP3:  .ASCIZ  \%N2%A[H]ELP OR ?       - TYPES THIS HELP TEXT.\
 2873  056262  056510  046105  020120
 2874  056270  051117  037440  026411
 2875  056276  052040  050131  051505
 2876  056304  052040  044510  020123
 2877  056312  042510  050114  052040
 2878  056320  054105  027124     000
 2879  056325     045  031116  040445  HELP4:  .ASCIZ  \%N2%A[E]XIT           - RETURN TO THE SUPERVISOR.\
 2880  056332  042533  054135  052111
 2881  056340  004411  020055  042522
 2882  056346  052524  047122  052040
 2883  056354  020117  044124  020105
 2884  056362  052523  042520  053122
 2885  056370  051511  051117  000056
 2886  056376  047045  022462  055501  HELP5:  .ASCIZ  \%N2%A[SH]OW [N]ODES    - PRINTS INFORMATION IN CURRENT NODE TABLE.\
 2887  056404  044123  047535  020127
 2888  056412  047133  047535  042504
 2889  056420  004523  020055  051120
 2890  056426  047111  051524  044440
 2891  056434  043116  051117  040515
 2892  056442  044524  047117  044440
 2893  056450  020116  052503  051122
 2894  056456  047105  020124  047516
 2895  056464  042504  052040  041101
 2896  056472  042514  000056
 2897  056476  047045  022462  055501  HELP6:  .ASCIZ  \%N2%A[SH]OW [M]ESSAGE  - PRINTS THE SELECTED MESSAGE TYPE, SIZE AND COP
 2898  056504  044123  047535  020127
 2899  056512  046533  042535  051523
 2900  056520  043501  004505  020055
 2901  056526  051120  047111  051524
 2902  056534  052040  042510  051440
 2903  056542  046105  041505  042524
 2904  056550  020104  042515  051523
 2905  056556  043501  020105  054524
 2906  056564  042520  020054  044523
 2907  056572  042532  040440  042116
 2908  056600  041440  050117  042511
```

```
 2909  056606  027123     000
 2910  056611     045  031116  040445  HELP7:  .ASCIZ  \%N2%A[SH]OW [C]OUNTERS - PRINTS THE LOW LEVEL COUNTERS OF THE HOST NODE
 2911  056616  051533  056510  053517
 2912  056624  055440  056503  052517
 2913  056632  052116  051105  020123
 2914  056640  020055  051120  047111
 2915  056646  051524  052040  042510
 2916  056654  046040  053517  046040
 2917  056662  053105  046105  041440
 2918  056670  052517  052116  051105
 2919  056676  020123  043117  052040
 2920  056704  042510  044040  051517
 2921  056712  020124  047516  042504
 2922  056720  000056
 2923  056722  047045  022462  055501  HELP8:  .ASCIZ  \%N2%A[R]UN [L]OOPPAIR/PASS=NN - RUNS THE LOOPPAIR TEST.\
 2924  056730  056522  047125  055440
 2925  056736  056514  047517  050120
 2926  056744  044501  027522  040520
 2927  056752  051523  047075  020116
 2928  056760  020055  052522  051516
 2929  056766  052040  042510  046040
 2930  056774  047517  050120  044501
 2931  057002  020122  042524  052123
 2932  057010  000056
 2933  057012  047045  022462  055501  HELP9:  .ASCIZ  \%N2%A[R]UN [A]LL/PASS=NN - RUNS THE NODE-TO-NODE TEST.\
 2934  057020  056522  047125  055440
 2935  057026  056501  046114  050057
 2936  057034  051501  036523  047116
 2937  057042  026440  051040  047125
 2938  057050  020123  044124  020105
 2939  057056  047516  042504  052055
 2940  057064  026517  047516  042504
 2941  057072  052040  051505  027124
 2942  057100     000
 2943  057101     045  031116  040445  HELP10: .ASCIZ  \%N2%A[R]UN [D]IRECT/PASS=NN - RUNS THE LOOP DIRECT TEST.\
 2944  057106  051133  052535  020116
 2945  057114  042133  044535  042522
 2946  057122  052103  050057  051501
 2947  057130  036523  047116  026440
 2948  057136  051040  047125  020123
 2949  057144  044124  020105  047514
 2950  057152  050117  042040  051111
 2951  057160  041505  020124  042524
 2952  057166  052123  000056
 2953  057172  047045  022462  055501  HELP11: .ASCIZ  \%N2%A[R]UN [P]ATTERN/PASS=NN - RUNS THE MESSAGE PATTERN TEST.\
 2954  057200  056522  047125  055440
 2955  057206  056520  052101  042524
 2956  057214  047122  050057  051501
 2957  057222  036523  047116  026440
 2958  057230  051040  047125  020123
 2959  057236  044124  020105  042515
 2960  057244  051523  043501  020105
 2961  057252  040520  052124  051105
 2962  057260  020116  042524  052123
 2963  057266  000056
 2964  057270  047045  022462  055501  HELP12: .ASCIZ  \%N2%A[M]ESSAGE/[T]YPE=A/[S]IZE=N/[C]OPIES=M - ALLOWS THE OPERATOR TO\
```

```
 2965  057276  056515  051505  040523
 2966  057304  042507  055457  056524
 2967  057312  050131  036505  027501
 2968  057320  051533  044535  042532
 2969  057326  047075  055457  056503
 2970  057334  050117  042511  036523
 2971  057342  020115  020055  046101
 2972  057350  047514  051527  052040
 2973  057356  042510  047440  042520
 2974  057364  040522  047524  020122
 2975  057372  047524     000
 2976  057375     045  022516  046501  HELP13: .ASCIZ  \%N%AMODIFY THE DEFAULT MESSAGE TYPE, SIZE AND COPY PARAMETERS.\
 2977  057402  042117  043111  020131
 2978  057410  044124  020105  042504
 2979  057416  040506  046125  020124
 2980  057424  042515  051523  043501
 2981  057432  020105  054524  042520
 2982  057440  020054  044523  042532
 2983  057446  040440  042116  041440
 2984  057454  050117  020131  040520
 2985  057462  040522  042515  042524
 2986  057470  051522  000056
 2987  057474  047045  022462  055501  HELP14: .ASCIZ  \%N2%A[N]ODE ADR/TYPE - ENTERS A PHYSICAL ADDRESS INTO THE NODE\
 2988  057502  056516  042117  020105
 2989  057510  042101  027522  054524
 2990  057516  042520  026440  042440
 2991  057524  052116  051105  020123
 2992  057532  020101  044120  051531
 2993  057540  041511  046101  040440
 2994  057546  042104  042522  051523
 2995  057554  044440  052116  020117
 2996  057562  044124  020105  047516
 2997  057570  042504     000
 2998  057573     045  022516  052101  HELP15: .ASCIZ  \%N%ATABLE.  THE TYPE CAN BE EITHER [T]ARGET (DEFAULT) OR [A]SSIST.\
 2999  057600  041101  042514  020056
 3000  057606  052040  042510  052040
 3001  057614  050131  020105  040503
 3002  057622  020116  042502  042440
 3003  057630  052111  042510  020122
 3004  057636  052133  040535  043522
 3005  057644  052105  024040  042504
 3006  057652  040506  046125  024524
 3007  057660  047440  020122  040533
 3008  057666  051535  044523  052123
 3009  057674  000056
 3010  057676  047045  022462  055501  HELP16: .ASCIZ  \%N2%A[SU]MMARY - PRINTS A SUMMARY OF THE TEST RESULTS.\
 3011  057704  052523  046535  040515
 3012  057712  054522  026440  050040
 3013  057720  044522  052116  020123
 3014  057726  020101  052523  046515
 3015  057734  051101  020131  043117
 3016  057742  052040  042510  052040
 3017  057750  051505  020124  042522
 3018  057756  052523  052114  027123
 3019  057764     000
 3020  057765     045  031116  040445  HELP17: .ASCIZ  \%N2%A[B]UILD - BUILDS A TABLE OF REMOTE NODE PHYSICAL ADDRESSES BY\
```

```
3021  057772  041133  052535  046111
3022  060000  020104  020055  052502
3023  060006  046111  051504  040440
3024  060014  052040  041101  042514
3025  060022  047440  020106  042522
3026  060030  047515  042524  047040
3027  060036  042117  020105  044120
3028  060044  051531  041511  046101
3029  060052  040440  042104  042522
3030  060060  051523  051505  041040
3031  060066  000131
3032  060070  047045  040445  044514  HELP18: .ASCIZ  \%N%ALISTENING TO ID MESSAGES ON THE NI.\
3033  060076  052123  047105  047111
3034  060104  020107  047524  044440
3035  060112  020104  042515  051523
3036  060120  043501  051505  047440
3037  060126  020116  044124  020105
3038  060134  044516  000056
3039  060140  047045  022462  055501  HELP19: .ASCIZ  \%N2%A[C]LEAR [N]ODE/ADR - REMOVES THE NODE SPECIFIED BY EITHER ADR\
3040  060146  056503  042514  051101
3041  060154  055440  056516  042117
3042  060162  027505  042101  020122
3043  060170  020055  042522  047515
3044  060176  042526  020123  044124
3045  060204  020105  047516  042504
3046  060212  051440  042520  044503
3047  060220  044506  042105  041040
3048  060226  020131  044505  044124
3049  060234  051105  040440  051104
3050  060242     000
3051  060243     045  022516  047501  HELP20: .ASCIZ  \%N%AOR NODE LOGICAL NAME FROM THE NODE TABLE.\
3052  060250  020122  047516  042504
3053  060256  046040  043517  041511
3054  060264  046101  047040  046501
3055  060272  020105  051106  046517
3056  060300  052040  042510  047040
3057  060306  042117  020105  040524
3058  060314  046102  027105     000
3059  060321     045  031116  040445  HELP21: .ASCIZ  \%N2%A[C]LEAR [N]ODE/[AL]L - CLEARS THE NODE TABLE.\
3060  060326  041533  046135  040505
3061  060334  020122  047133  047535
3062  060342  042504  055457  046101
3063  060350  046135  026440  041440
3064  060356  042514  051101  020123
3065  060364  044124  020105  047516
3066  060372  042504  052040  041101
3067  060400  042514  000056
3068  060404  047045  022462  055501  HELP22: .ASCIZ  \%N2%A[C]LEAR [M]ESSAGE - SETS ALL MESSAGE PARAMETERS TO DEFAULT.\
3069  060412  056503  042514  051101
3070  060420  055440  056515  051505
3071  060426  040523  042507  026440
3072  060434  051440  052105  020123
3073  060442  046101  020114  042515
3074  060450  051523  043501  020105
3075  060456  040520  040522  042515
3076  060464  042524  051522  052040
```

```
3077  060472  020117  042504  040506
3078  060500  046125  027124     000
3079  060505     045  031116  040445  HELP23. .ASCIZ  \%N2%A[C]LEAR [S]UMMARY - CLEARS THE TABLE OF SUMMARY TEST DATA.\
3080  060512  041533  046135  040505
3081  060520  020122  051533  052535
3082  060526  046515  051101  020131
3083  060534  020055  046103  040505
3084  060542  051522  052040  042510
3085  060550  052040  041101  042514
3086  060556  047440  020106  052523
3087  060564  046515  051101  020131
3088  060572  042524  052123  042040
3089  060600  052101  027101     000
3090  060605     045  031116  040445  HELP24: .ASCIZ  \%N2%A[I]DENTIFY ADR - USES THE REQUEST ID FUNCTION TO IDENTIFY A REMOTE
3091  060612  044533  042135  047105
3092  060620  044524  054506  040440
3093  060626  051104  026440  052440
3094  060634  042523  020123  044124
3095  060642  020105  042522  052521
3096  060650  051505  020124  042111
3097  060656  043040  047125  052103
3098  060664  047511  020116  047524
3099  060672  044440  042504  052116
3100  060700  043111  020131  020101
3101  060706  042522  047515  042524
3102  060714  047040  042117  020105
3103  060722  047117  052040  042510
3104  060730  047040  027111     000
3105  060735     045  031116  040445  HELP25: .ASCIZ  \%N2%A[SA]VE - SAVES THE CONTENTS OF THE NODE TABLE.\
3106  060742  051533  056501  042526
3107  060750  026440  051440  053101
3108  060756  051505  052040  042510
3109  060764  041440  047117  042524
3110  060772  052116  020123  043117
3111  061000  052040  042510  047040
3112  061006  042117  020105  040524
3113  061014  046102  027105     000
3114  061021     045  031116  040445  HELP26: .ASCIZ  \%N2%A[U]NSAVE - REPLACES THE CURRENT NODE TABLE WITH THE SAVED ONE.\
3115  061026  052533  047135  040523
3116  061034  042526  026440  051040
3117  061042  050105  040514  042503
3118  061050  020123  044124  020105
3119  061056  052503  051122  047105
3120  061064  020124  047516  042504
3121  061072  052040  041101  042514
3122  061100  053440  052111  020110
3123  061106  044124  020105  040523
3124  061114  042526  020104  047117
3125  061122  027105     000
3126  061125     045  022516  034123  HELP27: .ASCIZ  \%N%S8%ANOTES: 1) ADR IS THE PHYSICAL ADDRESS OF A NODE ON THE NI.\
3127  061132  040445  047516  042524
3128  061140  035123  030440  020051
3129  061146  042101  020122  051511
3130  061154  052040  042510  050040
3131  061162  054510  044523  040503
3132  061170  020114  042101  051104
```

```
 3133  061176  051505  020123  043117
 3134  061204  040440  047040  042117
 3135  061212  020105  047117  052040
 3136  061220  042510  047040  027111
 3137  061226     000
 3138  061227     045  022516  034123  HELP28: .ASCIZ  \%N%S8%A       2) PASS COUNT IS A DECIMAL NUMBER BETWEEN 1 AND 65534. A
 3139  061234  040445  020040  020040
 3140  061242  020040  031040  020051
 3141  061250  040520  051523  041440
 3142  061256  052517  052116  044440
 3143  061264  020123  020101  042504
 3144  061272  044503  040515  020114
 3145  061300  052516  041115  051105
 3146  061306  041040  052105  042527
 3147  061314  047105  030440  040440
 3148  061322  042116  033040  032465
 3149  061330  032063  020056  020101
 3150  061336  042504  040506  046125
 3151  061344  000124
 3152  061346  047045  051445  022470  HELP29: .ASCIZ  \%N%S8%A          VALUE OF 1 IS ASSUMED.\
 3153  061354  020101  020040  020040
 3154  061362  020040  020040  053040
 3155  061370  046101  042525  047440
 3156  061376  020106  020061  051511
 3157  061404  040440  051523  046525
 3158  061412  042105  000056
 3159  061416  047045  051445  022470  HELP30: .ASCIZ  \%N%S8%A          SPECIFYING -1 CAUSES THE TEST TO BE RUN INDEFINATELY.\
 3160  061424  020101  020040  020040
 3161  061432  020040  020040  051440
 3162  061440  042520  044503  054506
 3163  061446  047111  020107  030455
 3164  061454  041440  052501  042523
 3165  061462  020123  044124  020105
 3166  061470  042524  052123  052040
 3167  061476  020117  042502  051040
 3168  061504  047125  044440  042116
 3169  061512  043105  047111  052101
 3170  061520  046105  027131     000
 3171          061526                        .EVEN
 3172                                ;
 3173                                ;       TEST MESSAGES AND ARGUMENTS
 3174                                ;
 3175
 3176  061526  047045  040445  050040  PASABT: .ASCIZ  /%N%A PASS ABORTED!/
 3177  061534  051501  020123  041101
 3178  061542  051117  042524  020504
 3179  061550     000
 3180  061551     045  022516  022524  TSTMS1: .ASCIZ  /%N%T%A TEST -- /
 3181  061556  020101  042524  052123
 3182  061564  026440  020055     000
 3183  061571     045  022516  022524  TSTMS2: .ASCIZ  /%N%T%A NODE: %T/
 3184  061576  020101  047516  042504
 3185  061604  020072  052045     000
 3186  061611     045  022524  020101  TSTMS3: .ASCIZ  /%T%A ERROR/
 3187  061616  051105  047522  000122
 3188  061624  047045  052045  040445  TSTMS4: .ASCIZ  /%N%T%A NODE: %T%T%A NODE: %T/
```

```
3189  061632  047040  042117  035105
3190  061640  022440  022524  022524
3191  061646  020101  047516  042504
3192  061654  020072  052045     000
3193  061661     045  020101  020055  OK:      .ASCIZ  /%A - RESPONSE OK/
3194  061666  042522  050123  047117
3195  061674  042523  047440  000113
3196  061702  047045  040445  026440  OKRE:    .ASCIZ  /%N%A - RECEIVE ASSIST - RESPONSE OK/
3197  061710  051040  041505  044505
3198  061716  042526  040440  051523
3199  061724  051511  020124  020055
3200  061732  042522  050123  047117
3201  061740  042523  047440  000113
3202  061746  047045  040445  026440  OKTR:    .ASCIZ  /%N%A - TRANSMIT ASSIST - RESPONSE OK/
3203  061754  052040  040522  051516
3204  061762  044515  020124  051501
3205  061770  044523  052123  026440
3206  061776  051040  051505  047520
3207  062004  051516  020105  045517
3208  062012     000
3209  062013     045  022516  020101  OKFU:    .ASCIZ  /%N%A - FULL ASSIST - RESPONSE OK/
3210  062020  020055  052506  046114
3211  062026  040440  051523  051511
3212  062034  020124  020055  042522
3213  062042  050123  047117  042523
3214  062050  047440  000113
3215  062054  047045  040445  051105  MESPAT: .ASCIZ  /%N%AERROR OCCURED WITH %T%A MESSAGE TYPE/
3216  062062  047522  020122  041517
3217  062070  052503  042522  020104
3218  062076  044527  044124  022440
3219  062104  022524  020101  042515
3220  062112  051523  043501  020105
3221  062120  054524  042520     000
3222  062125     045  020101  040504  MESPA1: .ASCIZ  /%A DATA PATTERN: %T/
3223  062132  040524  050040  052101
3224  062140  042524  047122  020072
3225  062146  052045     000
3226  062151     101  046114  047040  ALLNOD: .ASCIZ  /ALL NODE/
3227  062156  042117  000105
3228  062162  047514  050117  040520  LUPAIR: .ASCIZ  /LOOPPAIR/
3229  062170  051111     000
3230  062173     114  047517  020120  DIRECT: .ASCIZ  /LOOP DIRECT/
3231  062200  044504  042522  052103
3232  062206     000
3233  062207     106  046125  020114  FULAST: .ASCIZ  /FULL ASSIST/
3234  062214  051501  044523  052123
3235  062222     000
3236  062223     124  040522  051516  TRAST:  .ASCIZ  /TRANSMIT ASSIST/
3237  062230  044515  020124  051501
3238  062236  044523  052123     000
3239  062243     122  041505  044505  RECAST: .ASCIZ  /RECEIVE ASSIST/
3240  062250  042526  040440  051523
3241  062256  051511  000124
3242  062262  042515  051523  043501  PATTRN: .ASCIZ  /MESSAGE PATTERN/
3243  062270  020105  040520  052124
3244  062276  051105  000116
```

```
3245  062302  047516  051040  051505  NORESP: .ASCIZ  /NO RESPONSE/
3246  062310  047520  051516  000105
3247  062316  054105  042503  051523  RETRY:  .ASCIZ  /EXCESSIVE COLLISION/
3248  062324  053111  020105  047503
3249  062332  046114  051511  047511
3250  062340  000116
3251  062342  042514  043516  044124  LENGTH: .ASCIZ  /LENGTH/
3252  062350     000
3253  062351     104  052101  020101  COMPAR: .ASCIZ  /DATA COMPARISON/
3254  062356  047503  050115  051101
3255  062364  051511  047117     000
3256          062372                  .EVEN
3257
3258  062372  046101  044120  000101  MSGTY0: .ASCIZ  /ALPHA/                 ;MESSAGE TYPES
3259  062400  047117  051505     000  MSGTY1: .ASCIZ  /ONES/
3260  062405     132  051105  051517  MSGTY2: .ASCIZ  /ZEROS/
3261  062412     000
3262  062413     061  046101  000124  MSGTY3: .ASCIZ  /1ALT/
3263  062420  040460  052114     000  MSGTY4: .ASCIZ  /0ALT/
3264  062425     103  044503  052124  MSGTY5: .ASCIZ  /CCITT/
3265  062432     000
3266  062433     117  042520  020122  MSGTY6: .ASCIZ  /OPER SEL/
3267  062440  042523  000114
3268  062444  054105  052111     000  CMDTY1: .ASCIZ  /EXIT/                  ;COMMAND TYPES
3269  062451     123  046525  040515  CMDTY2: .ASCIZ  /SUMMARY/
3270  062456  054522     000
3271  062461     102  044525  042114  CMDTY3: .ASCIZ  /BUILD/
3272  062466     000
3273  062467     123  047510  000127  CMDTY4: .ASCIZ  /SHOW/
3274  062474  052522  000116          CMDTY5: .ASCIZ  /RUN/
3275  062500  042515  051523  043501  CMDTY6: .ASCIZ  /MESSAGE/
3276  062506  000105
3277  062510  047516  042504     000  CMDTY7: .ASCIZ  /NODE/
3278  062515     103  042514  051101  CMDTY8: .ASCIZ  /CLEAR/
3279  062522     000
3280  062523     122  050505  042525  CMDTY9: .ASCIZ  /REQUEST ID/
3281  062530  052123  044440  000104
3282  062536  047516  042504  000123  ARGTY1: .ASCIZ  /NODES/                 ;ARGUMENT TYPES
3283  062544  042515  051523  043501  ARGTY2: .ASCIZ  /MESSAGES/
3284  062552  051505     000
3285  062555     103  052517  052116  ARGTY3: .ASCIZ  /COUNTERS/
3286  062562  051105  000123
3287  062566  047514  050117  040520  ARGTY4: .ASCIZ  /LOOPPAIR/
3288  062574  051111     000
3289  062577     101  046114     000  ARGTY5: .ASCIZ  /ALL/
3290  062603     040  051501  044523  ARGTY6: .ASCIZ  / ASSIST/
3291  062610  052123     000
3292  062613     124  051101  042507  ARGTY7: .ASCIZ  /TARGET/
3293  062620  000124
3294                                  .EVEN
3295  062622  047045  040445  040502  NOCLK:  .ASCIZ  /%N%ABAD CLOCK - PROGRAM WILL HANG ON ''TIMEOUT''!!/
3296  062630  020104  046103  041517
3297  062636  020113  020055  051120
3298  062644  043517  040522  020115
3299  062652  044527  046114  044040
3300  062660  047101  020107  047117
```

```
3301  062666  021040  044524  042515
3302  062674  052517  021124  020441
3303  062702     000
3304
3305                                  ;
3306                                  ;       UNA COUNTER INFORMATION MESSAGES
3307                                  ;
3308
3309  062703     045  022516  032523  CNTR00: .ASCIZ  /%N%S5%ACONTENTS OF NODE %T%A INTERNAL COUNTERS:/
3310  062710  040445  047503  052116
3311  062716  047105  051524  047440
3312  062724  020106  047516  042504
3313  062732  022440  022524  020101
3314  062740  047111  042524  047122
3315  062746  046101  041440  052517
3316  062754  052116  051105  035123
3317  062762     000
3318  062763     045  031116  040445  CNTR01: .ASCIZ  /%N2%ASECONDS SINCE LAST ZEROED:%S15%Z5/
3319  062770  042523  047503  042116
3320  062776  020123  044523  041516
3321  063004  020105  040514  052123
3322  063012  055040  051105  042517
3323  063020  035104  051445  032461
3324  063026  055045  000065
3325  063032  047045  040445  040520  CNTR02: .ASCIZ  /%N%APACKETS RECEIVED:%S19%T/
3326  063040  045503  052105  020123
3327  063046  042522  042503  053111
3328  063054  042105  022472  030523
3329  063062  022471  000124
3330  063066  047045  040445  052515  CNTR03: .ASCIZ  /%N%AMULTICAST PACKETS RECEIVED:%S9%T/
3331  063074  052114  041511  051501
3332  063102  020124  040520  045503
3333  063110  052105  020123  042522
3334  063116  042503  053111  042105
3335  063124  022472  034523  052045
3336  063132     000
3337  063133     045  022516  050101  CNTR04: .ASCIZ  /%N%APACKETS REC'D WITH ERROR - BITMAP:%S9%B3/
3338  063140  041501  042513  051524
3339  063146  051040  041505  042047
3340  063154  053440  052111  020110
3341  063162  051105  047522  020122
3342  063170  020055  044502  046524
3343  063176  050101  022472  034523
3344  063204  041045  000063
3345  063210  047045  040445  040520  CNTR05: .ASCIZ  /%N%APACKETS RECEIVED WITH ERROR:%S13%Z5/
3346  063216  045503  052105  020123
3347  063224  042522  042503  053111
3348  063232  042105  053440  052111
3349  063240  020110  051105  047522
3350  063246  035122  051445  031461
3351  063254  055045  000065
3352  063260  047045  040445  040504  CNTR06: .ASCIZ  /%N%ADATA BYTES RECEIVED:%S16%T/
3353  063266  040524  041040  052131
3354  063274  051505  051040  041505
3355  063302  044505  042526  035104
3356  063310  051445  033061  052045
```

```
3357  063316     000
3358  063317     045  022516  046501  CNTR07: .ASCIZ  /%N%AMULTICAST DA'' BYTES RECEIVED:%S6%T/
3359  063324  046125  044524  040503
3360  063332  052123  042040  052101
3361  063340  020101  054502  042524
3362  063346  020123  042522  042503
3363  063354  053111  042105  022472
3364  063362  033123  052045     000
3365  063367     045  022516  051101  CNTR08: .ASCIZ  /%N%ARECEIVED PACKETS LOST-INTERNAL:%S10%Z5/
3366  063374  041505  044505  042526
3367  063402  020104  040520  045503
3368  063410  052105  020123  047514
3369  063416  052123  044455  052116
3370  063424  051105  040516  035114
3371  063432  051445  030061  055045
3372  063440  000065
3373  063442  047045  040445  042522  CNTR09: .ASCIZ  /%N%ARECEIVED PACKETS LOST -LOCAL:%S12%Z5/
3374  063450  042503  053111  042105
3375  063456  050040  041501  042513
3376  063464  051524  046040  051517
3377  063472  020124  046055  041517
3378  063500  046101  022472  030523
3379  063506  022462  032532     000
3380  063513     045  022516  050101  CNTR10: .ASCIZ  /%N%APACKETS TRANSMITTED:%S16%T/
3381  063520  041501  042513  051524
3382  063526  052040  040522  051516
3383  063534  044515  052124  042105
3384  063542  022472  030523  022466
3385  063550  000124
3386  063552  047045  040445  052515  CNTR11: .ASCIZ  /%N%AMULTICAST PACKETS TRANSMITTED:%S6%T/
3387  063560  052114  041511  051501
3388  063566  020124  040520  045503
3389  063574  052105  020123  051124
3390  063602  047101  046523  052111
3391  063610  042524  035104  051445
3392  063616  022466  000124
3393  063622  047045  040445  040520  CNTR12: .ASCIZ  /%N%APACKETS TRANSMITTED 3+ TRYS:%S8%T/
3394  063630  045503  052105  020123
3395  063636  051124  047101  046523
3396  063644  052111  042524  020104
3397  063652  025463  052040  054522
3398  063660  035123  051445  022470
3399  063666  000124
3400  063670  047045  040445  040520  CNTR13: .ASCIZ  /%N%APACKETS TRANSMITTED 2 TRYS:%S9%T/
3401  063676  045503  052105  020123
3402  063704  051124  047101  046523
3403  063712  052111  042524  020104
3404  063720  020062  051124  051531
3405  063726  022472  034523  052045
3406  063734     000
3407  063735     045  022516  050101  CNTR14: .ASCIZ  /%N%APACKETS DEFFERED:%S19%T/
3408  063742  041501  042513  051524
3409  063750  042040  043105  042506
3410  063756  042522  035104  051445
3411  063764  034461  052045     000
3412  063771     045  022516  042101  CNTR15: .ASCIZ  /%N%ADATA BYTES TRANSMITTED:%S13%T/
```

```
3413  063776  052101  020101  054502
3414  064004  042524  020123  051124
3415  064012  047101  046523  052111
3416  064020  042524  035104  051445
3417  064026  031461  052045     000
3418  064033     045  022516  046501  CNTR16: .ASCIZ  /%N%AMULTICAST BYTES TRANSMITTED:%S8%T/
3419  064040  046125  044524  040503
3420  064046  052123  041040  052131
3421  064054  051505  052040  040522
3422  064062  051516  044515  052124
3423  064070  042105  022472  034123
3424  064076  052045     000
3425  064101     045  022516  052101  CNTR17: .ASCIZ  /%N%ATRANSMIT PACKETS ABORTED-BITMAP:%S8%B6/
3426  064106  040522  051516  044515
3427  064114  020124  040520  045503
3428  064122  052105  020123  041101
3429  064130  051117  042524  026504
3430  064136  044502  046524  050101
3431  064144  022472  034123  041045
3432  064152  000066
3433  064154  047045  040445  051124  CNTR18: .ASCIZ  /%N%ATRANSMIT PACKETS ABORTED:%S16%Z5/
3434  064162  047101  046523  052111
3435  064170  050040  041501  042513
3436  064176  051524  040440  047502
3437  064204  052122  042105  022472
3438  064212  030523  022466  032532
3439  064220     000
3440  064221     045  022516  054101  CNTR19: .ASCIZ  /%N%AXMIT COLLISION CHECK FAILURE:%S12%Z5/
3441  064226  044515  020124  047503
3442  064234  046114  051511  047511
3443  064242  020116  044103  041105
3444  064250  020113  040506  046111
3445  064256  051125  035105  051445
3446  064264  031061  055045  000065
3447
3448                                  ;
3449                                  ;       ERROR MESSAGES FOR DEUNA DRIVER
3450                                  ;
3451
3452  064272  047125  020101  047520  EMSG01: .ASCIZ  /UNA PORT COMMAND ERROR/
3453  064300  052122  041440  046517
3454  064306  040515  042116  042440
3455  064314  051122  051117     000
3456  064321     125  040516  043040  EMSG02: .ASCIZ  /UNA FATAL ERROR/
3457  064326  052101  046101  042440
3458  064334  051122  051117     000
3459  064341     125  042516  050130  EMSG03: .ASCIZ  /UNEXPLAINED UNA INTERRUPT/
3460  064346  040514  047111  042105
3461  064354  052440  040516  044440
3462  064362  052116  051105  052522
3463  064370  052120     000
3464  064373     125  045516  047516  EMSG04: .ASCIZ  /UNKNOWN UNA ERROR/
3465  064400  047127  052440  040516
3466  064406  042440  051122  051117
3467  064414     000
3468  064415     125  040516  053440  EMSG05: .ASCIZ  /UNA WON'T READ PCB ADDRESS/
```

```
3469  064422  047117  052047  051040
3470  064430  040505  020104  041520
3471  064436  020102  042101  051104
3472  064444  051505  000123
3473  064450  047125  041101  042514  EMSG06: .ASCIZ  /UNABLE TO READ PHYSICAL ADDRESS/
3474  064456  052040  020117  042522
3475  064464  042101  050040  054510
3476  064472  044523  040503  020114
3477  064500  042101  051104  051505
3478  064506  000123
3479  064510  047125  020101  044527  EMSG07: .ASCIZ  /UNA WILL NOT GO INTO RUNNING STATE/
3480  064516  046114  047040  052117
3481  064524  043440  020117  047111
3482  064532  047524  051040  047125
3483  064540  044516  043516  051440
3484  064546  040524  042524     000
3485  064553     124  046511  047505  EMSG08: .ASCIZ  /TIMEOUT!--TRANSMIT FLAG NOT SET/
3486  064560  052125  026441  052055
3487  064566  040522  051516  044515
3488  064574  020124  046106  043501
3489  064602  047040  052117  051440
3490  064610  052105     000
3491  064613     105  051122  051117  EMSG09: .ASCIZ  /ERROR DURING TRANSMIT PDMD COMMAND/
3492  064620  042040  051125  047111
3493  064626  020107  051124  047101
3494  064634  046523  052111  050040
3495  064642  046504  020104  047503
3496  064650  046515  047101  000104
3497  064656  051124  047101  046523  EMSG10: .ASCIZ  /TRANSMIT RING BOOKKEEPING ERROR/
3498  064664  052111  051040  047111
3499  064672  020107  047502  045517
3500  064700  042513  050105  047111
3501  064706  020107  051105  047522
3502  064714  000122
3503  064716  042522  042503  053111  EMSG11: .ASCIZ  /RECEIVE RING BOOKKEEPING ERROR/
3504  064724  020105  044522  043516
3505  064732  041040  047517  045513
3506  064740  042505  044520  043516
3507  064746  042440  051122  051117
3508  064754     000
3509  064755     115  051505  040523  EMSG14: .ASCIZ  /MESSAGE SIZE TOO BIG FOR MAX. PACKET LENGTH/
3510  064762  042507  051440  055111
3511  064770  020105  047524  020117
3512  064776  044502  020107  047506
3513  065004  020122  040515  027130
3514  065012  050040  041501  042513
3515  065020  020124  042514  043516
3516  065026  044124     000
3517  065031     104  044516  042040  EMSG15: .ASCIZ  /DNI DID NOT SET FROM RESET/
3518  065036  042111  047040  052117
3519  065044  051440  052105  043040
3520  065052  047522  020115  042522
3521  065060  042523  000124
3522  065064  047125  020101  044527  EMSG16: .ASCIZ  /UNA WILL NOT READ DESCRIPTOR RINGS/
3523  065072  046114  047040  052117
3524  065100  051040  040505  020104
```

```
3525  065106  042504  041523  044522
3526  065114  052120  051117  051040
3527  065122  047111  051507     000
3528  065127     103  047101  052047  EMSG18: .ASCIZ  /CAN'T GET INITIAL STATUS INFO FROM UNA/
3529  065134  043440  052105  044440
3530  065142  044516  044524  046101
3531  065150  051440  040524  052524
3532  065156  020123  047111  047506
3533  065164  043040  047522  020115
3534  065172  047125  000101
3535  065176  042515  051523  043501  EMSG19: .ASCIZ  /MESSAGE DATA COMPARISON ERROR/
3536  065204  020105  040504  040524
3537  065212  041440  046517  040520
3538  065220  044522  047523  020116
3539  065226  051105  047522  000122
3540  065234  047524  040524  020114  EMSG20: .ASCIZ  /TOTAL DATA COMPARE ERRORS/
3541  065242  040504  040524  041440
3542  065250  046517  040520  042522
3543  065256  042440  051122  051117
3544  065264  000123
3545  065266  047045  040445  047516  EMSG22: .ASCIZ  /%N%ANO RESPONSE FROM NODE./
3546  065274  051040  051505  047520
3547  065302  051516  020105  051106
3548  065310  046517  047040  042117
3549  065316  027105     000
3550  065321     105  051122  051117  EMSG23: .ASCIZ  /ERROR WHILE ATTEMPTING TO WRITE MODE/
3551  065326  053440  044510  042514
3552  065334  040440  052124  046505
3553  065342  052120  047111  020107
3554  065350  047524  053440  044522
3555  065356  042524  046440  042117
3556  065364  000105
3557  065366  051124  047101  046523  EMSG24: .ASCIZ  /TRANSMIT ERROR, ALL PACKETS NOT TRANSMITTED/
3558  065374  052111  042440  051122
3559  065402  051117  020054  046101
3560  065410  020114  040520  045503
3561  065416  052105  020123  047516
3562  065424  020124  051124  047101
3563  065432  046523  052111  042524
3564  065440  000104
3565  065442  051105  047522  020122  EMSG25: .ASCIZ  /ERROR WHILE ATTEMPTING TO WRITE MULTICAST ADDRESS LIST/
3566  065450  044127  046111  020105
3567  065456  052101  042524  050115
3568  065464  044524  043516  052040
3569  065472  020117  051127  052111
3570  065500  020105  052515  052114
3571  065506  041511  051501  020124
3572  065514  042101  051104  051505
3573  065522  020123  044514  052123
3574  065530     000
3575  065531     105  051122  051117  EMSG30: .ASCIZ  /ERROR WHILE ATTEMPTING PORT FUNCTION/
3576  065536  053440  044510  042514
3577  065544  040440  052124  046505
3578  065552  052120  047111  020107
3579  065560  047520  052122  043040
3580  065566  047125  052103  047511
```

```
3581  065574  000116
3582  065576  047125  041101  042514  EMSG31: .ASCIZ  /UNABLE TO READ INTERNAL COUNTERS/
3583  065604  052040  020117  042522
3584  065612  042101  044440  052116
3585  065620  051105  040516  020114
3586  065626  047503  047125  042524
3587  065634  051522     000
3588  065637     045  022516  044501  EMSG32: .ASCIZ  \%N%AILLEGAL TARGET/ASSIST PAIR IN NODE TABLE\
3589  065644  046114  043505  046101
3590  065652  052040  051101  042507
3591  065660  027524  051501  044523
3592  065666  052123  050040  044501
3593  065674  020122  047111  047040
3594  065702  042117  020105  040524
3595  065710  046102  000105
3596  065714  044524  042515  052517  EMSG33: .ASCIZ  /TIMEOUT ERROR/
3597  065722  020124  051105  047522
3598  065730  000122
3599  065732  005015  044524  042515  EMSG34: .ASCIZ  <15><12>/TIMEOUT OCCURED BEFORE LOOPBACK REPLY/
3600  065740  052517  020124  041517
3601  065746  052503  042522  020104
3602  065754  042502  047506  042522
3603  065762  046040  047517  041120
3604  065770  041501  020113  042522
3605  065776  046120  000131
3606  066002  040445  040506  046111  EMSG35: .ASCIZ  /%AFAILING NODE ADDRESS: %T%N/
3607  066010  047111  020107  047516
3608  066016  042504  040440  042104
3609  066024  042522  051523  020072
3610  066032  052045  047045     000
3611  066037     045  042101  052101  EMSG36: .ASCIZ  /%ADATA PATTERN: %T%N/
3612  066044  020101  040520  052124
3613  066052  051105  035116  022440
3614  066060  022524  000116
3615  066064  040445  040506  046111  EMSG37: .ASCIZ  /%AFAILING TARGET NODE ADDRESS: %T%N/
3616  066072  047111  020107  040524
3617  066100  043522  052105  047040
3618  066106  042117  020105  042101
3619  066114  051104  051505  035123
3620  066122  022440  022524  000116
3621  066130  040445  040506  046111  EMSG38: .ASCIZ  /%AFAILING ASSIST NODE ADDRESS: %T%N/
3622  066136  047111  020107  051501
3623  066144  044523  052123  047040
3624  066152  042117  020105  042101
3625  066160  051104  051505  035123
3626  066166  022440  022524  000116
3627  066174  005015  044524  042515  EMSG40: .ASCIZ  <15><12>/TIMEOUT OCCURED - LOOP MESSAGE TYPE - RECEIVE ASSIST/
3628  066202  052517  020124  041517
3629  066210  052503  042522  020104
3630  066216  020055  047514  050117
3631  066224  046440  051505  040523
3632  066232  042507  052040  050131
3633  066240  020105  020055  042522
3634  066246  042503  053111  020105
3635  066254  051501  044523  052123
3636  066262     000
```

```
3637  066263     015  052012  046511  EMSG41: .ASCIZ  <15><12>/TIMEOUT OCCURED - LOOP MESSAGE TYPE - TRANSMIT ASSIST/
3638  066270  047505  052125  047440
3639  066276  041503  051125  042105
3640  066304  026440  046040  047517
3641  066312  020120  042515  051523
3642  066320  043501  020105  054524
3643  066326  042520  026440  052040
3644  066334  040522  051516  044515
3645  066342  020124  051501  044523
3646  066350  052123     000
3647  066353     015  052012  046511  EMSG42: .ASCIZ  <15><12>/TIMEOUT OCCURED - LOOP MESSAGE TYPE - FULL ASSIST/
3648  066360  047505  052125  047440
3649  066366  041503  051125  042105
3650  066374  026440  046040  047517
3651  066402  020120  042515  051523
3652  066410  043501  020105  054524
3653  066416  042520  026440  043040
3654  066424  046125  020114  051501
3655  066432  044523  052123     000
3656
3657          066440                  .EVEN
3658  066440  047045  040445  020040  SIMSG1: .ASCIZ  /%N%A  NODE DEFAULT ADDRESS: %T/
3659  066446  047516  042504  042040
3660  066454  043105  052501  052114
3661  066462  040440  042104  042522
3662  066470  051523  020072  052045
3663  066476     000
3664  066477     045  022516  034123  SIMSG2: .ASCIZ  /%N%S8%ARECEIPT NUMBER: %06/
3665  066504  040445  042522  042503
3666  066512  050111  020124  052516
3667  066520  041115  051105  020072
3668  066526  047445  000066
3669  066532  047045  040445  020040  SIMSG3: .ASCIZ  /%N%A   MAINTENANCE VERSION: %Z2/
3670  066540  046440  044501  052116
3671  066546  047105  047101  042503
3672  066554  053040  051105  044523
3673  066562  047117  020072  055045
3674  066570  000062
3675  066572  047045  051445  034461  SIMSG4: .ASCIZ  /%N%S19%AECO: %Z2/
3676  066600  040445  041505  035117
3677  066606  022440  031132     000
3678  066613     045  022516  030523  SIMSG5: .ASCIZ  /%N%S14%AUSER ECO: %Z2/
3679  066620  022464  052501  042523
3680  066626  020122  041505  035117
3681  066634  022440  031132     000
3682  066641     045  022516  030523  SIMSG6: .ASCIZ  /%N%S14%AFUNCTION: %02/
3683  066646  022464  043101  047125
3684  066654  052103  047511  035116
3685  066662  022440  031117     000
3686  066667     045  022516  030523  SIMSG7: .ASCIZ  /%N%S16%ADEVICE: %02/
3687  066674  022466  042101  053105
3688  066702  041511  035105  022440
3689  066710  031117     000
3690
3691          066714                  .EVEN
3692  066714  047045  040445  041520  PCMSG:: .ASCIZ  /%N%APC OF CALLING ROUTINE = %06/
```

```
 3693  066722  047440  020106  040503
 3694  066730  046114  047111  020107
 3695  066736  047522  052125  047111
 3696  066744  020105  020075  047445
 3697  066752  000066
 3698                                  .EVEN
 3699  066754  047045  040445  054502  CMPER1: .ASCIZ  /%N%ABYTE NUMBER:%D4%A(D) EXPECTED=%O6%A(O) RECIEVED=%O6%A(O)/
 3700  066762  042524  047040  046525
 3701  066770  042502  035122  042045
 3702  066776  022464  024101  024504
 3703  067004  042440  050130  041505
 3704  067012  042524  036504  047445
 3705  067020  022466  024101  024517
 3706  067026  051040  041505  042511
 3707  067034  042526  036504  047445
 3708  067042  022466  024101  024517
 3709  067050     000
 3710  067051     045  022516  052101  CMPER2: .ASCIZ  /%N%ATOTAL MISMATCHES IN MESSAGE = %D4/
 3711  067056  052117  046101  046440
 3712  067064  051511  040515  041524
 3713  067072  042510  020123  047111
 3714  067100  046440  051505  040523
 3715  067106  042507  036440  022440
 3716  067114  032104     000
 3717  067117     045  022516  046101  LGERMS: .ASCIZ  /%N%ALENGTH ERROR -- BYTES EXPECTED: %O6%A BYTES RECEIVED: %O6/
 3718  067124  047105  052107  020110
 3719  067132  051105  047522  020122
 3720  067140  026455  041040  052131
 3721  067146  051505  042440  050130
 3722  067154  041505  042524  035104
 3723  067162  022440  033117  040445
 3724  067170  041040  052131  051505
 3725  067176  051040  041505  044505
 3726  067204  042526  035104  022440
 3727  067212  033117     000
 3728  067215     045  031116  051445  SUMMS1: .ASCIZ  /%N2%S8%ANODE: %T/
 3729  067222  022470  047101  042117
 3730  067230  035105  022440  000124
 3731  067236  047045  040445  054122  SUMMS2: .ASCIZ  /%N%AxX NOT COMPLETE    RX COMPLETE    LENGTH ERRORS/
 3732  067244  047040  052117  041440
 3733  067252  046517  046120  052105
 3734  067260  020105  020040  051040
 3735  067266  020130  047503  050115
 3736  067274  042514  042524  020040
 3737  067302  020040  042514  043516
 3738  067310  044124  042440  051122
 3739  067316  051117  000123
 3740  067322  047045  051445  022466  SUMMS3: .ASCIZ  /%N%S6%Z5%S12%Z5%S10%Z5/
 3741  067330  032532  051445  031061
 3742  067336  055045  022465  030523
 3743  067344  022460  032532     000
 3744  067351     045  022516  041501  SUMMS4: .ASCIZ  /%N%ACOMPARE ERRORS    BYTES COMPARED    BYTES XFER/
 3745  067356  046517  040520  042522
 3746  067364  042440  051122  051117
 3747  067372  020123  020040  041040
 3748  067400  052131  051505  041440
```

```
3749  067406  046517  040520  042522
3750  067414  020104  020040  041040
3751  067422  052131  051505  054040
3752  067430  042506  000122
3753  067434  047045  051445  022466  SUMMS5: .ASCIZ  /%N%S6%Z5%S8%T/
3754  067442  032532  051445  022470
3755  067450  000124
3756  067452  051445  022465  000124  SUMMS6: .ASCIZ  /%S5%T/
3757                                  .EVEN
3758
3759
```

```
                                .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.
                                ;--

067460                          BGNMSG  ERR1                                    ERR1::
067460  012737  000020  003672          MOV     #CEXIT,CFLAG
067466                          PRINTX  #PCMSG,PCCALL                                   MOV     PCCALL,-(SP)
067466  013746  050564                                                                  MOV     #PCMSG,-(SP)
067472  012746  066714                                                                  MOV     #2,-(SP)
067476  012746  000002                                                                  MOV     SP,R0
067502  010600                                                                          TRAP    C$PNTX
067504  104415                                                                          ADD     #6,SP
067506  062706  000006
067512                          DOCLN
067512  104444                                                                  L10002: TRAP    C$DCLN
067514                          ENDMSG
067514                                                                          L10002:
067514  104423                                                                          TRAP    C$MSG

067516                          BGNMSG  ERR2                                    ERR2::
067516                                                                          ERR2::
067516  010146                  MOV     R1,-(SP)
067520  013701  002370          MOV     P$TYPE,R1
067524  006301                  ASL     R1
067526  062701  003262          ADD     #MSGTAB,R1
067532                          PRINTX  #EMSG35,#STRBUF
067532  012746  002322                                                                  MOV     #STRBUF,-(SP)
067536  012746  066002                                                                  MOV     #EMSG35,-(SP)
067542  012746  000002                                                                  MOV     #2,-(SP)
067546  010600                                                                          MOV     SP,R0
067550  104415                                                                          TRAP    C$PNTX
067552  062706  000006                                                                  ADD     #6,SP
067556                          PRINTX  #EMSG36,(R1)
067556  011146                                                                          MOV     (R1),-(SP)
067560  012746  066037                                                                  MOV     #EMSG36,-(SP)
067564  012746  000002                                                                  MOV     #2,-(SP)
067570  010600                                                                          MOV     SP,R0
067572  104415                                                                          TRAP    C$PNTX
067574  062706  000006                                                                  ADD     #6,SP
067600  012601                  MOV     (SP)+,R1
067602                          ENDMSG
067602                                                                          L10003:
067602  104423                                                                          TRAP    C$MSG

067604                          BGNMSG  ERR3                                    ERR3::
067604                                                                          ERR3::
067604                          PRINTX  #EMSG37,#STRBUF
067604  012746  002322                                                                  MOV     #STRBUF,-(SP)
067610  012746  066064                                                                  MOV     #EMSG37,-(SP)
067614  012746  000002                                                                  MOV     #2,-(SP)
```

```
067620 010600                                                                  MOV     SP,R0
067622 104415                                                                  TRAP    C$PNTX
067624 062706 000006                                                           ADD     #6,SP
067630                          PRINTX  #EMSG38,#STRBU1
067630 012746 002344                                                           MOV     #STRBU1,-(SP)
067634 012746 066130                                                           MOV     #EMSG38,-(SP)
067640 012746 000002                                                           MOV     #2,-(SP)
067644 010600                                                                  MOV     SP,R0
067646 104415                                                                  TRAP    C$PNTX
067650 062706 000006                                                           ADD     #6,SP
067654                  ENDMSG
067654                                                                 L10004:
067654 104423                                                                  TRAP    C$MSG

                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        ;       THESE MESSAGE AREAS ARE USED TO OUTPUT SUPPLEMENTARY INFORMATION
                        ;       AFTER AN ERROR CALL.  THEY ARE INVOKED BY APPENDING THE NAME
                        ;       OF THE AREA TO AN ERROR CALL:  ERRXXX 1,ERRORMESSAGE,AREANAME.
                        ;       THE CORRESPONDING MESSAGE AREA IS SET UP IN THIS SECTION:
                        ;               BGNMSG  AREANAME
                        ;               [CODE]
                        ;               ENDMSG
                        ;
                        ;       THE AREAS IN THIS SECTION ARE FOR MESSAGES USED IN MORE THAN ONE
                        ;       TEST.  USE THE PRINTB (PRINT BASIC) AND PRINTX (PRINT EXTENDED)
                        ;       MACROS.
                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
.SBTTL  GLOBAL SUBROUTINES SECTION

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

;++
; FUNCTIONAL DESCRIPTION:
;     SUBROUTINE TO....

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;       COMPLETE THE "SUBROUTINE TO...." STATEMENT WITH A FUNCTIONAL
;       DESCRIPTION OF THIS SUBROUTINE.
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

; INPUTS:

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;       LIST THE INPUT DATA THAT ARE EXPLICITLY PASSED TO THIS SUBROUTINE.
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

; IMPLICIT INPUTS:

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;       LIST THE INPUT DATA THAT ARE IMPLICITLY USED BY THIS SUBROUTINE;
;       FOR EXAMPLE, DATA READ FROM COMMON AREAS.
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

; OUTPUTS:

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;       LIST THE OUTPUT DATA THAT ARE EXPLICITLY GIVEN BY THIS SUBROUTINE
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

; IMPLICIT OUTPUTS:

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;       LIST THE OUTPUT DATA THAT ARE IMPLICITLY GIVEN BY THIS SUBROUTINE;
;       FOR EXAMPLF, DATA STORED IN COMMON AREAS.
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

; SUBORDINATE ROUTINES USED:

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;       LIST THE SUBROUTINES CALLED BY THIS SUBROUTINE.
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

; FUNCTIONAL SIDE EFFECTS:

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;       DESCRIBE ANY EFFECTS THIS SUBROUTINE MAY HAVE UPON OTHER
;       MODULES OF THE DIAGNOSTIC PROGRAM.  AN EXAMPLE OF THIS IS
;       THE SUBROUTINE INHIBITS ALL INTERRUPTS WITH PRIORITY 7.
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
                        ; CALLING SEQUENCE:

                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        ;        GIVE THE EXACT CALLING SEQUENCE USED TO ACCESS THIS SUBROUTINE.
                        ;        FOR EXAMPLE:    MOV COUNT,R1    ;MOVE INPUT TO R1
                        ;                        JSR     PC,ROUTINE       ;GO TO ROUTINE
                        ;                        BCS     ERROR            ;CARRY SET IF ROUTINE HAD ERROR
                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        ;--

                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        ;      INSERT THE CODE FOR THIS SUBROUTINE.   THE NAME OF THE SUBROUTINE SHOULD
                        ;      BE DEFINED WITH A DOUBLE-COLON (::);  THIS WILL MAKE THE SUBROUTINE GLOBAL.
                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        ;      BEGIN EACH SUBROUTINE AT THE TOP OF A NEW PAGE.
                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                        .SBTTL  CLKSET  CLOCK SETUP SUBROUTINE

                        ;--+
                        ; FUNCTIONAL DESCRIPTION:
                        ;                 THIS SUBROUTINE SETS UP THE CLOCK INFORMATION TABLE FOLLOWING
                        ;                 A "CLOCK" CALL EXECUTED IN THE INITIALIZATION CODE.  BUT SINCE
                        ;                 THE "CLOCK" CALL SAYS NOTHING ABOUT AN LSI-11'S CLOCK, THE
                        ;                 ROUTINE IS ONLY USED IF A LINE OR P-CLOCK IS FOUND.
                        ;
                        ; INPUTS -        R1 - POINTS TO SUPERVISOR SPACE WHERE CLOCK INFO WAS RETURNED
                        ;                 R2 - POINTS TO "CLK" TABLE WHERE CLOCK INFO WILL BE KEPT
                        ;
                        ; OUTPUTS -       "CLKCSR" GETS LOADED WITH THE CLOCK'D CSR ADDRESS
                        ;                 "CLKBR" GETS LOADED WITH THE CLOCK'S INTERRUPT LEVEL
                        ;                 "CLKVEC" GETS LOADED WITH THE CLOCK'S INTERRUPT VECTOR
                        ;                 "CLKHZ" GETS LOADED WITH THE LINE FREQ. (IN HERTZ)
                        ;
                        ; CALLING PROCEDURE:
                        ;                 JSR     PC,CLKSET       ; CALL CLOCK SETUP WITH R1 AND R2 SETUP
                        ;
                        ;--+
067656                  CLKSET:
067656  012122                  MOV     (R1)+,(R2)+             ; LOAD CLOCK'S CSR ADDR. INTO "CLKCSR"
067660  012112                  MOV     (R1)+,(R2)              ; LOAD CLOCK'S INTR. LEVEL INTO "CLKBR"
067662  006312                  ASL     (R2)                    ; ADJUST THE INTR. LEVEL FOR LOADING
067664  006312                  ASL     (R2)                    ;   INTO THE PSW WITH A "SETVEC" CALL
067666  006312                  ASL     (R2)
067670  006312                  ASL     (R2)
067672  006322                  ASL     (R2)+
067674  012122                  MOV     (R1)+,(R2)+             ; LOAD CLOCK'S INTR. VECTOR INTO "CLKVEC"
067676  012122                  MOV     (R1)+,(R2)+             ; LOAD CLOCK'S FREQ. INTO "CLKHZ"
067700  000207                  RTS     PC
```

```
3953
3954                                    .SBTTL  CLKINT  CLOCK INTERRUPT SERVIDE ROUTINE
3955
3956                                    ;--+
3957                                    ; FUNCTIONAL DESCRIPTION:
3958                                    ;               THIS IS THE CLOCK INTERRUPT SERVICE ROUTINE WHICH TAKES CARE
3959                                    ;               OF KEEPING THE "TIME-SINCE-START" AND COUNTING DOWN ANY OF THE
3960                                    ;               "EVENT" TIMERS.  THE TIMERS ARE USED TO TIME COMPLETION OF
3961                                    ;               DEVICE REQUESTS.  THE "TIME-SINCE-START" IS USED TO BE LOGGED
3962                                    ;               WITH EACH ENTRY INTO THE EVENT LOG.
3963                                    ;
3964                                    ; IMPLICIT INPUTS - TIMTCK - THE CURRENT NO. OF TICKS LEFT TO BE COUNTED UNTIL
3965                                    ;                     A SECOND HAS BEEN COUNTED OFF
3966                                    ;                   CLKHZ - THE NO. OF TICKS IN A SECOND, DETERMINED BY THE
3967                                    ;                     SYS. LINE FREQ.
3968                                    ;                   TIMMIN & TIMSEC - CURRENT VALUE OF "TIME-SINCE-START" IN
3969                                    ;                     MINUTES AND SECONDS
3970                                    ;                   TIMER 1,2 AND S - CURRENT VALUES OF "EVENT TIMERS"
3971                                    ;
3972                                    ; IMPLICIT OUTPUTS - NEW VALUE OF EVENT TIMER "1" & "2" DECREMENTED BY 1 TICK
3973                                    ;                       IF IT WAS NON-ZERO
3974                                    ;                    NEW VALUE OF EVENT TIMER "S" DECREMENTED BY 1 SECOND IF IT
3975                                    ;                       WAS NON-ZERO
3976                                    ;
3977                                    ; SIDE EFFECTS -        THE CLOCK IS DISABLED UPON ENTRY AND REENABLED WHEN LEAVING
3978                                    ;
3979                                    ; CALLING PROCEDURE - THIS ROUTINE IS CALLED WHEN THE CLODK INTERRUPTS THRU
3980                                    ;                       "CLKVEC".  THE ADDRESS OF THIS ROUTINE WAS LOADED
3981                                    ;                       INTO THE CLOCK'S INTERRUPT VECTOR WITH A "SETVEC" CALL
3982                                    ;
3983                                    ;--+
3984
3985  067702                                    BGNSRV  CLKINT
3986  067702                                                                              CLKINT::
3987
3988  067702  005077  113766                    CLR     @CLKCSR                 ; DISABLE THE CLOCK FROM INTERRUPTING
3989  067706  005337  003712                    DEC     TIMTCK                  ; DECREMENT THE NO. OF TICKS/SEC
3990  067712  001015                            BNE     1$                      ; GO CHECK TIMERS
3991  067714  013737  003702  003712            MOV     CLKHZ,TIMTCK            ; RESET THE NO. OF TICKS/SEC.
3992  067722  005237  003710                    INC     TIMSEC                  ; INC. NO OF SECS-SINCE-START
3993  067726  022737  000074  003710            CMP     #60.,TIMSEC             ; SEE IF WE'VE COUNTED 60 SEC.S YET
3994  067734  001004                            BNE     1$                      ; IF NOT, GO CHECK TIMERS
3995  067736  005237  003706                    INC     TIMMIN                  ;  ELSE, INC. MINUTES-SINCE-START
3996  067742  005037  003710                    CLR     TIMSEC                  ;  AND RESTART SECOND COUNTER
3997
3998  067746  005737  003714          1$:       TST     TIMER1                  ; SEE IF TIMER1 TIMING ANYTHING
3999  067752  001402                            BEQ     2$                      ;  IF=0, NO, CHECK NEXT TIMER
4000  067754  005337  003714                    DEC     TIMER1                  ;  ELSE DECREMENT THE TIMER VALUE (BY 1 TICK)
4001  067760  005737  003716          2$:       TST     TIMER2                  ; SEE IF TIMER2 TIMING ANYTHING
4002  067764  001402                            BEQ     3$                      ;  IF=0, NO, CHECK NEXT TIMER
4003  067766  005337  003716                    DEC     TIMER2                  ;  ELSE DECREMENT TIMER VALUE (BY 1 TICK)
4004  067772  005737  003720          3$:       TST     TIMERS                  ; SEE IF TIMERS TIMING ANYTHING
4005  067776  001406                            BEQ     4$                      ;  IF=0, NOTHING BE TIMED, LEAVE
4006  070000  023737  003702  003712            CMP     CLKHZ,TIMTCK            ; SEE IF A SECOND HAS BEEN COUNTED OFF
4007  070006  001002                            BNE     4$                      ;  BR IF NO
4008  070010  005337  003720                    DEC     TIMERS                  ;  ELSE, DECREMENT TIMER VALUE (BY 1 SEC.)
```

```
4009  070014  013777  003704  113652  4$:     MOV     CLKEN,@CLKCSR           ; REENABLE THE CLOCK TO INTERRUPT
4010  070022                                ENDSRV
4011  070022                                                                L10005:
4012  070022  000002                                                               RTI
4013
4014
4015                                  .SBTTL  PREG14  PRESERVE REGISTERS 1 THROUGH 4 ACROSS SUBROUTINE CALLS
4016                                  ;*************************************************************************
4017                                  ;INPUTS:        THE RELATIVE ADDRESS OF THE CALLED ROUTINE MUST FOLLOW THE
4018                                  ;               CALL TO THIS ROUTINE (SEE CALLING SEQUENCE).
4019                                  ;
4020                                  ;OUTPUTS:       -REGISTERS 1 THROUGH 4 ARE PRESERVED ACROSS THE CALLED ROUTINE.
4021                                  ;               -A CHECK IS MADE TO ENSURE THE HARDWARE STACK HASN'T OVER-RUN
4022                                  ;                THE PARAMETER STACK.
4023                                  ;
4024                                  ;CALLING SEQUENCE:      THIS IS BEST HANDLED BY THE "CALL" MACRO.  THE ACTUAL
4025                                  ;                       CALL IS:
4026                                  ;                                       JSR     R4,PREG14
4027                                  ;                                       .WORD   [SUBROUTINE NAME]-ANCHOR
4028                                  ;
4029                                  ;COMMENTS:      THIS ROUTINE IS RE-ENTRANT AND RELOCATABLE.
4030                                  ;               THIS ROUTINE IS DRS COMPATIBLE.
4031                                  ;
4032                                  ;SUBORDINATE ROUTINES CALLED:   THE ROUTINE SPECIFIED IN THE CALL.
4033                                  ;*************************************************************************
4034
4035                                                          ;R4 IS ALREADY ON THE R6 STACK.
4036  070024  010346                 PREG14: MOV     R3,-(SP)        ;PUSH R3, R2, R1
4037  070026  010246                         MOV     R2,-(SP)        ;
4038  070030  010146                         MOV     R1,-(SP)        ;
4039
4040  070032  010437  050564                 MOV     R4, PCCALI
4041  070036  012401                         MOV     (R4)+,R1        ;GET THE RELATIVE ADDRESS OF THE CALLED
4042                                                                  ;ROUTINE.
4043  070040  060701                         ADD     PC,R1           ;MAKE IT AN ABSOLUTE ADDRESS.
4044
4045  070042  010446                 ANCHOR: MOV     R4,-(SP)        ;SAVE THE RETURN TO THE CALLING ROUTINE.
4046
4047  070044  020506                         CMP     R5,SP           ;CHECK FOR STACK OVER-RUN.
4048  070046  103401                         BLO     1$              ;
4049  070050  000000                         HALT                    ;HANDLE STACK OVER-RUN CONDITION.
4050
4051  070052  004711                 1$:     JSR     PC,(R1)         ;CALL THE SPECIFIED ROUTINE.
4052
4053  070054  012604                         MOV     (SP)+,R4        ;RESTORE THE RETURN TO THE CALLING ROUTINE.
4054
4055  070056  012601                         MOV     (SP)+,R1        ;RESTORE THE REGISTERS.
4056  070060  012602                         MOV     (SP)+,R2        ;
4057  070062  012603                         MOV     (SP)+,R3        ;
4058  070064  000204                         RTS     R4              ;BACK TO THE CALLING ROUTINE.
4059
4060
```

```
                                .SBTTL  WAIT    WAIT FOR DEUNA INTERRUPT WITH TIMEOUT

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;               THIS SUBROUTINE WAITS FOR DNI INTERRUPTS FROM THE DEUNA
                                ;               OR REPORTS A TIMEOUT.   IF A UNA INTERRUPT HAS OCCURED,
                                ;               THE SUBROUTINE ERROR IS CALLED TO HANDLE IT.
                                ;
                                ;               SUCCESS OR FAILURE IS REPORTED VIA P1.
                                ;
                                ; INPUTS -
                                ;               NONE
                                ;
                                ; OUTPUTS-
                                ;               P1:     SUCCESS/FAILURE      0=SUCCESS/-1=FAILURE
                                ;
                                ; CALLING SEQUENCE:
                                ;               CALL    WAIT
                                ;               P$POP   P1
                                ;--
070066  012703  000012  WAIT::  MOV     #10.,R3                 ; MOVE NO. OF COUNTS TO R3
070072  012704  003714          MOV     #TIMER1,R4              ;  AND TIMER TO BE USED TO R4
070076  005002                  CLR     R2                      ;LOCAL STATUS PARAMETER
070100  010314                  MOV     R3,(R4)                 ;SET NUMBER OF TICKS. (GLOBAL)
070102  005737  050530  1$:     TST     ERRFLG                  ;CHECK IF ERROR OCCURED
070106  001011                  BNE     3$                      ; BR IF YES
070110  005737  050522          TST     DNIFLG                  ;CHECK FOR DNI INTERRUPT
070114  001403                  BEQ     2$                      ; BR IF INTERRUPT RECEIVED
070116  005037  050522          CLR     DNIFLG
070122  000410                  BR      6$
070124  005714          2$:     TST     (R4)                    ;HAS TIMER EXPIRED?
070126  001365                  BNE     1$                      ; BR IF NO TO WAIT FOR INTERRUPT
070130  000403                  BR      5$                      ;BR TO 5$
070132                  3$:     CALL    ERROR                   ;CALL ERROR ROUTINF
070140  012702  177777  5$:     MOV     #-1,R2                  ;INDICATE FAILURE
070144                  6$:     RETURN  R2                      ;RETURN WITH SUCCESS/FAILURE INDICATION

                                .SBTTL  ERROR   HANDLE UNA INTERRUPT ERRORS

                                ;--+
                                ; FUNCTIONAL DESCRIPTION:
                                ;               THIS SUBROUTINE CHECKS THE ERROR FLAGS SET BY
                                ;               UNAISR THE INTERRUPT SERVICE ROUTINE AND PRINTS
                                ;               OUT THE APPRORIATE ERROR MESSSAGES
                                ;
                                ; INPUTS -
                                ;   IMPLICIT:   ERROR FLAGS SHOULD BE SET BY UNAISR ROUTINES.
                                ; OUTPUTS -
                                ;   IMPLICIT:   ERROR MESSAGES ARE PRINTED OUT TO THE OPERATOR CONSOLE.
                                ;
                                ; CALLING SEQUENCE:
                                ;               CALL ERROR
                                ;
                                ;--+
```

```
                                ERROR:: DEC     ERRFLG                  ;DECRIMENT ERROR COUNTER TO SHOW
                                                                        ;THAT IT HAS BEEN HANDLED
                                        TST     PCEFLG                  ;SEE IF PORT COMMAND ERROR
                                        BNE     5$                      ; IF YES, BRANCH
                                        TST     FATFLG                  ;SEE IF UNA FATAL ERROR
                                        BNE     10$                     ; IF YES, BRANCH
                                        TST     BCOUNT                  ;SEE IF UNEXPLAINED INTERRUPT
                                        BNE     15$                     ; IF YES, BRANCH
                                        ERRDF   4,EMSG04,ERR1           ;ELSE UNKNOWN ERROR
                                                                                TRAP    C$ERDF
                                                                                .WORD   4
                                                                                .WORD   EMSG04
                                                                                .WORD   ERR1
                                        BR      20$                     ; EXIT
                                5$:     ERRDF   1,EMSG01,ERR1           ;PORT COMMAND ERROR
                                                                                TRAP    C$ERDF
                                                                                .WORD   1
                                                                                .WORD   EMSG01
                                                                                .WORD   ERR1
                                        DEC     PCEFLG                  ; INDICATE THAT IT WAS HANDLED
                                        BR      20$                     ; EXIT
                                10$:    ERRDF   2,EMSG02,ERR1           ;UNA FATAL ERROR
                                                                                TRAP    C$ERDF
                                                                                .WORD   2
                                                                                .WORD   EMSG02
                                                                                .WORD   ERR1
                                        DEC     FATFLG                  ; KEEP UP ON BOOK KEEPING
                                        BR      20$                     ; EXIT
                                15$:    ERRDF   3,EMSG03,ERR1           ;UNEXPLAINED INTERRUPT
                                                                                TRAP    C$ERDF
                                                                                .WORD   3
                                                                                .WORD   EMSG03
                                                                                .WORD   ERR1
                                        DEC     BCOUNT                  ; BOOK KEEPING
                                20$:    RETURN                          ;RETURN


                                .SBTTL  UNAINI  INITIALIZE THE UNA


                                ;***************************************************************;
                                ;                                                               ;
                                ; SUBROUTINE TO                                                 ;
                                ;               1) SETS UNA IN THE READY STATE                  ;
                                ;               2) INITIALIZES ALL UNA GLOBAL DATA LOCATIONS    ;
                                ;                       TO DEFAULT VALUES.                      ;
                                ;                                                               ;
                                ; CALLED BY:                                                    ;
                                ;               CALL UNAINI                                     ;
                                ;                                                               ;
                                ; INPUTS:      NONE                                            ;
                                ;                                                               ;
                                ; OUTPUTS:     NONE                                            ;
                                ;                                                               ;
                                ; SIDEEFFECTS: ALL GLOBAL LOCATIONS ARE ZEROED                  ;
```

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


                                                ;****************************************;
                                                ;        RESET AND STOP UNA HARDWARE     ;
                                                ;****************************************;

070262  013703  047632                  UNAINI::MOV     PCSR0, R3                       ; MOVE ADDRESS OF PCSR0 TO R3
070266  012713  000040                          MOV     #RSET, (R3)                     ; HARDWARE RESET UNA

070272  005002                                  CLR     R2                              ; LOOP COUNTER INIT
070274  011301                          7$:     MOV     (R3), R1                        ; READ PCSR0
070276  032701  004000                          BIT     #DNI, R1                        ; WAIT FOR COMMAND TO FINISH
070302  001006                                  BNE     9$                              ; BACK TIL DNI =1
070304  005302                                  DEC     R2                              ; COUNT DOWN DELAY
070306  001372                                  BNE     7$                              ; BACK UNTIL TIMEOUT
070310                                          ERRDF   15,EMSG15,ERR1                  ; PRINT '' DNI DID NOT SET FROM''
070310  104455                                                                                  TRAP    C$ERDF
070312  000017                                                                                  .WORD   15
070314  065031                                                                                  .WORD   EMSG15
070316  067460                                                                                  .WORD   ERR1
                                                                                        ;       '' A RESET''

070320  012713  004000                  9$:     MOV     #DNI, (R3)                      ; WRITE ONE TO CLEAR DNI

                                                ;****************************************;
                                                ; NOW ENABLE INTERRUPTS FOR THE ALL      ;
                                                ;        PORT COMMANDS ET AL             ;
                                                ;****************************************;

070324  112713  000100                          MOVB    #INTE, (R3)                     ; ENABLE INTERRUPTS

                                                ;****************************************;
                                                ; LOAD ADDRESS OF PCBB                   ;
                                                ;****************************************;

070330  012763  047666  000004                  MOV     #PCBB0,4(R3)                    ; LOWER 16 BITS OF ADRS
070336  005063  000006                          CLR     6(R3)                           ; UPPER 2

070342                                          CALL    COMAND  #GETPCB                         ; LOAD ADDRESS
070354                                          P$POP   R2                              ; GET SUCCESS/FAILURE REPORT
070356  001404                                  BEQ     10$                             ; CONTINUE IF OK
070360                                          ERRDF   5,EMSG05,ERR1                   ; ELSE REPORT ERROR
070360  104455                                                                                  TRAP    C$ERDF
070362  000005                                                                                  .WORD   5
070364  064415                                                                                  .WORD   EMSG05
070366  067460                                                                                  .WORD   ERR1

                                                ;****************************************;
070370                                  10$:    ;          INIT GLOBAL DATA              ;
                                                ;****************************************;
070370  005037  047642                          CLR     PCSR0C                          ; INIT CONTENTS OF CUR PCSR0
070374  005037  047644                          CLR     PCSR1C                          ;       PCSR1
070400  005037  047646                          CLR     PCSR2C                          ;       PCSR2
070404  005037  047650                          CLR     PCSR3C                          ;       PCSR3
```

```
4229
4230                                    ;****************** **********************;
4231                                    ;        INIT RING POINTERS AND          ;
4232                                    ;           RING ENTRIES                 ;
4233                                    ;****************************************;
4234
4235  070410  012737  003772  003756          MOV     #XRING,XRGCUR                   ; SET POINTERS TO BEGINING OF RING
4236  070416  012737  003772  003762          MOV     #XRING,XRGNXT
4237  070424  012737  004066  003760          MOV     #RRING,RRGCUR
4238  070432  012737  004066  003764          MOV     #RRING,RRGNXT
4239
4240  070440  012702  000006                  MOV     #NO.NTR,R2                      ; RESET OWNERSHIP AND STATUS OF
4241  070444  013701  003756          12$:    MOV     XRGCUR,R1                       ;  RING ENTRIES
4242  070450  012761  000000  000004          MOV     #0,4(R1)                        ; FOR TRANSMIT RING...
4243  070456                                  CALL    GETXNX  #XRGCUR
4244  070470  005302                          DEC     R2
4245  070472  001364                          BNE     12$
4246
4247  070474  012702  000006                  MOV     #NO.NTR,R2                      ; ...AND RECIEVE RING
4248  070500  013701  003760          13$:    MOV     RRGCUR,R1
4249  070504  012761  100000  000004          MOV     #100000,4(R1)
4250  070512                                  CALL    GETRNX  #RRGCUR
4251  070524  005302                          DEC     R2
4252  070526  001364                          BNE     13$
4253
4254
4255                                    ;****************************************;
4256                                    ;       CLEAR UP ALL PCB AND UCB,        ;
4257                                    ;    SET BACK  TO DEFAULT VALUES         ;
4258                                    ;****************************************;
4259
4260  070530  012701  047750                  MOV     #$RDDE, R1                      ; READ DEF PHY ADDR PCB
4261  070534  005061  000002                  CLR     2(R1)
4262  070540  005061  000004                  CLR     4(R1)
4263  070544  005061  000006                  CLR     6(R1)
4264
4265  070550  012701  047760                  MOV     #$RDPH, R1                      ; RE.D CURRENT PHY ADDR PCB
4266  070554  005061  000002                  CLR     2(R1)
4267  070560  005061  000004                  CLR     4(R1)
4268  070564  005061  000006                  CLR     6(R1)
4269
4270  070570  012701  047770                  MOV     #$WDPH, R1                      ; WRITE CURRENT PHY ADDR PCB
4271  070574  005061  000002                  CLR     2(R1)
4272  070600  005061  000004                  CLR     4(R1)
4273  070604  005061  000006                  CLR     6(R1)
4274
4275  070610  012701  050100                  MOV     #$RDRN, R1                      ; READ RING FORMAT
4276  070614  005061  000004                  CLR     4(R1)
4277  070620  005061  000006                  CLR     6(R1)
4278
4279  070624  012701  050110                  MOV     #UCB10, R1                      ; RING ENTRIES:
4280          000006                          .REPT 6
4281                                          CLR     (R1)+                           ; INIT TO 0
4282                                          .ENDR
4283
4284  070644  012701  050124                  MOV     #$WDRN, R1                      ; WRITE RING FORMAT
```

```
4285  070650  005061  000004                CLR     4(R1)
4286  070654  005061  000006                CLR     6(R1)
4287
4288  070660  012701  050150                MOV     #$RDCN, R1                      ; READ COUNTERS
4289  070664  005061  000004                CLR     4(R1)
4290
4291  070670  012701  050160                MOV     #UCB12, R1                      ; COUNTER ENTRIES
4292  070674  012702  000040                MOV     #40,R2
4293  070700  005021              14$:      CLR     (R1)+
4294  070702  005302                        DEC     R2
4295  070704  001375                        BNE     14$
4296
4297  070706  012701  050260                MOV     #$CLRC, R1                      ; CLEAR COUNTERS
4298  070712  005061  000004                CLR     4(R1)
4299
4300  070716  012701  050270                MOV     #$RDMO, R1                      ; READ MODE
4301  070722  005061  000002                CLR     2(R1)
4302  070726  005061  000004                CLR     4(R1)
4303  070732  005061  000006                CLR     6(R1)
4304
4305  070736  012701  050300                MOV     #$WDMO, R1                      ; WRITE MODE
4306  070742  005061  000002                CLR     2(R1)
4307  070746  005061  000004                CLR     4(R1)
4308  070752  005061  000006                CLR     6(R1)
4309
4310  070756  012701  050310                MOV     #$RDST, R1                      ; READ STATUS
4311  070762  005061  000002                CLR     2(R1)
4312  070766  005061  000004                CLR     4(R1)
4313  070772  005061  000006                CLR     6(R1)
4314
4315  070776  012701  050320                MOV     #$CLRS, R1                      ; READ AND CLEAR STATUS
4316  071002  005061  000002                CLR     2(R1)
4317  071006  005061  000004                CLR     4(R1)
4318  071012  005061  000006                CLR     6(R1)
4319
4320  071016  005037  050516                CLR     NIRCNT                          ; CLEAR BUFFERS RECEIVED COUNTER
4321  071022                                CALL    FUNCT #WDRNGS                   ; WRITE DESCRIPTOR RINGS
4322  071034                                P$POP   R2                              ; CHECK FOR ERROR
4323  071036  001404                        BEQ     15$                             ; IF OK, CONTINUE
4324  071040                                ERRDF   16,EMSG16,ERR1                  ; ELSE REPORT ERROR
4325  071040  104455                                                                        TRAP    C$ERDF
4326  071042  000020                                                                        .WORD   16
4327  071044  065064                                                                        .WORD   EMSG16
4328  071046  067460                                                                        .WORD   ERR1
4329
4330  071050              15$:      CALL    COMAND #STRT                    ; PUT UNA IN RUNNING STATE
4331  071062                        P$POP   R2                              ; CHECK FOR ERROR
4332  071064  001404                BEQ     20$                             ; IF OK, CONTINUE
4333  071066                        ERRDF   7,EMSG07,ERR1                   ; ELSE REPORT ERROR
4334  071066  104455                                                                TRAP    C$ERDF
4335  071070  000007                                                                .WORD   7
4336  071072  064510                                                                .WORD   EMSG07
4337  071074  067460                                                                .WORD   ERR1
4338
4339  071076              20$:      CALL    FUNCT #RDPHYA                   ; READ UNA PHYSICAL ADDRESS
4340  071110                        P$POP   R2                              ; CHECK FOR ERROR
```

```
4341  071112  001404                    BEQ     25$                             ; IF OK, CONTINUE
4342  071114                            ERRDF   6,EMSG06,ERR1                   ; ELSE, REPORT ERROR
4343  071114  104455                                                                    TRAP    C$ERDF
4344  071116  000006                                                                    .WORD   6
4345  071120  064450                                                                    .WORD   EMSG06
4346  071122  067460                                                                    .WORD   ERR1
4347
4348  071124  012701  047670    25$:    MOV     #PCBB2, R1                      ; STORE PHYSICAL ADDRESS
4349  071130  012704  047762            MOV     #PHYADR, R4
4350  071134  012124                    MOV     (R1)+,(R4)+                     ; MOVE FIRST TWO BYTES
4351  071136  012124                    MOV     (R1)+,(R4)+                     ;  AND SECOND TWO
4352  071140  011114                    MOV     (R1),(R4)                       ;  AND DONE
4353
4354  071142                            RETURN
4355
4356                            .SBTTL  UNAISR  UNA INTERRUPT SERVICE ROUTINE
4357                            ;+
4358                            ;       THIS INTERRUPT ROUTINE WILL BE THE ONLY INTERRUPT ROUTINE
4359                            ;       WHICH THE UNA HARDWARE WILL VECTOR TO UPON HARDWARE INTERRUPT.
4360                            ;
4361                            ;       THE REASON FOR SUCH INTERRUPTS ARE TO BE DETERMINED
4362                            ;       FROM THE APPROPRIATE BITS IN THE PCSR0.
4363                            ;
4364                            ;       IN ADDITION ALL WRITE ONE TO CLEAR BITS OF THE PCSR0
4365                            ;       ARE CLEARED FROM THIS ROUTINE.
4366                            ;-
4367
4368  071144                    UNAISR::
4369  071144  010146                    MOV     R1,-(SP)                ;SAVE R1
4370  071146  010246                    MOV     R2,-(SP)                ;...
4371  071150  010346                    MOV     R3,-(SP)                ;...
4372
4373  071152  005003                    CLR     R3                      ;SETUP WRITE 1 TO CLR MASK
4374  071154  013701  047632            MOV     PCSR0,R1                ;GET PCSR0 ADDRESS
4375
4376  071160  011103                    MOV     (R1),R3                 ;AND ITS CONTENTS
4377
4378  071162  012137  047642            MOV     (R1)+,PCSR0C            ;SAVE PCSR'S FOR DEBUG
4379  071166  012137  047644            MOV     (R1)+,PCSR1C
4380  071172  012137  047646            MOV     (R1)+,PCSR2C
4381  071176  011137  047650            MOV     (R1),PCSR3C
4382  071202  013701  047632            MOV     PCSR0,P1
4383
4384  071206  000303                    SWAB    R3                      ;REORIENT CONTENTS OF PCSR0
4385  071210  110361  000001            MOVB    R3,1(R1)                ;WRITE ONE TO CLEAR
4386                                                                    ; ONLY CLEAR UPPER BYTE
4387  071214  000303                    SWAB    R3                      ;REORIENT CONTENTS OF PCSR0
4388
4389
4390  071216  032703  100400            BIT     #SERI!FATI,R3           ;ANY FATAL STATUS ??
4391  071222  001403                    BEQ     10$
4392
4393  071224  005237  050512            INC     FATFLG                  ;SET FLAG
4394  071230  000443                    BR      90$                     ;EXIT
4395
4396  071232  032703  040000    10$:    BIT     #PCEI,R3                ;PORT COMMAND ERROR INTERRUPT?
```

```
4397  071236  001402                    BEQ     20$                     ;NO
4398
4399  071240  005237  050514            INC     PCEFLG                  ;YES, INCREMENT FLAG
4400
4401  071244  032703  020000  20$:      BIT     #RXI,R3                 ;RECV INTERRUPT ??
4402  071250  001402                    BEQ     30$                     ;NO
4403  071252  005237  050516            INC     NIRCNT                  ;YES, SET FLAG
4404
4405  071256  032703  010000  30$:      BIT     #TXI,R3                 ;TRANSMIT INTERRUPT ??
4406  071262  001402                    BEQ     40$                     ;NO
4407  071264  005037  050520            CLR     XFLAG                   ;YES, SET FLAG
4408
4409  071270  032703  004000  40$:      BIT     #DNI,R3                 ;COMMAND DONE ??
4410  071274  001402                    BEQ     45$                     ;NO
4411  071276  005237  050522            INC     DNIFLG                  ;YES, COUNT EACH DNI
4412
4413  071302  032703  002000  45$:      BIT     #RCBI,R3                ;RECIEVE BUFFER UNAVAILABLE?
4414  071306  001402                    BEQ     50$                     ;NO
4415  071310  005237  050524            INC     RBFCNT                  ; COUNT THEM
4416
4417  071314  032703  034000  50$:      BIT     #RXI!TXI!DNI,R3         ;CHECK FOR NON-ERROR INTERRUPT
4418  071320  001007                    BNE     90$                     ;EXIT IF ONE OCCURED
4419  071322  032703  142000            BIT     #SERI!PCEI!RCBI,R3      ;CHECK FOR ERROR INTERRUPT
4420  071326  001002                    BNE     80$                     ;IF ONE OCCURED, INCR. ERRFLG
4421  071330  005237  050526            INC     BCOUNT                  ;ELSE, NONSENSE INTERRUPT
4422  071334  005237  050530  80$:      INC     ERRFLG
4423  071340  012603            90$:    MOV     (SP)+,R3                ;RESTORE REGISTERS
4424  071342  012602                    MOV     (SP)+,R2                ;RESTORE REGISTERS
4425  071344  012601                    MOV     (SP)+,R1                ;RESTORE REGISTERS
4426  071346  000002                    RTI                             ;AUF WIEDERSEHEN
4427
4428
4429
4430                          .SBTTL  COMAND  SUBR TO ISSUE A UNA PORT COMMAND
4431
4432                          ;--+
4433                          ; FUNCTIONAL DESCRIPTION
4434                          ;               THIS SUBROUTINE ISSUES A UNA PORT COMMAND.  ERRORS
4435                          ;               ARE HANDLED BY THE SUBROUTINE ERROR AND REPORTED IN
4436                          ;               P2 IF ONE OCCURED.
4437                          ;
4438                          ; INPUTS -      P1 - THE UNA PORT COMMAND MNEMONIC OF THE
4439                          ;                    DESIRED COMMAND.
4440                          ; OUTPUTS -     P2 - SUCCESS REPORT.  CONTAINS 0 FOR SUCCESS
4441                          ;                    -1 IF A UNA ERROR OCCURED. THIS PARAMETER
4442                          ;                    IS PASSED DIRECTLY FROM THE WAIT
4443                          ;                    ROUTINE AND IS UNTOUCHED BY COMAND.
4444                          ;
4445                          ; CALLING PROCEDURE - CALL  COMAND  #<COMMAND TYPE>
4446                          ; SIDE EFFECTS - IF AN ERROR HAS OCCURED, THE ROUTINE ERROR WILL
4447                          ;                BE CALLED.
4448                          ; REGISTER USAGE - R1 CONTAINS THE COMMAND TYPE.
4449                          ;
4450                          ;--+
4451
4452  071350                  COMAND::
```

```
4453  071350                          P$POP   R1              ;MOVE COMMAND TYPE TO R1
4454  071352  052701  000100          BIS     #INTE,R1        ;ADD INTERRUPT TO COMMAND
4455  071356  010177  156250          MOV     R1,@PCSR0       ;MOV COMMAND TO PCSR0
4456  071362                          CALL    WAIT            ;WAIT FOR DONE INTERRUPT
4457  071370                  10$:    RETURN                  ;RETURN - ERROR INFO STILL ON
4458                                                          ; PARAMETER STACK FROM WAIT SUB.
4459
4460                          .SBTTL  FUNCT   SUBR TO  PERFORM A UNA PORT FUNCTION
4461
4462
4463                          ;--+
4464                          ; FUNCTIONAL DESCRIPTION:
4465                          ;               THIS SUBROUTINE PERFORMS A UNA PORT FUNCTION.  THE
4466                          ;               FUNCTION SPECIFIC PCB IS MOVED INTO THE UNA PCB.
4467                          ;
4468                          ; INPUTS -      P1 - THE UNA PORT FUNCTION MNEMONIC OF THE
4469                          ;                    DESIRED FUNCTION.
4470                          ; OUTPUTS -     P2 - SUCCESS REPORT.  CONTAINS 0 FOR SUCCESS
4471                          ;                    -1 IF A UNA ERROR OCCURED,
4472                          ;                    THIS PARAMETERIS PASSED DIRECTLY FROM THE
4473                          ;                    COMAND SUB AND IS NOT AFFECTED BY FUNCT.
4474                          ;
4475                          ; CALLING PROCEDURE - CALL  FUNCT #<FUNCTION TYPE>
4476                          ; SIDE EFFECTS - IF AN ERROR HAS OCCURED, THE ROUTINE ERROR WILL
4477                          ;                BE CALLED.
4478                          ; REGISTER USAGE - R1 CONTAINS THE FUNCTION TYPE, WHICH IS TRANSFORMED
4479                          ;                  TO THE ADDRESS OF THE FUNCTION SPECIFIC PCB.
4480                          ;                R2 CONTAINS THE ADDRESS OF THE UNA PCB.
4481                          ;
4482                          ;--+
4483
4484  071372          FUNCT:: P$POP    R1                     ; GET FUNCTION TYPE INTO R1
4485  071374  006301          ASL     R1                      ; MULTIPLY BY TWO
4486  071376  062701  047676  ADD     #FUNTAB,R1              ; ADD FUNCTION TABLE OFFSET
4487                                                          ; R1 NOW CONTAINS ADDRESS OF ADDRESS
4488                                                          ;  OF FUNCTION SPECIFIC PCB
4489  071402  012702  047666  MOV     #PCBB0, R2              ; PUT UNA PCB INTO R2
4490  071406  011101          MOV     (R1),R1                 ; PUT ADDRESS OF PCB INTO R1
4491  071410  012122          MOV     (R1)+,(R2)+             ; MOV PCB'S
4492  071412  012122          MOV     (R1)+,(R2)+             ; MOV PCB'S
4493  071414  012122          MOV     (R1)+,(R2)+             ; MOV PCB'S
4494  071416  012122          MOV     (R1)+,(R2)+             ; MOV PCB'S
4495  071420                  CALL COMAND #GETFNT             ; ISSUE GET PORT FUNCTION COMMAND
4496  071432                  RETURN                          ; SUCCESS INFO FROM COMAND SUBROUTINE
4497
4498                                                          ; IS STILL ON PARAMETER STACK
4499                          .SBTTL  XMIT    TRANSMIT UNA PACKETS
4500
4501                          ;--+
4502                          ; FUNCTIONAL DESCRIPTION:
4503                          ;               THIS SUBROUTINE SETS UP THE TRANSMIT DESCRIPTOR
4504                          ;               RING ENTRIES AND ISSUES A POLL DEMAND COMMAND TO
4505                          ;               THE UNA.
4506                          ;
4507                          ; INPUTS -      NONE
4508                          ;
```

```
                        ;
                        ; OUTPUTS -     P1 - SUCCESS REPORT.  0 FOR SUCCESS, -1 FOR FAILURE
                        ;
                        ; CALL PROCEDURE - CALL XMIT
                        ;                   P$POP P1
                        ;
                        ; SIDE EFFECTS - THE RING POINTER XRGNXT WILL BE UPDATED TO POINT THE NEXT
                        ;                AVAILABLE ENTRY AFTER THE TRANSMIT OPERATION.
                        ;
                        ; REGISTER USAGE - R1 POINTS TO TIMOUT TIMER LOCATION
                        ;                  R2 IS USED AS A POINTER IF RETRYS IS SET
                        ;                  R3 IS USED TO PASS THE SUCCESS/FAILURE MESSAGE BACK
                        ;                  R4 IS USED AS A POINTER TO RING ENTRIES OR STATUS INFO.
                        ;--+
071434  005037  050534  XMIT::  CLR     RETRYS
071440  013704  003756  1$:     MOV     XRGCUR,R4               ; MOVE RING ENTRY LOCATION INTO R4
071444  032764  100000  000004        BIT     #OWN,4(R4)              ; MAKE SURE WE OWN THIS
071452  001125                      BNE     40$                     ;  ELSE, BOOKKEEPING ERROR
071454  013714  050566              MOV     BUFLEN,(R4)             ; MOVE BUFFER LENGTH INTO FIRST WORD OF
                                                                    ; NEXT AVAILABLE RING ENTRY
071460  052764  101400  000004        BIS     #OWN!STP!ENP,4(R4)      ; SET OWNERSHIP, START AND END OF PACKET BITS
071466  012737  000001  050520  20$:   MOV     #1,XFLAG                ; SET TRANSMIT FLAG
071474                              CALL    COMAND #PDMD            ; ISSUE PDMD COMMAND
071506                              P$POP   R3                      ; CHECK FOR ERRORS
071510  001126                      BNE     50$                     ; IF YES, EXIT
071512  012701  003716  22$:   MOV     #TIMER2,R1              ; SET UP TO WAIT FOR TRANSMIT TO COMPLETE
071516  012711  000100              MOV     #100.,(R1)
071522  005737  050520  23$:   TST     XFLAG                   ; SEE IF TRANSMIT DONE BIT SET
071526  001403                      BEQ     24$                     ;  IF SET, SKIP WAIT LOOP
071530  005711                      TST     (R1)                    ;  ELSE, SEE IF TIMEOUT YET
071532  001373                      BNE     23$                     ;   NO, WAIT
071534  000506                      BR      45$                     ;   YES, EXIT
071536  032764  100000  000004  24$:   BIT     #OWN,4(R4)              ; SEE WHO OWNS THIS ENTRY
071544  001070                      BNE     40$                     ; IF UNA STILL OWNS THIS, SOMETHINGS WRONG
071546  032764  040000  000004        BIT     #ERRS,4(R4)             ; SEE IF ANY ERRORS
071554  001013                      BNE     30$                     ;  IF YES, BRANCH AND TAKE CARE OF THEM
071556                          26$:   CALL    GETXNX  #XRGCUR         ; UPDATE "TRANSMIT RING CURRENT" POINTER
071570  005003                      CLR     R3                      ;  INDICATE SUCCESS
071572  023737  003756  003762        CMP     XRGCUR,XRGNXT           ; SEE IF CURRENT POINTER = NEXT POINTER
071600  001057                      BNE     42$                     ;  IF NO, ERROR
071602  000473                      BR      55$                     ; RETURN
071604  032764  016000  000004  30$:   BIT     #DEF!ONE!MORE,4(R4)     ; WAS MESSAGE STILL SENT?
071612  001361                      BNE     26$                     ;  IF YES, GO TO NEXT ONE
071614  032764  002000  000006        BIT     #RTRY,6(R4)             ; ELSE, DID UNA GIVE UP AFTER 16 TRIES
071622  001434                      BEQ     32$                     ;  IF NOT, FATAL DEVICE ERROR, EXIT
071624  005237  050534              INC     RETRYS                  ;  IF YES, KEEP COUNT OF THEM
071630  022737  000003  050534        CMP     #3,RETRYS               ;   GIVE UP AFTER 3 TIMES
071636  100747                      BMI     26$                     ;  ELSE, SET UP TO RETRANSMIT
071640                              CALL    GETXNX  #XRGCUR         ;  UPDATE POINTERS
071652                              CALL    GETXNX  #XRGNXT         ;
071664  016402  000002              MOV     2(R4),R2                ;  SET UP TO COPY DATA BUFFER
071670  013704  003756              MOV     XRGCUR,R4               ;  R2 POINTS TO OLD BUFFER
071674  016403  000002              MOV     2(R4),R3                ;  R3 POINTS TO NEW BUFFER
071700  013704  050566              MOV     BUFLEN,R4               ;  R4 COUNTS NUMBER OF BYTES TO COPY
071704  112223                  31$:   MOVB    (R2)+,(R3)+             ;  COPY DATA
071706  005304                      DEC     R4
```

```
4565  071710  001375                  BNE     31$                     ;  HAVE WE COPIED ALL OF IT
4566  071712  000652                  BR      1$                      ;  IF YES, TRY AGAIN
4567  071714                  32$:    ERRDF   9,EMSG09,ERR1           ; ELSE, FATAL DEVICE ERROR
4568  071714  104455                                                          TRAP    C$ERDF
4569  071716  000011                                                          .WORD   9
4570  071720  064613                                                          .WORD   EMSG09
4571  071722  067460                                                          .WORD   ERR1
4572  071724  000420                  BR      50$                     ; EXIT
4573  071726                  40$:    ERRDF   10,EMSG10,ERR1          ; TRANSMIT RING BOOKKEEPING ERROR
4574  071726  104455                                                          TRAP    C$ERDF
4575  071730  000012                                                          .WORD   10
4576  071732  064656                                                          .WORD   EMSG10
4577  071734  067460                                                          .WORD   ERR1
4578  071736  000413                  BR      50$
4579  071740                  42$:    ERRDF   12,EMSG10,ERR1          ; BOOKKEEPING ERROR, XRGNXT SHOULD = XRGCUR
4580  071740  104455                                                          TRAP    C$ERDF
4581  071742  000014                                                          .WORD   12
4582  071744  064656                                                          .WORD   EMSG10
4583  071746  067460                                                          .WORD   ERR1
4584  071750  000406                  BR      50$
4585  071752  005237  050532    45$:  INC     TIMOUT
4586  071756                        ERRHRD  8,EMSG08,ERR1            ; REPORT ERROR
4587  071756  104456                                                          TRAP    C$ERHRD
4588  071760  000010                                                          .WORD   8
4589  071762  064553                                                          .WORD   EMSG08
4590  071764  067460                                                          .WORD   ERR1
4591  071766  012703  177777    50$:  MOV     #-1,R3                  ; ERROR INDICATOR
4592  071772                    55$:  RETURN  R3                      ; RETURN
4593
4594                            .SBTTL  RECEVE  RECEIVE UNA RING BUFFERS
4595
4596                            ;--+
4597                            ; FUNCTIONAL DESCRIPTION
4598                            ;               THIS SUBROUTINE TAKES INCOMING DATA BUFFERS FROM
4599                            ;               THE UNA AND CHECKS FOR ERRORS.  THIS PROCESS CONTINUES
4600                            ;               FOR ALL PENDING BUFFERS.
4601                            ;
4602                            ; INPUTS -      NONE
4603                            ; OUTPUTS -     P1 - THE NUMBER OF PACKETS HANDLED BY THIS CALL TO RECEVE
4604                            ;
4605                            ; CALLING PROCEDURE - CALL RECEVE
4606                            ;                     P$POP  P1
4607                            ;
4608                            ; SIDE EFFECTS - THE POINTERS RRGCUR AND RRGNXT ARE UPDATED.
4609                            ;
4610                            ; REGISTER USAGE - R1 IS USED TO HOLD CURRENT PACKET STATUS INFORMATION
4611                            ;                  R2 COUNTS THE NUMBER OF PACKETS HANDLED
4612                            ;                  R4 POINTS TO THE RING DESCRIPTOR ENTRY
4613                            ;
4614                            ;--+
4615
4616  071776                    RECEVE::
4617  071776  005002                  CLR     R2                      ; CLEAR PACKETS HANDLED COUNTER
4618  072000  005737  050516    1$:   TST     NIRCNT                  ; SEE IF ANY PACKETS TO RECEIVE
4619  072004  001476                  BEQ     60$                     ;  IF NO, EXIT
4620  072006  013704  003760          MOV     RRGCUR,R4               ; MOVE CURRENT RECEIVE RING POINTER TO R4
```

```
4621  072012  016401  000004            MOV     4(R4),R1                ; MOVE STATUS OF PACKET TO R1
4622  072016  012764  100000  000004    MOV     #100000,4(R4)           ; RESET RING DESCRIPTOR
4623  072024  032701  100000            BIT     #OWN,R1                 ; SEE WHO OWNS THIS BUFFER
4624  072030  001060                    BNE     45$                     ;  IF STILL OWNED BY UNA, ERROR
4625  072032  016403  000002            MOV     2(R4),R3                ; MOVE BUFFER ADDRESS INTO R3
4626  072036  016303  000014            MOV     PROTOT(R3),R3           ; MOVE PROTOTYPE INTO R3
4627  072042  020337  050544            CMP     R3,PROT00               ; SEE IF IT IS AN ACCEPTABLE PROTOCALL TYPE
4628  072046  001420                    BEQ     10$                     ;  IF YES, CONT.
4629  072050  020337  050546            CMP     R3,PROT02               ;  ELSE CHECK OTHER GOOD TYPE
4630  072054  001415                    BEQ     10$                     ;   IF OK, CONT.
4631  072056                    5$:     CALL    GETRNX  #RRGCUR         ;   ELSE, THROW OUT AND UPDATE
4632  072070                            CALL    GETRNX  #RRGNXT         ;    CURRENT AND NEXT POINTERS
4633  072102  005337  050516            DEC     NIRCNT                  ;  DECREMENT RECEIVE COUNTER
4634  072106  000435                    BR      60$                     ;  EXIT
4635  072110  032701  040000    10$:    BIT     #ERRS,R1                ; SEE IF ANY ERRORS
4636  072114  001413                    BEQ     20$                     ;  FOR NO ERRORS BR TO 20$
4637  072116  005237  050536            INC     RCVERR                  ;  ELSE, INCREMENT RECEIVE ERROR COUNTER
4638  072122                            PRINTF  #RECERR                 ;   PRINT ERROR MESSAGE
4639  072122  012746  052631                                                    MOV     #RECERR,-(SP)
4640  072126  012746  000001                                                    MOV     #1,-(SP)
4641  072132  010600                                                            MOV     SP,R0
4642  072134  104417                                                            TRAP    C$PNTF
4643  072136  062706  000004                                                    ADD     #4,SP
4644  072142  000745                    BR      5$                      ; UPDATE POINTERS AND RETURN
4645  072144  005237  050540    20$:    INC     RCVBUF                  ; INCREMENT GOOD BUFFERS RECEIVED COUNTER
4646  072150  005202            25$:    INC     R2                      ; KEEP COUNT OF HOW MANY BUFFERS RECEIVED
4647  072152  005337  050516            DEC     NIRCNT                  ; KEEP BOOKKEEPING IN ORDER
4648  072156                            CALL    GETRNX  #RRGCUR         ; UPDATE "RECEIVE RING CURRENT" POINTER
4649  072170  000404                    BR      60$
4650  072172                    45$:    ERRDF   11,EMSG11,ERR1          ; RECEIVE RING BOOKKEEPING ERROR
4651  072172  104455                                                            TRAP    C$ERDF
4652  072174  000013                                                            .WORD   11
4653  072176  064716                                                            .WORD   EMSG11
4654  072200  067460                                                            .WORD   ERR1
4655  072202                    60$:    RETURN  R2                      ; RETURN WITH NUMBER OF ENTRYS HANDLED
4656
4657
4658                                    ;--+
4659                                    ;  EDPACK                                          ETHERNET DATA PACK
4660                                    ;
4661                                    ;  THIS ROUTINE WILL CONVERT A STRING OF HEX CHARACTERS INTO A RIGHT
4662                                    ;  JUSTIFIED BINARY  STREAM  (WITH LEADING ZEROS),  COMPATIBLE  WITH
4663                                    ;  ETHERNET CONVENTIONS.   THE SOURCE STRING MUST BE FORMATTED USING
4664                                    ;  EITHER  A  WORD  BY  WORD  HEX  DESCRIPTION OR A BYTE BY BYTE HEX
4665                                    ;  DESCRIPTION.  THE  RETURNED  STRING  WILL  BE  BYTE  ORIENTED  AS
4666                                    ;  REQUIRED BY THE ETHERNET:
4667                                    ;  LO-BYTE-WORD0 HI-BYTE-WORD0 LO-BYTE-WORD1 HI-BYTE-WORD1, ETC.
4668                                    ;
4669                                    ;  INPUT ARGUMENTS          P1 - ADDRESS OF THE SOURCE  (HEX)  STRING
4670                                    ;                                TO BE CONVERTED TO A BINARY  STREAM.
4671                                    ;                            P2 - ADDRESS OF THE  DESIRED  DESTINATION
4672                                    ;                                BUFFER WHICH WILL ACCEPT BINARY DATA
4673                                    ;                            P3 - LENGTH (IN BYTES) OF THE DESTINATION
4674                                    ;                                BUFFER
4675                                    ;  EXPLICIT OUTPUTS         P4 - ZERO IF SUCCESSFUL, -1 IF BUFFER TOO
4676                                    ;                                LONG OR ODD NUMBER OF HEX CHARACTERS
```

```
                                ;       IMPLIED OUTPUTS         THE  BUFFER  AT  P2  WILL CONTAIN A RIGHT
                                ;                               JUSTIFIED  BINARY STREAM W/ LEADING ZEROS
                                ;                               AND  CORRESPONDING  TO  HEX STRING AT R5.
                                ;       SUBORDINATE PROCEDURES  HXFORM. (STRIP NONHEX CHARACTERS)
                                ;                               HEXBIN. (HEX TO BINARY CONVERSION)
                                ;       CALLING PROCEDURE       CALL EDPACK P1,P2,P3    ;INPUT P1-P3 PARAMETERS
                                ;                               P$POP   P4              ;OUTPUT    P4 PARAMETER
                                ;---
                                LOCDST: .BLKB   74.             ;MAX NUMBER OF CHARACTERS THAT MAY BE ENTERRED
                                SOURCE: .WORD                   ;SOURCE ADDRESS

                                EDPACK: P$POP   SOURCE,R4,R3            ;R4-DESTINATION, R3-NUMBER OF CHARS RQD
                                                                        ;SOURCE-SRC ADDRESS,  ORIENT-WORD/BYTE?
                                        CLR     R2                      ;ASSUME NO ERRORS, VALUE RETURNED
                                        ASL     R3                      ;NUMBER OF CHARACTERS REQUIRED W/ '0'S
                                        CALL    HXFORM SOURCE,#LOCDST,R3
                                        P$POP   R1,R2                   ;R1=ADDRESS OF LAST CHAR
                                                                        ;R2=SUCCESS/FAIL CODE (0/-1)
                                        TST     R2                      ;R1 WILL POINT TO RIGHTMOST CHARACTER
                                        BNE     9$                      ;RIGHT JUSTIFY BUFFER
                                                                        ;CONVERT HEX AT LOCDST TO BINARY
                                        ASR     R3                      ;R3 BYTES IN OUTPUT BIT STREAM
                                        CALL    HEXBIN #LOCDST,R4,R3

                                9$:     RETURN R2                       ;RETURN WITH SUCCESS/FAILURE INDICATION

                                ;--+
                                ;       HXFORM                                          HEX FORMAT ROUTINE
                                ;
                                ;       THIS ROUTINE WILL ACCEPT A HEX STRING, AND STRIP OUT  THE  NON-HEX
                                ;       CHARACTERS (WITHOUT FLAGGING AN ERROR FOR THE NON-HEX CHARACTERS).
                                ;       A NULL WILL BE LEFT AS THE END OF STRING DELIMITTER.
                                ;
                                ;       INPUT ARGUMENTS     P1- ADDRESS OF THE SOURCE  ASCII  STRING  (NULL
                                ;                               DELIMITTER AT END OF STRING)
                                ;                           P2- ADDRESS OF THE DESTINATION ASCII HEX STRING
                                ;                               (STRIPPED OF NON-HEX AND RIGHT  JUSTIFIED)
                                ;                           P3- THE NUMBER OF HEX  CHARACTERS  DESIRED  @R4
                                ;       EXPLICIT OUPUTS     P4- WILL  CONTAIN  THE  ADDRESS  OF  THE  LAST
                                ;                               CHARACTER IN THE OUTPUT BUFFER.
                                ;                           P5- THE SUCCESS/FAILURE (0/-1) INDICATOR
                                ;
                                ;       IMPLICIT OUTPUTS        THE BUFFER AT R4  WILL  CONTAIN  FORMATTED
                                ;                               HEX (ASCII) STRING.
                                ;       CALLING PROCEDURE       CALL HXFORM P1,P2,P3
                                ;                               P$POP P4,P5
                                ;---
                                HXN:    .WORD
                                HXFORM: P$POP   R3,R2,HXN       ;ADDRESS OF SOURCE STRING
                                                                ;ADDRESS OF DESTINATION STRING
                                                                ;NUMBER OF HEX CHARACTERS DESIRED

                                        MOV     R2,R4           ;DESTINATION ADDRESS, R2: DESTINATION POINTER
                                        ADD     HXN,R4          ;POINT TO END OF OUTPUT BUFFER (DESTINATION)
                                                                ;DO WHILE NO NULL FOUND IN SOURCE STRING
```

```
 4733  072432  112301           1$:     MOVB    (R3)+,R1        ;GET BYTE IN QUESTION (& POINT TO NEXT BYTE)
 4734  072434  142701  000200   2$:     BICB    #200,R1         ;ENSURE HI BIT IS LO (SEVEN BIT ASCII)
 4735                                                           ;CHECK FOR VALID HEX CHARACTER
 4736  072440  001414                    BEQ     4$              ;THE NULL IS A VALID CHARACTER (BUT THE LAST)
 4737  072442  120127  000060           CMPB    R1,#60          ;IF GREATER THAN "0" THEN RANGE OK
 4738  072446  100413                    BMI     6$              ;6$-OUT OF RANGE, BYTE VALUE TOO SMALL
 4739  072450  120127  000072           CMPB    R1,#72          ;IF BYTE IS LESS THAN 72 AND >=60 THEN RANGE OK
 4740  072454  100406                    BMI     4$              ;RANGE OK IF >=60 AND <72 ELSE, CONTINUE CHECK
 4741  072456  120127  000101           CMPB    R1,#101         ;BYTE MUST BE >=101 TO CONTINUE CHECK
 4742  072462  100405                    BMI     6$              ;IF BYTE >71 AND <101 THEN BYTE OUT OF RANGE.
 4743  072464  122701  000106           CMPB    #106,R1         ;BYTE MUST BE <=106 TO BE OK, ELSE: NOT OK.
 4744  072470  100402                    BMI     6$              ;BYTE NOT OK, DON'T PLACE THIS BYTE IN OUTPUT.
 4745                                                           ;PLACE THE HEX BYTE IN THE OUTPUT BUFFER.
 4746  072472  110122           4$:     MOVB    R1,(R2)+        ;BYTE IN RANGE. CONFIRMED. BYTE OK. POINT TO
 4747  072474  001405                    BEQ     9$              ;NEXT BYTE DEST ADDRESS. IF NULL, THEN EXIT.
 4748                                                           ;(NO ERRORS)
 4749                                                           ;IF NOT NULL, THEN CHECK FOR BUFFER OVERFLOW.
 4750                                                           ;R4 POINTS TO LAST CHARACTER POSITION (OUTPUT)
 4751  072476  020402           6$:     CMP     R4,R2           ;R2 - PRESENT WRITE ADDRESS
 4752  072500  100354                    BPL     1$              ;(SHOULD BE POSITIVE RESULT OR 0) (MORE TO DO)
 4753  072502  012704  177777   7$:     MOV     #-1,R4          ;SET ERROR CONDITION (EXIT WITH ERROR)
 4754  072506  000404                    BR      11$             ;ERROR DETECTED EXIT PATH -> (TOO MANY CHARS)
 4755
 4756                                                           ;SUCCESSFUL EXIT PATH
 4757  072510  005302           9$:     DEC     R2              ;POINT TO THE LAST ACTUAL CHARACTER AT DEST BFR
 4758  072512  020402                    CMP     R4,R2           ;CHECK FOR MINIMUN OF 12 CHARACTERS.
 4759  072514  001372                    BNE     7$              ;BRANCH IF LESS THAN 12, ERROR.
 4760  072516  005004                    CLR     R4              ;INDICATE SUCESS
 4761  072520                   11$:     RETURN  R2,R4           ;ADDRESS OF LAST CHARACTER (R2) IS P4
 4762                                                           ;ERROR INDICATOR (R4) IS P5
 4763
 4764                           ;--+
 4765                           ;  |     HEXBIN                          HEX TO BINARY CONVERSION PROCEDURE
 4766                           ;  |
 4767                           ;  |     THIS PROCEDURE WILL CONVERT A STRING  OF  HEX  (ASCII)  CHARACTERS
 4768                           ;  |     DIRECTLY TO A BINARY STREAM.  THE DESTINATION BINARY  STREAM  WILL
 4769                           ;  |     REQUIRE ONLY HALF AS MANY BYTES AS  THE  HEX  STRING  BECAUSE  TWO
 4770                           ;  |     HEX CHARACTERS ARE REQUIRED TO REPRESENT A SINGLE BINARY BYTE.
 4771                           ;  |
 4772                           ;  |     INPUTS           P1 - SOURCE STRING ADDRESS  (DELIMITTED BY A NULL)
 4773                           ;  |                       P2 - DESTINATION  ADDRESS  FOR  THE  BINARY  DATA.
 4774                           ;  |                       P3 - THE NUMBER OF BINARY BYTES REQUIRED (HALF THE
 4775                           ;  |                            NUMBER OF CHARACTERS AT P1.
 4776                           ;  |     OUTPUTS                NO EXPLICIT OUTPUTS
 4777                           ;  |     IMPLIED OUTPUTS        THE BUFFER AT P2 WILL CONTAIN  THE  BINARY
 4778                           ;  |                            STREAM, CONVERTED DIRECTLY FROM THE BUFFER
 4779                           ;  |                            AT P1.
 4780                           ;  |     SUBORDINATE PROCEDURES NONE
 4781                           ;  |     CALLING PROCEDURE      CALL    HEXBIN P1,P2,P3
 4782                           ;---
 4783  072526  000000           HN:     .WORD
 4784  072530  030460  031462  032464  CMPSTR: .ASCIZ  /0123456789ABCDEF/
 4785  072536  033466  034470  041101
 4786  072544  042103  043105     000
 4787          072552                   .EVEN
 4788
```

```
 4789  072552                          HEXBIN: P$POP   R1,R2,HN        ;R1=SOURCE STRING ADDRESS
 4790                                                                  ;R2=DESTINATION STRING ADDRESS
 4791                                                                  ;HN=NUMBER OF BYTES REQUIRED
 4792
 4793  072562  060237  072526                  ADD     R2 HN          ;HN NOW POINTS TO THE LAST_BYTE_POSITION+1
 4794
 4795  072566  012704  072530          1$:     MOV     #CMPSTR,R4     ;POINTER IN THE COMPARE STRING
 4796  072572  121124                  2$:     CMPB    (R1),(R4)+     ;COMPARE CURRENT CHAR WITH A CHAR IN CMPSTR
 4797  072574  001376                          BNE     2$             ;REPEAT UNTIL CHARACTER FOUND IN LIST
 4798  072576  005201                          INC     R1             ;POINT TO THE NEXT ASCII BYTE
 4799  072600  162704  072531                  SUB     #CMPSTR+1,R4   ;R4 NOW CONTAINS THE ACTUAL BINARY VALUE FOR
 4800                                                                  ;THE NIBBLE DESCRIBED BY THE CURRENT BYTE.
 4801                                                                  ;NOTE: NIBBLE IS THE HI PORTION OF THE BYTE
 4802  072604  006304                          ASL     R4             ;MOVE NIBBLE TO THE HI END OF THE BYTE
 4803  072606  006304                          ASL     R4
 4804  072610  006304                          ASL     R4
 4805  072612  006304                          ASL     R4
 4806  072614  010403                          MOV     R4,R3          ;SAVE THE HI NIBBLE
 4807
 4808  072616  012704  072530                  MOV     #CMPSTR,R4     ;POINTER INTO COMPARE STRING
 4809  072622  121124                  3$:     CMPB    (R1),(R4)+     ;COMPARE CURRENT CHAR WITH A CHAR IN CMPSTR
 4810  072624  001376                          BNE     3$             ;REPEAT UNTIL MATCH FOUND IN CMPSTR LIST
 4811  072626  005201                          INC     R1             ;POINT TO THE NEXT ASCII BYTE
 4812  072630  162704  072531                  SUB     #CMPSTR+1,R4   ;R4 NOW CONTAINS THE ACTUAL BINARY VALUE FOR
 4813                                                                  ;THE NIBBLE DESCRIBED BY THE CURRENT BYTE.
 4814                                                                  ;NOTE: NIBBLE IS THE HI PORTION OF THE BYTE
 4815  072634  050403                          BIS     R4,R3          ;NOW THE TWO CHARACTERS HAVE MADE A SINGLE BYTE
 4816                                                                  ;NOW PLACE THE COMPLETE BYTE IN THE DESTINATION
 4817  072636  110322                          MOVB    R3,(R2)+       ;AND POINT TO THE NEXT DESTINATION BYTE
 4818  072640  020237  072526                  CMP     R2,HN          ;IF THE DESTINATION POINTER [R2] REACHES THE
 4819  072644  100750                          BMI     1$             ;LAST CHARACTER POSITION+1 [HN] THEN DONE.
 4820  072646                                  RETURN                 ;RETURN TO CALLER
 4821
 4822
 4823
 4824
 4825                                  ;---+
 4826                                  ;   BINHEX                          BINARY TO HEX CONVERSION PROCEDURE
 4827                                  ;
 4828                                  ;   THIS PROCEDURE WILL CONVERT A BINARY DATA STREAM INTO A HEX STRING.
 4829                                  ;
 4830                                  ;   INPUTS  -  P1- BINARY DATA BUFFER ADDRESS
 4831                                  ;              P2- NUMBER OF BYTES IN THE BUFFER
 4832                                  ;              P3- ADDRESS OF OUTPUT BUFFER FOR HEX STRING.
 4833                                  ;                   HEX CHARACTER PAIRS SEPERATED BY "-"S
 4834                                  ;                  (NOTE: THIS BUFFER MUST BE AT LEAST 3*P2 BYTES LONG)
 4835                                  ;   OUTPUTS -  NONE
 4836                                  ;   IMPLICIT OUTPUTS          THE BUFFER AT P3 WILL CONTAIN THE HEX STRING
 4837                                  ;                             FOLLOWED BY A NULL CHARACTER.
 4838                                  ;   SUBORDINATE ROUTINES      NONE
 4839                                  ;   CALLING PROCEDURE         CALL BINHEX P1,P2,P3
 4840                                  ;---
 4841  072650  030460  031462  032464  HEXC:   .ASCII  /0123456789ABCDEF/
 4842  072656  033466  034470  041101
 4843  072664  042103  043105
 4844  072670  000000                  LST:    .WORD
```

```
4845
4846  072672                          BINHEX: P$POP   R1,LST,R2        ;R1 HAS THE INPUT BUFFER ADDRESS
4847                                                                  ;LST: HAS THE NUMBER OF BYTES IN INPUT BUFFER
4848                                                                  ;R2 HAS THE OUTPUT BUFFER ADDRESS
4849  072702  060137  072670                  ADD     R1,LST          ;LST IS NOW ADDRESS OF LAST SOURCE BYTE + 1
4850  072706  112103                  1$:     MOVB    (R1)+,R3        ;GET THE CURRENT BYTE AND POINT TO NEXT BYTE
4851  072710  110304                          MOVB    R3,R4           ;SEPARATE NIBBLES AND GET CHARACTERS SEPARATELY
4852  072712  042703  177760                BIC     #177760,R3      ;ONLY RIGHT BINARY NIBBLE REMAINS IN R3
4853  072716  006204                          ASR     R4              ;SHIFT OVER FOR LEFT BINARY NIBBLE IN R4
4854  072720  006204                          ASR     R4
4855  072722  006204                          ASR     R4
4856  072724  006204                          ASR     R4
4857  072726  042704  177760                BIC     #177760,R4      ;ONLY LEFT  BINARY NIBBLE REMAINS IN R4
4858                                                                  ;R4 IS THE MOST  SIGNIFICANT NIBBLE (FIRST)
4859                                                                  ;R3 IS THE LEAST SIGNIFICANT NIBBLE (SECOND)
4860  072732  116422  072650                MOVB    HEXC(R4),(R2)+  ;PUT THE ASCII BYTE INTO THE BUFFER HI POSITION
4861  072736  116322  072650                MOVB    HEXC(R3),(R2)+  ;PUT THE ASCII BYTE INTO THE BUFFER LO POSITION
4862  072742  112722  000055                MOVB    #'-,(R2)+       ;PUT - BETWEEN HEX PAIRS
4863  072746  020137  072670                CMP     R1,LST          ;RESULT IS NEGATIVE UNTIL R1=LST
4864  072752  103755                        BLO     1$              ;UNTIL R1=LST. (TRANSFER ALL SOURCE BYTES)
4865  072754  105042                        CLRB    -(R2)           ;TERMINATE OUTPUT BUFFER WITH A NULL
4866  072756                                RETURN
4867
4868
4869                                  .SBTTL  BLDLD   BUILD LOOP DIRECT DATA BUFFERS FOR TRANSMIT.
4870
4871                                  ;--+
4872                                  ; FUNCTIONAL DESCRIPTION:
4873                                  ;               THIS SUBROUTINE BUILDS LOOP DIRECT PACKETS FOR TRANSMISSION
4874                                  ;               FROM THE UNA. SOURCE ADDRESS, DESTINATION ADDRESS,
4875                                  ;               PROT. TYPE, AND LOOP DIRECT HEADER INFO ARE ADDED
4876                                  ;               TO THE MESSAGE BUFFER.  THE MESSAGE BUFFER IS BUILT
4877                                  ;               BY A CALL TO BLDBUF.
4878                                  ;
4879                                  ; INPUTS -      P1 - THE ADDRESS OF THE DESTINATION ADDRESS (FROM NODE TABLE)
4880                                  ;               IMPLICIT - P$SIZE CONTAINS THE SIZE OF THE MESSAGE BUFFER DATA
4881                                  ;                          XRGNXT POINTS TO THE NEXT AVAILABLE RING ENTRY
4882                                  ;                          PHYADR HOLDS THE CURRENT LOCAL UNA PHYSICAL ADDRESS
4883                                  ;
4884                                  ; OUTPUTS -     BUILT MESSAGE PACKET.
4885                                  ;
4886                                  ; CALLING PROCEDURE - CALL BLDLD P1
4887                                  ;
4888                                  ; SIDE EFFECTS - THE MESSAGE PACKET IS BUILD AND CONTAINED IN THE
4889                                  ;                 BUFFER POINTED TO BY XRGNXT WHEN THE ROUTINE WAS ENTERED.
4890                                  ;                XRGNXT IS UPDATED TO POINT TO THE NEXT RING ENTRY
4891                                  ;
4892                                  ; REGISTER USAGE - R1 HOLDS ADDRESS OF DESTINATION ADDRESS
4893                                  ;                  R2 IS A POINTER FOR THE LOOP DIRECT HEADER INFO
4894                                  ;                  R3 HOLDS THE PACKET LENGTH
4895                                  ;                  R4 HOLDS ADDRESS OF NEXT RING ENTRY DATA BUFFER
4896                                  ;
4897                                  ;--+
4898
4899  072760                          BLDLD::
4900  072760                                  P$POP   R1                      ; PUT ADDRESS OF DEST. ADDRESS IN R1
```

```
 4901  072762  013704  003762                MOV     XRGNXT,R4               ; MOVE NEXT PACKET ADDRESS TO R4
 4902  072766  032764  100000  000004        BIT     #OWN,4(R4)              ; CHECK OWNERSHIP BIT
 4903  072774  001110                        BNE     40$                     ; IF DON'T OWN, BOOKKEEPING ERROR,
 4904  072776  016404  000002                MOV     2(R4),R4                ; POINT R4 TO DATA BLOCK
 4905  073002  013703  002372                MOV     P$SIZE,R3               ; PUT MESSAGE SIZE INTO R3
 4906  073006  062703  000040                ADD     #40,R3                  ; ADD HEADER AND LOOP DIRECT INFO TO LENGTH
 4907  073012  010337  050560                MOV     R3,XFER                 ; PUT BYTES TRANSFERED INTO WORD
 4908  073016  162737  000016  050560        SUB     #16,XFER                ;  AND CORRECT FOR HEADER
 4909  073024  022703  002756                CMP     #XPKLEN,R3              ; SEE IF LONGER THAN ONE PACKET
 4910  073030  002477                        BLT     45$                     ;  IF YES, ERROR
 4911  073032  010337  050566                MOV     R3,BUFLEN               ; PUT PACKET LENGTH IN BUFLEN
 4912  073036  012164  000000                MOV     (R1)+,DESTIN(R4)        ; MOVE FIRST TWO BYTES OF ADDRESS
 4913  073042  012164  000002                MOV     (R1)+,DESTIN+2(R4)      ; MOVE BYTES THREE AND FOUR
 4914  073046  011164  000004                MOV     (R1),DESTIN+4(R4)       ; MOVE BYTES FIVE AND SIX
 4915  073052  005064  000006                CLR     SOURCC(R4)              ; LEAVE BLANK SPACE FOR SOURCE ADDRESS
 4916  073056  005064  000010                CLR     SOURCC+2(R4)            ;   SIX BYTES WORTH
 4917  073062  005064  000012                CLR     SOURCC+4(R4)
 4918  073066  013764  050544  000014        MOV     PROTOO,PROTOT(R4)       ; MOVE PROTOCALL TYPE INTO HEADER
 4919  073074  012702  050720                MOV     #LOPDIR,R2              ; MOVE LOOPDIRECT FORMAT HEADER LOC. TO R2
 4920  073100  012264  000016                MOV     (R2)+,LDSKIP(R4)        ;  SKIP COUNT
 4921  073104  011264  000020                MOV     (R2),LDFCT1(R4)         ;  FUNCTION CODE (FORWARD)
 4922  073110  013764  047762  000022        MOV     PHYADR,LDADR1(R4)       ;  LOCAL NODE ADDRESS
 4923  073116  013764  047764  000024        MOV     PHYADR+2,LDADR1+2(R4)   ;   SIX BYTES
 4924  073124  013764  047766  000026        MOV     PHYADR+4,LDADR1+4(R4)   ;
 4925  073132  016264  000010  000030        MOV     10(R2),LDFCT2(R4)       ;  FUNCTION CODE (REPLY)
 4926  073140  013764  047762  000032        MOV     PHYADR,LDADR2(R4)       ;  LOCAL NODE ADDRESS
 4927  073146  013764  047764  000034        MOV     PHYADR+2,LDADR2+2(R4)   ;   SIX BYTES
 4928  073154  013764  047766  000036        MOV     PHYADR+4,LDADR2+4(R4)   ;
 4929  073162  062704  000040                ADD     #LDADR2+6,R4            ; POINT R4 TO FIRST DATA BYTE
 4930  073166  010437  050570                MOV     R4,CMPBUF               ; STORE BUFFER LOCATION FOR DATA COMPARE
 4931  073172                                CALL    BLDBUF R4               ; BUILD DATA BUFFER
 4932  073202                                CALL    GETXNX  #XRGNXT         ; UPDATE POINTER TO NEXT RING ENTRY
 4933  073214  000411                        BR      60$                     ; EXIT
 4934  073216                        40$:    ERRDF   13,EMSG10,ERR1          ; TRANSMIT RING BOOKKEEPING ERROR
 4935  073216  104455                                                                TRAP    C$ERDF
 4936  073220  000015                                                                .WORD   13
 4937  073222  064656                                                                .WORD   EMSG10
 4938  073224  067460                                                                .WORD   ERR1
 4939  073226  000404                        BR      60$                     ; EXIT
 4940  073230                        45$:    ERRDF   14,EMSG14,ERR1          ; MESSAGE SIZE TOO SIG
 4941  073230  104455                                                                TRAP    C$ERDF
 4942  073232  000016                                                                .WORD   14
 4943  073234  064755                                                                .WORD   EMSG14
 4944  073236  067460                                                                .WORD   ERR1
 4945  073240                        60$:    RETURN
 4946
```

```
                                .SBTTL  BLDFAS  BUILD PACKET FOR FULL ASSIST TRANSMISSION.

                                ;--+
                                ; FUNCTIONAL DESCRIPTION:
                                ;               THIS SUBROUTINE BUILDS FULL ASSIST PACKETS FOR TRANSMISSION
                                ;               FROM THE UNA.  SOURCE ADDRESS, DESTINATION ADDRESS, PROT. TYPE
                                ;               AND FULL ASSIST HEADER INFO ARE ADDED TO THE MESSAGE BUFFER.
                                ;               THE MESSAGE BUFFER IS BUILT BY A CALL TO BLDBUF.
                                ;
                                ; INPUTS -      P1 - THE ADDRESS OF THE DESTINATION ADDRESS (FROM NODE TABLE)
                                ;               IMPLICIT - P$SIZE CONTAINS THE SIZE OF THE MESSAGE BUFFER DATA
                                ;                          XRGNXT POINTS TO THE NEXT AVAILABLE RING ENTRY
                                ;                          PHYADR HOLDS THE CURRENT LOCAL NODE ADDRESS
                                ;
                                ; OUTPUTS -     THE BUILT BUFFER
                                ;
                                ; CALLING PROCEDURE - CALL BLDFAS P1
                                ;
                                ; SIDE EFFECTS - XRGNXT SI UPDATED TO POINT TO THE NEXT RING ENTRY
                                ;
                                ; REGISTER USAGE - R1 HOLDS ADDRESS OF TARGET NODE ADDRESS
                                ;                  R2 HOLDS ADDRESS OF ASSIST NODE ADDRESS
                                ;                  R3 HOLDS THE PACKET LENGTH
                                ;                  R4 HOLDS ADDRESS OF NEXT RING ENTRY DATA BUFFER
                                ;
                                ;--+

073242                          BLDFAS::
073242                                  P$POP   R1,R2                   ; PUT ADDRESS OF TARGET ADDRESS INTO R1
                                                                        ;  AND ADDRESS OF ASSIST ADDRESS INTO R2
073246  013704  003762                  MOV     XRGNXT,R4               ; MOVE NEXT PACKET ADDRESS TO R4
073252  032764  100000  000004          BIT     #OWN,4(R4)              ; CHECK OWNERSHIP BIT
073260  001140                          BNE     40$                     ; IF DON'T OWN, BOOKKEEPING ERROR,
073262  016404  000002                  MOV     2(R4),R4                ; POINT R4 TO DATA BLOCK
073266  013703  002372                  MOV     P$SIZE,R3               ; PUT MESSAGE SIZE INTO R3
073272  062703  000060                  ADD     #60,R3                  ; ADD HEADER INFO TO LENGTH
073276  010337  050560                  MOV     R3,XFER                 ; PUT 'BYTES TRANSFERED' INTO WORD
073302  162737  000016  050560          SUB     #16,XFER                ;  AND CORRECT FOR HEADER
073310  022703  002756                  CMP     #XPKLEN,R3              ; SEE IF LONGER THAN ONE PACKET
073314  002527                          BLT     45$                     ;  IF YES, ERROR
073316  010337  050566                  MOV     R3,BUFLEN               ; PUT PACKET LENGTH IN BUFLEN
073322  011264  000000                  MOV     (R2),DESTIN(R4)         ; MOVE FIRST TWO BYTES OF ADDRESS
073326  016264  000002  000002          MOV     2(R2),DESTIN+2(R4)      ; MOVE BYTES THREE AND FOUR
073334  016264  000004  000004          MOV     4(R2),DESTIN+4(R4)      ; MOVE BYTES FIVE AND SIX
073342  005064  000006                  CLR     SOURCC(R4)              ; LEAVE BLANK SPACE FOR SOURCE ADDRESS
073346  005064  000010                  CLR     SOURCC+2(R4)            ;   SIX BYTES WORTH
073352  005064  000012                  CLR     SOURCC+4(R4)
073356  013764  050544  000014          MOV     PROTOC,PROTOT(R4)       ; MOVE PROTOCALL TYPE INTO HEADER
073364  012764  000000  000016          MOV     #0,FASKIP(R4)           ;  SKIP COUNT
073372  012764  000002  000020          MOV     #2,FAFCT1(R4)           ;  FUNCTION CODE (FORWARD)
073400  011164  000022                  MOV     (R1),FAADR1(R4)         ;  TARGET NODE ADDRESS
073404  016164  000002  000024          MOV     2(R1),FAADR1+2(R4)      ;   SIX BYTES
073412  016164  000004  000026          MOV     4(R1),FAADR1+4(R4)      ;
073420  012764  000002  000030          MOV     #2,FAFCT2(R4)           ;  FUNCTION CODE (FORWARD)
073426  011264  000032                  MOV     (R2),FAADR2(R4)         ;  ASSIST NODE ADDRESS
073432  016264  000002  000034          MOV     2(R2),FAADR2+2(R4)      ;   SIX BYTES
```

```
 5003  073440  016264  000004  000036          MOV     4(R2),FAADR 4(R4)       ;
 5004  073446  012764  000002  000040          MOV     #2,FAFCT3(R4)           ;  FUNCTION CODE (FORWARD)
 5005  073454  013764  047762  000042          MOV     PHYADR,FAADR3(R4)       ;  LOCAL NODE ADDRESS
 5006  073462  013764  047764  000044          MOV     PHYADR+2,FAADR3+2(R4)   ;   SIX BYTES
 5007  073470  013764  047766  000046          MOV     PHYADR+4,FAADR3+4(R4)   ;
 5008  073476  012764  000001  000050          MOV     #1,FAFCT4(R4)           ;  FUNCTION CODE (REPLY)
 5009  073504  013764  047762  000052          MOV     PHYADR,FAADR4(R4)       ;  LOCAL NODE ADDRESS
 5010  073512  013764  047764  000054          MOV     PHYADR+2,FAADR4+2(R4)   ;   SIX BYTES
 5011  073520  013764  047766  000056          MOV     PHYADR+4,FAADR4+4(R4)   ;
 5012  073526  062704  000060                  ADD     #FAADR4+6,R4            ; POINT R4 TO FIRST DATA BYTE
 5013  073532  010437  050570                  MOV     R4,CMPBUF               ; STORE BUFFER LOCATION FOR DATA COMPARE
 5014  073536                                  CALL    BLDBUF R4               ; BUILD DATA BUFFER
 5015  073546                                  CALL    GETXNX  #XRGNXT         ; UPDATE POINTER TO NEXT RING ENTRY
 5016  073560  000411                          BR      50$                     ; EXIT
 5017  073562                          40$:    ERRDF   28,EMSG10,ERR1          ; TRANSMIT RING BOOKKEEPING ERROR
 5018  073562  104455                                                                  TRAP    C$ERDF
 5019  073564  000034                                                                  .WORD   28
 5020  073566  064656                                                                  .WORD   EMSG10
 5021  073570  067460                                                                  .WORD   ERR1
 5022  073572  000404                          BR      50$                     ; EXIT
 5023  073574                          45$:    ERRDF   29,EMSG14,ERR1          ; MESSAGE SIZE TOO BIG
 5024  073574  104455                                                                  TRAP    C$ERDF
 5025  073576  000035                                                                  .WORD   29
 5026  073600  064755                                                                  .WORD   EMSG14
 5027  073602  067460                                                                  .WORD   ERR1
 5028  073604                          50$:    RETURN
 5029
```

```
                                        .SBTTL  BLDAST  BUILD TRANSMIT AND RECEIVE ASSIST PACKETS

                                        ;--+
073606                                  BLDAST::
073606                                          P$POP   R1,R2                   ; PUT DESTINATION ADDRESS INTO R1
                                                                                ;  ASSIST ADDRESS INTO R2
073612  013704  003762                          MOV     XRGNXT,R4               ; MOVE NEXT PACKET ADDRESS TO R4
073616  032764  100000  000004                  BIT     #OWN,4(R4)              ; CHECK OWNERSHIP BIT
073624  001125                                  BNE     40$                     ; IF DON'T OWN, BOOKKEEPING ERROR
073626  016404  000002                          MOV     2(R4),R4                ; POINT R4 TO DATA BLOCK
073632  013703  002372                          MOV     P$SIZE,R3               ; PUT MESSAGE SIZE INTO R3
073636  062703  000050                          ADD     #50,R3                  ; ADD HEADER INFO INTO LENGTH
073642  010337  050560                          MOV     R3,XFER                 ; PUT 'BYTES TRANSFERED' INTO WORD
073646  162737  000016  050560                  SUB     #16,XFER                ;  AND CORRECT FOR HEADER
073654  022703  002756                          CMP     #XPKLEN,R3              ; SEE IF LONGER THAN ONE PACKET
073660  002514                                  BLT     45$                     ;  IF YES, ERROR
073662  010337  050566                          MOV     R3,BUFLEN               ; PUT PACKET LENGTH INTO BUFLEN
073666  011164  000000                          MOV     (R1),DESTIN(R4)         ; MOVE DESTINATION ADDRESS INTO HEADER
073672  016164  000002  000002                  MOV     2(R1),DESTIN+2(R4)      ;  SIX BYTES WORTH
073700  016164  000004  000004                  MOV     4(R1),DESTIN+4(R4)
073706  005064  000006                          CLR     SOURCC(R4)              ; LEAVE BLANK SPACE FOR SOURCE ADDRESS
073712  005064  000010                          CLR     SOURCC+2(R4)            ;  SIX BYTES WORTH
073716  005064  000012                          CLR     SOURCC+4(R4)
073722  013764  050544  000014                  MOV     PROTOO,PROTOT(R4)       ; MOVE PROTOCALL TYPE INTO HEADER
073730  012764  000000  000016                  MOV     #0,FASKIP(R4)           ; SKIP COUNT
073736  012764  000002  000020                  MOV     #2,FAFCT1(R4)           ; FUNCTION CODE (FORWARD)
073744  011264  000022                          MOV     (R2),FAADR1(R4)         ;  TARGET NODE ADDRESS
073750  016264  000002  000024                  MOV     2(R2),FAADR1+2(R4)      ;   SIX BYTES
073756  016264  000004  000026                  MOV     4(R2),FAADR1+4(R4)
073764  012764  000002  000030                  MOV     #2,FAFCT2(R4)           ; FUNCTION CODE (FORWARD)
073772  013764  047762  000032                  MOV     PHYADR,FAADR2(R4)       ;  LOCAL NODE ADDRESS
074000  013764  047764  000034                  MOV     PHYADR+2,FAADR2+2(R4)   ;   SIX BYTES WORTH
074006  013764  047766  000036                  MOV     PHYADR+4,FAADR2+4(R4)
074014  012764  000001  000040                  MOV     #1,FAFCT3(R4)           ; FUNCTION CODE (REPLY)
074022  013764  047762  000042                  MOV     PHYADR,FAADR3(R4)       ; LOCAL NODE ADDRESS
074030  013764  047764  000044                  MOV     PHYADR+2,FAADR3+2(R4)
074036  013764  047766  000046                  MOV     PHYADR+4,FAADR3+4(R4)
074044  062704  000050                          ADD     #FAADR3+6,R4            ; POINT R4 TO FIRST DATA BYTE
074050  010437  050570                          MOV     R4,CMPBUF               ; STORE BUFFER LOCATION FOR DATA COMPARE
074054                                          CALL    BLDBUF  R4              ; BUILD DATA BUFFER
074064                                          CALL    GETXNX  #XRGNXT         ; UPDATE RING POINTER
074076  000411                                  BR      50$
074100                                  40$:    ERRDF   35,EMSG10,ERR1          ; TRANSMIT RING BOOKKEEPING ERROR
074100  104455                                                                          TRAP    C$ERDF
074102  000043                                                                        .WORD   35
074104  064056                                                                        .WORD   EMSG10
074106  067460                                                                        .WORD   ERR1
074110  000404                                  BR      50$
074112                                  45$:    ERRDF   36,EMSG14,ERR1          ; MESSAGE SIZE TOO BIG ERROR
074112  104455                                                                          TRAP    C$ERDF
074114  000044                                                                        .WORD   36
074116  064755                                                                        .WORD   EMSG14
074120  067460                                                                        .WORD   ERR1
074122                                  50$:    RETURN

                                        ;
                                        ;
```

```
                                .SBTTL  BLDREQ  BUILD REQUEST ID PACKETS FOR TRANSMIT.

                                ;--+
                                ; FUNCTIONAL DESCRIPTION:
                                ;               THIS SUBROUTINE BUILDS REQUEST ID PACKETS FOR TRANSMISSION
                                ;               FROM THE UNA.  SOURCE ADDRESS, DESTINATION ADDRESS,
                                ;               PROTOCALL TYPE, SEQUENCE NUMBER AND REQUEST ID
                                ;               HEADER INFOR ARE BUILT IN THIS SUBROUTINE.
                                ;
                                ; INPUTS -      IMPLICIT - THE DESTINATION ADDRESS IS CONTAINED IN ADRBUF.
                                ;
                                ; OUTPUTS -     BUILT MESSAGE PACKET
                                ;
                                ; CALLING PROCEDURE - CALL BLDREQ
                                ;
                                ; SIDE EFFECTS - THE MESSAGE PACKET IS BUILT AND CONTAINED IN THE
                                ;               BUFFER POINTED TO BY XRGNXT WHEN THE ROUTINE WAS
                                ;               ENTERED.  XRGNXT IS UPDATED TO POINT TO THE NEXT ENTRY.
                                ;
                                ; REGISTER USAGE - R1 HOLDS ADDRESS OF DESTINATION ADDRESS.
                                ;                 R2 IS A POINTER FOR REQUEST ID HEADER INFO.
                                ;                 R4 HOLDS ADDRESS OF NEXT RING ENTRY DATA BUFFER.
                                ;
                                ;--+

074124                          BLDREQ::
074124  013704  ??3762                 MOV     XRGNXT,R4               ; MOVE NEXT PACKET ADDRESS TO R4
074130  032764  100000  000004         BIT     #OWN,4(R4)              ; CHECK OWNERSHIP BIT
074136  001044                       BNE     40$                     ; IF DON'T OWN, BOOKKEEPING ERROR
074140  016404  000002               MOV     2(R4),R4                ; POINT R4 TO DATA BLOCK
074144  012737  000100  050566       MOV     #100,BUFLEN             ; MOVE BUFFER SIZE TO BUFLEN
074152  012701  002314               MOV     #ADRBUF,R1              ; MOVE ADDRESS OF DEST. ADR. TO R1
074156  012164  000000               MOV     (R1)+,DESTIN(R4)        ; MOVE FIRST TWO BYTE OF DEST. ADR.
074162  012164  000002               MOV     (R1)+,DESTIN+2(R4)      ;  AND BYTES THREE AND FOUR
074166  011164  000004               MOV     (R1),DESTIN+4(R4)       ;  AND LAST TWO BYTES
074172  005064  000006               CLR     SOURCC(R4)              ; LEAVE BLANK SPACE FOR SOURCE ADDR.
074176  005064  000010               CLR     SOURCC+2(R4)            ;  SIX BYTES WORTH
074202  005064  000012               CLR     SOURCC+4(R4)
074206  013764  050546  000014       MOV     PROTO2,PROTOT(R4)       ; MOVE PROTOCALL TYPE INTO HEADER
074214  012702  050712               MOV     #REQID,R2               ; MOVE REQUEST ID HEADER LOC. TO R2
074220  012264  000016               MOV     (R2)+,HEADER(R4)        ;  BYTE COUNT
074224  012264  000020               MOV     (R2)+,HEADER+2(R4)      ;  FUNCTION CODE (REQUEST ID)
074230  011264  000022               MOV     (R2),HEADER+4(R4)       ;  RECIEPT NO.
074234                               CALL    GETXNX #XRGNXT          ; UPDATE POINTER TO NEXT RING ENTRY
074246  000404                       BR      50$                     ; EXIT
074250                          40$:    ERRDF   21,EMSG10,ERR1          ; TRANSMIT RING BOOKKEEPING ERROR
074250  104455                                                                  TRAP    C$ERDF
074252  000025                                                                  .WORD   21
074254  064656                                                                  .WORD   EMSG10
074256  067460                                                                  .WORD   ERR1
074260                          50$:    RETURN

                                .SBTTL  GET?NX  GET NEXT TRANSMIT OR RECIEVE RING ENTRY

                                ;--+
```

```
                        ;+
                        ; FUNCTIONAL DESCRIPTION
                        ;               THIS SUBROUTINE GETS THE NEXT TRANSMIT OR RECIEVE RING
                        ;               ENTRY.  IT IS ENTERED AT SEPERATE POINTS DEPENDING ON
                        ;               WHICH RING IS BEING USED.
                        ;
                        ; INPUTS -      P1 - THE ADDRESS OF THE RING POINTER TO BE UPDATED.
                        ;
                        ; OUTPUTS -     THE RING POINTER IS UPDATED TO POINT TO THE NEXT AVAILABLE
                        ;                 ENTRY.
                        ;
                        ; CALLING PROCEDURE - CALL  GETXNX  #P1          ; FOR TRANSMIT UPDATES
                        ;                     CALL  GETRNX  #P1          ; FOR RECIEVE UPDATES
                        ;
                        ; SIDE EFFECTS - NONE
                        ;
                        ; REGISTER USAGE - R1 POINTS TO THE FIRST ENTRY IN THE RING
                        ;                  R2 POINTS TO THE LAST ENTRY IN THE RING
                        ;                  R3 IS THE ADDRESS OF THE RING POINTER TO BE UPDATED
                        ;
                        ;--+

074262                  GETRNX::
074262  013701  003754          MOV     RRGSRT,R1               ; MOVE FIRST RING ENTRY TO R1
074266  013702  003770          MOV     RRGLST,R2               ; MOVE LAST RING ENTRY TO R2
074272  000404                  BR      GETCOM                  ; GO TO COMMON CODE

074274                  GETXNX::
074274  013701  003752          MOV     XRGSRT,R1               ; MOVE FIRST RING ENTRY TO R1
074300  013702  003766          MOV     XRGLST,R2               ; MOVE LAST RING ENTRY TO R2
074304                  GETCOM: P$POP   R3                      ; GET ADDRESS OF RING POINTER IN R3
074306  021302                  CMP     (R3),R2                 ; SEE IF POINTER POINTS TO LAST RING
074310  001403                  BEQ     15$                     ;  IF YES, BRANCH
074312  062713  000012          ADD     #10.,(R3)               ;  ELSE, ADD ENTRY LENGTH TO POINTER
074316  000401                  BR      25$                     ; EXIT
074320  010113          15$:    MOV     R1,(R3)                 ; POINT POINTER TO FIRST ENTRY IN RING
074322                  25$:    RETURN


                        .SBTTL  BLDBUF  BUILD MESSAGE BUFFERS

                        ;--+
                        ; FUNCTIONAL DESCRIPTION
                        ;               THIS SUBROUTINE CREATES A MESSAGE BUFFER TO BE USED
                        ;               FOR TRANSMISSION.
                        ;
                        ; INPUTS -      P1 - BUFFER LOCATION TO PUT BUILT MESSAGE
                        ;               IMPLICIT - P$SIZE CONTAINS THE SIZE THE BUFFER IS TO BE
                        ;                          P$TYPE CONTAINS THE MESSAGE TYPE
                        ; OUTPUTS -     IMPLICIT - BUFFER STARTING AT LOCATION P1 CONTAINS A
                        ;                          MESSAGE P$SIZE BYTES LONG USING THE MESSAGE
                        ;                          TYPE SPECIFIED BY P$TYPE.
                        ;
                        ; CALLING PROCEEDURE - CALL BLDBUF P1
                        ;
                        ; SIDE EFFECTS - NONE
```

```
5198                                    ;
5199                                    ; REGISTER USAGE - R1 POINTS TO THE NEXT BYTE OF STORED MESSAGE TO BE BUILT
5200                                    ;                  IN THE MESSAGE BUFFER.
5201                                    ;                R2 = (MESSAGE TYPE X 2), USED AS OFFSET FOR POINTERS
5202                                    ;                R3 POINTS TO THE NEXT BYTE OF THE BUFFER UNDER CONSTRUCTION
5203                                    ;                R4 POINTS TO THE LAST BYTE OF THE BUFFER UNDER CONSTRUCTION
5204                                    ;
5205                                    ;--+
5206
5207  074324                            BLDBUF::
5208  074324                                    P$POP   R3                      ; PUT BUFFER ADDRESS INTO R3
5209  074326  013702  002370                   MOV     P$TYPE,R2               ; PUT MESSAGE TYPE INTO R2
5210  074332  006302                            ASL     R2                      ; MULTIPLY BY 2
5211  074334  013704  002372                   MOV     P$SIZE,R4               ; PUT SIZE INTO R4
5212  074340  060304                            ADD     R3,R4                   ; MAKE R4 = LAST BYTE OF BUFFER
5213  074342  016201  003316                   MOV     MSGAD(R2),R1            ; POINT R1 TO FIRST BYTE OF STORED MESSAGE
5214  074346  005037  050542                   CLR     COUNT                   ; CLEAR BYTE COUNTER
5215  074352  005237  050542           10$:    INC     COUNT                   ; COUNT NO. OF BYTES COPIED
5216  074356  112123                            MOVB    (R1)+,(R3)+             ; PUT BYTE IN BUFFER
5217  074360  026237  003300  050542           CMP     MSGCNT(R2),COUNT        ; ARE WE AT END OF STORED MESSAGE
5218  074366  001004                            BNE     20$                     ;  IF NO, CHECK IF DONE
5219  074370  016201  003316                   MOV     MSGAD(R2),R1            ;  ELSE, POINT R1 TO BEGINING
5220  074374  005037  050542                   CLR     COUNT                   ;  AND CLEAR COUNTER
5221  074400  020304                    20$:    CMP     R3,R4                   ; IS BUFFER FILLED?
5222  074402  001363                            BNE     10$                     ;  IF NO, LOOP
5223  074404                                    RETURN                          ;  ELSE, RETURN
5224
```

```
5225
5226                            .SBTTL  DATCMP  COMPARE DATA BUFFERS
5227
5228                            ;--+
5229                            ; FUNCTIONAL DESCRIPTION
5230                            ;               THIS SUBROUTINE COMPARES TWO DATA BUFFERS BYTE BY BYTE.
5231                            ;               IF COMPARISON ERRORS OCCURED, LOCATION, EXPECTED DATA
5232                            ;               AND RECIEVED DATA ARE PRINTED OUT FOR THE FIRST FIVE
5233                            ;               ERRORS.  THE TOTAL NUMBER OF ERRORS IS ALSO PRINTED.
5234                            ;
5235                            ; INPUTS -      P1 - THE SIZE (IN BYTES) OF     BUFFER TO BE COMPARED.
5236                            ;               P2 - THE ADDRESS OF BUFFER    .OMPARE OTHER BUFFER AGAINST.
5237                            ;               P3 - THE ADDRESS OF THE SEC.   BUFFER.
5238                            ;
5239                            ; OUTPUTS -     P4 - THE NUMBER OF COMPARISON ERRORS.
5240                            ;
5241                            ; CALLING PROCEDURE - CALL DATCMP P1,P2,P3
5242                            ;                     P$POP P4
5243                            ;
5244                            ; SIDE EFFECTS - NONE.
5245                            ;
5246                            ; REGISTER USAGE - R1 CONTAINS THE COMPARE SIZE
5247                            ;                  R2 CONTAINS THE ADDRESS OF THE BYTE BEING COMPARED IN BUFFER 1
5248                            ;                  R3 CONTAINS THE ADDRESS OF THE BYTE BEING COMPARED IN BUFFER 2
5249                            ;                  R4 CONTAINS THE BYTE OFFSET (BYTES FROM BEGINING OF BUFFER
5250                            ;
5251                            ;--+
5252
5253  074406                    DATCMP::
5254  074406                            P$POP   R1,R2,R3                ; PUT COMPARE SIZE IN R1
5255                                                                    ;  BUFFER 1 ADDRESS IN R2 AND
5256                                                                    ;  BUFFER 2 ADDRESS IN R3
5257  074414  005004                    CLR     R4                      ; INITIALIZE BYTE OFFSET
5258  074416  005037  050550            CLR     TEMP                    ;  AND ERROR COUNTER
5259
5260  074422  005204            10$:    INC     R4                      ; INCREMENT OFFSET COUNTER
5261  074424  121213                    CMPB    (R2),(R3)               ; COMPARE BUFFERS
5262  074426  001421                    BEQ     20$                     ;  IF SAME, BRANCH
5263  074430  005237  050550            INC     TEMP                    ; INCREMENT ERROR COUNTER
5264  074434  022737  000005  050550    CMP     #5,TEMP                 ; IF MORE THAN 5 ERRORS,
5265  074442  002413                    BLT     20$                     ; DON'T PRINT MESSAGE
5266  074444                            PRINTX  #CMPER1,R4,(R2),(R3)    ; PRINT ERROR MESSAGE
5267  074444  011346                                                                    MOV     (R3),-(SP)
5268  074446  011246                                                                    MOV     (R2),-(SP)
5269  074450  010446                                                                    MOV     R4,-(SP)
5270  074452  012746  066754                                                           MOV     #CMPER1,-(SP)
5271  074456  012746  000004                                                           MOV     #4,-(SP)
5272  074462  010600                                                                    MOV     SP,R0
5273  074464  104415                                                                    TRAP    C$PNTX
5274  074466  062706  000012                                                           ADD     #12,SP
5275  074472  005202            20$:    INC     R2                      ; INCREMENT BUFFER 1 POINTER
5276  074474  005203                    INC     R3                      ; INCREMENT BUFFER 2 POINTER
5277  074476  005301                    DEC     R1                      ; DECREMENT COMPARE SIZE
5278  074500  001350                    BNE     10$                     ;  IF NOT FINISHED, GO BACK FOR MORE
5279  074502  022737  000000  050550    CMP     #0,TEMP                 ; WERE THERE ANY ERRORS?
5280  074510  001412                    BEQ     30$                     ;  IF NO, EXIT
```

```
                        PRINTX  #CMPER2,TEMP
                                                                        MOV     TEMP,-(SP)
                                                                        MOV     #CMPER2,-(SP)
                                                                        MOV     #2,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #6,SP
30$:    RETURN  TEMP                    ; RETURN WITH ERROR COUNT ON STACK


.SBTTL  WRITES  WRITE DATA ONTO SUMMARY TABLE

;--+
; FUNCTIONAL DESCRIPTION:
;               THIS SUBROUTINE UPDATES THE SUMMARY TABLE DATA FOR
;               THE NODES SPECIFIED IN THE CALL STATEMENT.  EITHER ONE
;               OR TWO NODES CAN UPDATED PER CALL.  AFTER THE CALL,
;               THE SUMMARY DATA COUNTERS ARE CLEARED.  THE SUMMARY TABLE
;               IS CHECKED FOR A MATCHING NODE ADDRESS AND UPDATES THE
;               DATE FOR THAT NODE, OR ADDS THE NODE TO THE TABLE IF IT
;               DOESN'T EXIT.  AN ERROR IS REPORTED IF THE END OF THE TABLE
;               IS REACHED.
;
; INPUTS -      P1 - THE NUMBER OF NODES TO UPDATE (1 OR 2).
;               P2 - THE ADDRESS OF THE FIRST NODE ADDRESS.
;               P3 - THE ADDRESS OF THE SECOND NODE ADDRESS IF P1 = 2 OR
;                    BLANK IF P1 = 1.
;
; OUTPUTS -     THE SUMMARY TABLE IS UPDATED.
;
; CALLING PROCEDURE - CALL WRITES P1,P2(,P3)
;
; SIDE EFFECTS - THE SUMMARY COUNTERS ARE CLEARED.
;
; REGISTER USAGE - R1 POINTS TO THE CURRENT LOCATION IN THE SUMMARY TABLE.
;                  R2 POINTS TO THE NODE TO BE UPDATED'S ADDRESS.
;                  R3 IS SCRATCH
;                  R4 HOLDS THE SECOND NODE TO BE UPDATED ADDRESS.
;
;--+

WRITES: P$POP   TEMP                    ; SEE HOW MANY NODES TO WRITE
        CMP     TEMP,#1                 ;  IF ONLY ONE, GET ADDRESS
        BNE     5$
        P$POP   R2
        BR 6$
5$:     P$POP   R2,R4                   ;  IF TWO, GET BOTH ADDRESSES
6$:     MOV     #STATBL,R1              ; MOVE STATISTICAL TABLE ADDRESS INTO R1
7$:     TST     (R1)                    ; SEE IF SLOT IS EMPTY
        BEQ     15$                     ;  IF YES, BR
        CMP     (R1),#-1                ; SEE IF TABLE FULL
        BEQ     25$                     ;  IF YES, ERROR
        CALL    CMPADR  R1,R2           ; LOOK FOR MATCHING ADDRESS
        P$POP   R3
        BEQ     20$                     ; IF YES, BR
        ADD     #26,R1                  ; ELSE, POINT R1 TO NEXT ENTRY
```

```
 5337  074630  000761                        BR      7$                      ;  AND CHECK AGAIN
 5338  074632  011211                15$:    MOV     (R2),(R1)               ; ADD NEW ADDRESS TO TABLE
 5339  074634  016261  000002  000002        MOV     2(R2),2(R1)             ;  SIX BYTES WORTH
 5340  074642  016261  000004  000004        MOV     4(R2),4(R1)             ;
 5341  074650  062701  000006                20$:    ADD     #6,R1           ; POINT R1 TO DATA
 5342  074654  063721  050500                        ADD     S.NREC,(R1)+    ;  UPDATE SUMMARY DATA, RECEIVES NOT COMPLETE
 5343  074660  063721  050476                        ADD     S.REC,(R1)+     ;   RECEIVES COMPLETE
 5344  074664  063721  050502                        ADD     S.LEN,(R1)+     ;   LENGTH ERRORS
 5345  074670  063721  050504                        ADD     S.COMP,(R1)+    ;   COMPARE ERRORS
 5346  074674  063721  050506                        ADD     S.BYTE,(R1)+    ;   BYTES COMPARED
 5347  074700  103001                              BCC     22$             ;     IF OVERFLOW, INCREMENT NEXT WORD
 5348  074702  005511                              ADC     (R1)
 5349  074704  062701  000002                22$:    ADD     #2,R1           ;     POINT R1 TO NEXT DATA
 5350  074710  063721  050510                        ADD     S.XFER,(R1)+    ;   BYTES TRANSFERED
 5351  074714  103001                              BCC     23$             ;     IF OVERFLOW, INCREMENT NEXT WORD
 5352  074716  005511                              ADC     (R1)
 5353  074720  062701  000002                23$:    ADD     #2,R1           ;     POINT R1 TO NEXT DATA
 5354  074724  005337  050550                        DEC     TEMP            ; DECR NO OF NODES COUNTER
 5355  074730  001414                              BEQ     30$             ; IF NO MORE, EXIT
 5356  074732  010402                              MOV     R4,R2           ; POINT R2 TO NEXT NODE
 5357  074734  000715                              BR      6$              ;  AND UPDATE SUMMARY DATA
 5358  074736                        25$:    PRINTF  #TABFUL,#SUMM           ; PRINT TABLE FULL MESSAGE
 5359  074736  012746  053706                                                                MOV     #SUMM,-(SP)
 5360  074742  012746  053560                                                                MOV     #TABFUL,-(SP)
 5361  074746  012746  000002                                                                MOV     #2,-(SP)
 5362  074752  010600                                                                        MOV     SP,R0
 5363  074754  104417                                                                        TRAP    C$PNTF
 5364  074756  062706  000006                                                                ADD     #6,SP
 5365  074762  005037  050500          30$:    CLR     S.NREC                  ; CLEAR SUMMARY DATA COUNTERS
 5366  074766  005037  050476                  CLR     S.REC
 5367  074772  005037  050502                  CLR     S.LEN
 5368  074776  005037  050504                  CLR     S.COMP
 5369  075002  005037  050506                  CLR     S.BYTE
 5370  075006  005037  050510                  CLR     S.XFER
 5371  075012                                  RETURN
 5372
```

```
5373
5374                                    .SBTTL  BINDEC  CONVERT A 32 BIT BINARY NUMBER TO DECIMAL
5375
5376                                    ;--+
5377                                    ; FUNCTIONAL DESCRIPTION:
5378                                    ;               THIS SUBROUTINE CONVERTS A 32 BIT BINARY NUMBER TO
5379                                    ;               A DECIMAL NUMBER REPRESENTED AS AN ASCIZ STRING.
5380                                    ;
5381                                    ; INPUTS -      P1 - THE ADDRESS OF THE FIRST WORD OF BINARY DATA
5382                                    ;                    BITS 0-15.  THE SECOND WORD, BITS 16-31, IS
5383                                    ;                    EXPECTED TO IMMEDIATELY FOLLOW THE FIRST WORD.
5384                                    ;
5385                                    ; OUTPUTS -     THE ASCII STRING WILL BE LOCATED STARTING AT DECSTR
5386                                    ;
5387                                    ; SIDE EFFECTS - NONE
5388                                    ;
5389                                    ; REGISTER USAGE - R1 POINTS TO BITS 0-15 OF BINARY DATA
5390                                    ;                  R2 POINTS TO BITS 16-31 OF BINARY DATA
5391                                    ;                  R3 POINTS TO THE OUTPUT STRING
5392                                    ;                  R4 POINTS TO THE POWERS OF 10 TABLE
5393                                    ;
5394                                    ;--+
5395
5396  075014                            BINDEC::
5397  075014                                    P$POP   R1                      ; PUT ADDRESS OF BINARY WORD INTO R1
5398  075016  010546                            MOV     R5,-(SP)
5399  075020  010102                            MOV     R1,R2                   ; PUT ADDRESS OF SECOND WORD INTO R2
5400  075022  062702  000002                    ADD     #2,R2
5401  075026  012703  075200                    MOV     #DECSTR,R3              ; PUT ADDRESS OF OUPUT STRING INTO R3
5402  075032  012704  075130                    MOV     #TENPWR,R4              ; ADDRESS OF TEN POWER TABLE
5403  075036  012705  075132                    MOV     #TENPWR+2,R5
5404  075042  012737  000012  075116            MOV     #10.,4$
5405  075050  005037  075214          1$:       CLR     PART                    ; CLEAR PARTIAL COUNTER
5406  075054  161411                  2$:       SUB     (R4),(R1)               ; SUBTRACT 10 POWER
5407  075056  005612                            SBC     (R2)
5408  075060  161512                            SUB     (R5),(R2)
5409  075062  002403                            BLT     3$                      ; BRANCH IF 10 POWER TOO LARGE
5410  075064  005237  075214                    INC     PART                    ; ELSE ADD 1 TO PARTIAL
5411  075070  000771                            BR      2$                      ; LOOP
5412  075072  062411                  3$:       ADD     (R4)+,(R1)              ; RESTORE BINARY WORDS
5413  075074  005512                            ADC     (R2)                    ; AND POINT R4 TO NEXT TABLE ENTRIES
5414  075076  062412                            ADD     (R4)+,(R2)
5415  075100  022525                            CMP     (R5)+,(R5)+
5416  075102  052737  000060  075214            BIS     #'0,PART                ; CHANGE PARTIAL TO ASCII
5417  075110  113723  075214                    MOVB    PART,(R3)+              ; AND PUT INTO OUTPUT STRING
5418  075114  005327                            DEC     (PC)+                   ; HAVE WE DONE ALL 10 DIGITS
5419  075116  000000                  4$:       .WORD   0
5420  075120  001353                            BNE     1$                      ; IF NO, BRANCH
5421  075122  105023                            CLRB    (R3)+                   ; IF YES, TERMINATE WITH ZERO
5422  075124  012605                            MOV     (SP)+,R5
5423  075126                                    RETURN
5424
5425  075130  145000                TENPWR:     145000                          ; 1.0 E09
5426  075132  035632                            35632
5427  075134  160400                            160400                          ; 1.0 E08
5428  075136  002765                            2765
```

```
 5429  075140  113200                  113200                          ; 1.0 E07
 5430  075142  000230                  230
 5431  075144  041100                  041100                          ; 1.0 E06
 5432  075146  000017                  17
 5433  075150  103240                  103240                          ; 1.0 E05
 5434  075152  000001                  1
 5435  075154  023420                  23420                           ; 1.0 E04
 5436  075156  000000                  0
 5437  075160  001750                  1750                            ; 1.0 E03
 5438  075162  000000                  0
 5439  075164  000144                  144                             ; 1.0 E02
 5440  075166  000000                  0
 5441  075170  000012                  12                              ; 1.0 E01
 5442  075172  000000                  0
 5443  075174  000001                  1                               ; 1.0 E00
 5444  075176  000000                  0
 5445
 5446  075200  000014          DECSTR::.BLKB   12.                     ; 12 BYTES FOR ASCIZ OUTPUT STRING
 5447  075214  000000          PART::  .WORD   0                       ; PARTIAL COUNTER
 5448
```

```
 5449
 5450                                   .SBTTL  COMMAND LINE TRAVERSE ROUTINES
 5451
 5452                                   ;++
 5453                                   ;       P$TRV SUBROUTINE
 5454                                   ;
 5455                                   ;PARSE THE COMMAND LINE SUBROUTINE
 5456                                   ;TAKE ACTIONS (VIA ACTION TREE) AS PARSING LINE
 5457                                   ;PARSING DIRECTIONS FROM ''CLI PARSING NODES''
 5458                                   ;   REGS USED:
 5459                                   ;
 5460                                   ;       R1,R5=SCRATCH                                   P$NUM=NUMERIC CODE FROM DATA
 5461                                   ;       R2=ACTION CODE PARAMETER FROM TREE
 5462                                   ;       R3=PARSE TREE POINTER
 5463                                   ;       R4=INPUT STRING POINTER
 5464                                   ; CALLING SEQUENCE:
 5465                                   ;       JSR     PC,P$TRV
 5466                                   ;--
 5467
 5468  075216                           P$TRV:
 5469  075216  013704  003144                  MOV     P$BUFA,R4
 5470  075222  013703  003146                  MOV     P$TREE,R3
 5471  075226  121327  000003           P$TR5:  CMPB    (R3),#3                 ;SEE IF ONE OF FIRST THREE SPECIAL CODES
 5472  075232  003405                           BLE     5$                      ;IF YES, DON'T CHECK INPUT STRING
 5473  075234  105714                           TSTB    (R4)                    ;SEE IF ANY CHARS LEFT IN INPUT STRING
 5474  075236  001441                           BEQ     P$EXIT                  ;BR IF NO
 5475  075240  121327  000013                  CMPB    (R3),#11.               ;SEE IF SPECIAL CLI CHAR CODE OR ASCII
 5476  075244  003023                           BGT     20$                     ;BR IF REGULAR ASCII CHAR.
 5477  075246  111301                   5$:     MOVB    (R3),R1                 ;GET SPECIAL CHAR CODE INTO R5
 5478  075250  006301                           ASL     R1
 5479  075252  016101  075266                  MOV     10$(R1),R1              ;BUILD TRAVERSE ROUTINE ADDRESS
 5480  075256  062701  075266                  ADD     #10$,R1
 5481  075262  004711                           JSR     PC,(R1)                 ;JSR TO SPECIAL CLI TRAVERSE ROUTINE
 5482  075264  000760                           BR      P$TR5                   ;GO SEE IF MORE OF STRING LEFT
 5483
 5484
 5485  075266  000114                   10$:    .WORD   TRVERR-10$              ;TRAVERSE TABLE FOR ''CLI FUNCTIONS''
 5486  075270  000134                           .WORD   TRVEXI-10$              ;1
 5487  075272  000152                           .WORD   TRVBR-10$               ;2
 5488  075274  000162                           .WORD   TRVBIF-10$              ;3
 5489  075276  000204                           .WORD   TRVSPA-10$              ;4
 5490  075300  000270                           .WORD   TRVNUM-10$              ;5
 5491  075302  000612                           .WORD   TRVALP-10$              ;6
 5492  075304  000656                           .WORD   TRVALN-10$              ;7
 5493  075306  000270                           .WORD   TRVOCT-10$              ;8
 5494  075310  000256                           .WORD   TRVDEC-10$              ;9
 5495  075312  000744                           .WORD   TRVSTR-10$              ;10
 5496
 5497                                   ;NOT A SPECIAL CODE
 5498
 5499  075314  121314                   20$:    CMPB    (R3),(R4)               ;SEE IF FIRST CHAR OF STRING IS A MATCH
 5500  075316  001403                           BEQ     22$                     ;BR IF A MATCH
 5501  075320  004737  075364                  JSR     PC,TRVBRC               ;IF NOT A MATCH, GO TAKE MISS BRANCH
 5502  075324  000740                           BR      P$TR5                   ; THEN GO BACK PT'G TO MISS NODE
 5503  075326  005204                   22$:    INC     R4                      ;IF A MATCH, INCR. CHAR POINTER
 5504  075330  004737  075344                  JSR     PC,TRVACT               ; GO DO ACTION DEFINED BY
```

```
5505  075334  062703  000004            ADD     #4,R3                   ; ACTION CODE IN CLI NODE, THEN
5506                                                                      ; ADJUST PTR TO NEXT CLI NODE
5507  075340  000732                    BR      P$TR5
5508
5509  075342  000207            P$EXIT: RTS     PC                      ;RETURN FROM PARSER
5510
5511                                    ;-------------------------------------------------------------------------- -
5512
5513                                    ;GOTO USER ACTION ROUTINE
5514  075344  116302  000001    TRVACT: MOVB    1(R3),R2                ;GET ACTION CODE FROM CLI NODE
5515  075350  042702  177400            BIC     #177400,R2              ;CLEAR ANY SIGN EXTENSION
5516  075354  013701  003150            MOV     P$ACT,R1                ;GET ADDRESS OF CLI ACTION ROUTINE
5517  075360  004711                    JSR     PC,(R1)                 ;GO DO ACTION DEFINED BY CODE
5518  075362  000207                    RTS     PC                      ;RETURN TO CALLING CODE
5519
5520                                    ;TAKE BRANCH IN TREE
5521  075364  016301  000002    TRVBRC: MOV     2(R3),R1                ;GET BRANCH DISPLACEMENT FROM TREE
5522  075370  060103                    ADD     R1,R3                   ; AND POINT R3 TO THE "MISS" NODE
5523  075372  000207                    RTS     PC                      ; RETURN TO P$TRV
5524
5525                                    ;NO BRANCH TAKEN
5526  075374  062703  000004    TRVNOB: ADD     #4,R3                   ;THINGS OK, UPDATE R3 TO POINT TO NEXT
5527  075400  000207                    RTS     PC                      ; NODE AND RETURN TO P$TRV
5528
5529                                    ;--------------------------------------------------------------------------
5530                                    ;ERROR HANDLING
5531  075402  004737  075344    TRVERR: JSR     PC,TRVACT               ;TAKE ERROR ACTION
5532  075406  112737  177777  003161        MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
5533  075414  005726                    TST     (SP)+                   ;GET RID OF "JSR PUSH TO TRVERR"
5534  075416  000137  075342            JMP     P$EXIT                  ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
5535
5536                                    ;EXIT ACTION CODE
5537  075422  004737  075344    TRVEXI: JSR     PC,TRVACT               ;TAKE EXIT ACTION
5538  075426  105037  003161            CLRB    P$GDBD                  ;SET GOOD/BAD FLAG TO "SUCCESS (0)"
5539  075432  005726                    TST     (SP)+                   ;GET RID OF "JSR PUSH TO TRVEXI"
5540  075434  000137  075342            JMP     P$EXIT                  ;RETURN DIRECT TO EXIT OF P$TRV ROUTINE
5541
5542                                    ;BRANCH ACTION CODE
5543  075440  004737  075344    TRVBR:  JSR     PC,TRVACT               ;GO TAKE BRANCH ACTION
5544  075444  000137  075364            JMP     TRVBRC
5545
5546                                    ;BRANCH-IF ACTION CODE
5547  075450  004737  075344    TRVBIF: JSR     PC,TRVACT
5548  075454  105737  003161            TSTB    P$GDBD                  ;SEE IF P$GDBD SET OR CLEARED BY ACTION
5549  075460  001402                    BEQ     1$                      ;IF CLEAR FALL THRU TO NEXT NODE
5550  075462  000137  075364            JMP     TRVBRC                  ;ELSE TAKE THE "MISS" BRANCH
5551  075466  000137  075374    1$:     JMP     TRVNOB                  ;JUST UPDATE TO NEXT NODE IF THINGS OK
5552
5553                                    ;SPACE ACTION CODE
5554  075472  005001            TRVSPA: CLR     R1                      ;CLEAR "SPACE OR TAB FOUND" FLAG
5555  075474  121427  000011    1$:     CMPB    (R4),#11                ;SEE IF CHAR. IN CMD LINE= TAB
5556  075500  001003                    BNE     2$                      ;BR IF NO, NOT A TAB
5557  075502  005204                    INC     R4                      ;INC INPUT STRING POINTER
5558  075504  005201                    INC     R1                      ;INDICATE A TAB FOUND
5559  075506  000772                    BR      1$                      ;GO CHECK NEXT CHAR
5560
```

```
 5561  075510  121427  000040          2$:     CMPB    (R4),#40                ;SEE IF CHAR. IN CMD LINE= SPACE
 5562  075514  001003                          BNE     10$                     ;BR IF NO, NON-SPACE OR NON-TAB CHAR.
 5563  075516  005204                          INC     R4                      ;INC INPUT STRING POINTER
 5564  075520  005201                          INC     R1                      ;INDICATE A SPACE FOUND
 5565  075522  000764                          BR      1$                      ;GO CHECK NEXT CHAR
 5566  075524  005701                  10$:    TST     R1                      ;SEE IF ANY SPACES OR TABS FOUND
 5567  075526  001404                          BEQ     15$                     ;BR IF NO, TAKE NO ACTION
 5568  075530  004737  075344                  JSR     PC,TRVACT               ;GO TAKE ACTION IF ANY FOUND
 5569  075534  000137  075374                  JMP     TRVNOB                  ;JUST GO UPDATE R3 TO NEXT NODE IF OK
 5570  075540  000137  075364          15$:    JMP     TRVBRC                  ;TAKE BRANCH (MISS) IF NONE FOUND
 5571
 5572
 5573  075544  012737  000012  003156  TRVDEC: MOV     #10.,P$RADX             ;USE DECIMAL AS RADIX AND ASSUME +
 5574  075552  000137  075564                  JMP     TRVNMA
 5575  075556                          TRVOCT: ;(SAME AS TRVNUM SINCE DEFAULT RADIX IS OCTAL)
 5576  075556  012737  000010  003156  TRVNUM: MOV     #8.,P$RADX              ;USE OCTAL AS RADIX AND ASSUME +
 5577  075564                          TRVNMA: PUSH    R5
 5578  075566  005001                          CLR     R1                      ;CLEAR DIGIT COUNTER
 5579  075570  121427  000053                  CMPB    (R4),#'+                ;SEE IF THERE'S A + SIGN THERE
 5580  075574  001001                          BNE     10$                     ; BR IF NO
 5581  075576  000406                          BR      11$                     ; ELSE P$RADX ALREADY SAYS +, JUST BR
 5582  075600  121427  000055          10$:    CMPB    (R4),#'-                ;SEE IF THERE'S A - SIGN THERE
 5583  075604  001004                          BNE     1$                      ; BR IF NO
 5584  075606  112737  177777  003157          MOVB    #-1,P$RADX+1            ;SET 'MINUS FLAG' (HI BYTE OF P$RADX)
 5585  075614  005204                  11$:    INC     R4                      ;BUMP R4 TO POINT TO FIRST CHAR
 5586
 5587  075616  121427  000060          1$:     CMPB    (R4),#60                ;SEE IF CHAR. LESS THAN A '0'
 5588  075622  002434                          BLT     2$                      ;BR IF YES (NOT NUMERIC)
 5589  075624  121427  000067                  CMPB    (R4),#67                ;SEE IF CHAR. GREATER THAN A '7'
 5590  075630  003426                          BLE     13$                     ; BR IF YES
 5591  075632  123727  003156  000012          CMPB    P$RADX,#10.             ;SEE IF IN DECIMAL MODE
 5592  075640  001417                          BEQ     12$                     ; BR IF YES (CAN USE HIGHER LIMIT)
 5593  075642  121427  000071                  CMPB    (R4),#71                ;SEE IF DIGIT WAS A 8 OR 9
 5594  075646  003022                          BGT     2$                      ;BR IF NON-NUMERIC
 5595  075650                                  PRINTF  #CLIBRX                 ;ELSE WAS A 8 OR 9 WHEN IN OCTAL RADIX
 5596  075650  012746  052554                                                          MOV     #CLIBRX,-(SP)
 5597  075654  012746  000001                                                          MOV     #1,-(SP)
 5598  075660  010600                                                                  MOV     SP,R0
 5599  075662  104417                                                                  TRAP    C$PNTF
 5600  075664  062706  000004                                                          ADD     #4,SP
 5601  075670  112737  177777  003161          MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
 5602  075676  000475                          BR      5$                      ; PRINT ERROR AND TAKE MISS
 5603
 5604  075700  121427  000071          12$:    CMPB    (R4),#71                ;SEE IF CHAR. GREATER THAN A '9'
 5605  075704  003003                          BGT     2$                      ;BR IF YES (NOT NUMERIC)
 5606  075706  005204                  13$:    INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR.
 5607  075710  005201                          INC     R1                      ;INDICATE A NUMERIC FOUND
 5608  075712  000741                          BR      1$                      ;GO LOOK AT NEXT CHAR.
 5609
 5610  075714  005701                  2$:     TST     R1                      ;SEE IF FOUND ANY NUMERICS
 5611  075716  001465                          BEQ     5$                      ;BR IF NO, TAKE 'MISS' BRANCH
 5612  075720  010405                          MOV     R4,R5                   ;GET POINTER TO START OF NUMERIC STRING
 5613  075722  160105                          SUB     R1,R5
 5614  075724  005037  003154                  CLR     P$NUM                   ;CLEAR LOC. WHERE VALUE WILL BE STORED
 5615  075730  112502                  3$:     MOVB    (R5)+,R2                ;GET ASCII CHAR AND CONVERT IT TO A #
 5616  075732  162702  000060                  SUB     #60,R2
```

```
 5617  075736  006337  003154          ASL     P$NUM                   ;SHIFT CURRENT VALUE TO MAKE ROOM
 5618  075742  103440                  BCS     7$                      ;ERROR IF NUMBER TOO BIG
 5619  075744  013737  003154  003152  MOV     P$NUM,P$CNT             ;SAVE FOR LATER IN CASE DECIMAL RADIX
 5620  075752  006337  003154          ASL     P$NUM
 5621  075756  103432                  BCS     7$                      ;ERROR IF NUMBER TOO BIG
 5622  075760  006337  003154          ASL     P$NUM
 5623  075764  103427                  BCS     7$                      ;ERROR IF NUMBER TOO BIG
 5624  075766  123727  003156  000012  CMPB    P$RADX,#10.             ;SEE IF DECIMAL RADIX
 5625  075774  001004                  BNE     4$                      ;BR IF NOT EQUAL
 5626  075776  063737  003152  003154  ADD     P$CNT,P$NUM
 5627  076004  103417                  BCS     7$                      ;ERROR IF NUMBER TOO BIG
 5628  076006  060237  003154  4$:     ADD     R2,P$NUM
 5629  076012  103414                  BCS     7$                      ;ERROR IF NUMBER TOO BIG
 5630  076014  005301                  DEC     R1
 5631  076016  001344                  BNE     3$
 5632  076020  105737  003157          TSTB    P$RADX+1                ;SEE IF NUM WAS PRECEDED BY A - SIGN
 5633  076024  001402                  BEQ     15$                     ; BR IF NO
 5634  076026  005437  003154          NEG     P$NUM                   ; ELSE NEGATE THE NUMBER BEFORE LEAVING
 5635  076032          15$:    POP     R5                              ;RESTORE R5
 5636  076034  004737  075344          JSR     PC,TRVACT               ;SINCE NUMERIC FOUND, GO TAKE ACTION
 5637  076040  000137  075374          JMP     TRVNOB                  ;GO POINT R3 TO NEXT NODE
 5638
 5639  076044          7$:     PRINTF  #CLINBG                         ;PRINT NUMBER TOO BIG ERROR
 5640  076044  012746  052527                                                  MOV     #CLINBG,-(SP)
 5641  076050  012746  000001                                                  MOV     #1,-(SP)
 5642  076054  010600                                                          MOV     SP,R0
 5643  076056  104417                                                          TRAP    C$PNTF
 5644  076060  062706  000004                                                  ADD     #4,SP
 5645  076064  112737  177777  003161          MOVB    #-1,P$GDBD              ;SET ERROR RETURN FLAG
 5646  076072          5$:     POP     R5                              ;RESTORE R5
 5647  076074  000137  075364          JMP     TRVBRC                  ;TAKE "MISS" BRANCH
 5648
 5649
 5650  076100  005001          TRVALP: CLR     R1                      ;CLEAR ALPHA FOUND FLAG
 5651  076102  121427  000101  1$:     CMPB    (R4),#101               ;SEE IF CHAR. LESS THAN A "A"
 5652  076106  002406                  BLT     2$                      ;BR IF YES (NOT ALPHA)
 5653  076110  121427  000132          CMPB    (R4),#132               ;SEE IF CHAR. GREATER THAN A "Z"
 5654  076114  003003                  BGT     2$                      ;BR IF YES (NOT ALPHA)
 5655  076116  005204                  INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR
 5656  076120  005201                  INC     R1                      ;INDICATE AN ALPHA WAS FOUND
 5657  076122  000767                  BR      1$                      ;GO LOOK AT NEXT CHAR.
 5658  076124  005701          2$:     TST     R1                      ;SEE IF ANY ALPHA'S WERE FOUND
 5659  076126  001404                  BEQ     3$                      ;BR IF NO
 5660  076130  004737  075344          JSR     PC,TRVACT               ;IF ANY FOUND TAKE ACTION
 5661  076134  000137  075374          JMP     TRVNOB                  ;THEN UPDATE R3 TO NEXT NODE -NO BRANCH
 5662  076140  000137  075364  3$:     JMP     TRVBRC                  ;NONE FOUND, TAKE MISS BRANCH
 5663
 5664  076144  005001          TRVALN: CLR     R1                      ;CLEAR ALPHANUM FOUND FLAG
 5665  076146  121427  000060  10$:    CMPB    (R4),#60                ;SEE IF CHAR. LESS THAN A "0"
 5666  076152  002417                  BLT     2$                      ;BR IF YES (NOT NUMERIC OR ALPHA)
 5667  076154  121427  000072          CMPB    (R4),#72                ;SEE IF CHAR. GREATER THAN A "9"
 5668  076160  003003                  BGT     1$                      ;BR IF YES (NOT NUMERIC)
 5669  076162  005204                  INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR.
 5670  076164  005201                  INC     R1                      ;INDICATE A NUMERIC FOUND
 5671  076166  000767                  BR      10$                     ;GO LOOK AT NEXT CHAR.
 5672  076170  121427  000101  1$:     CMPB    (R4),#101               ;SEE IF CHAR. LESS THAN A "A"
```

```
5673  076174  002406                BLT     2$                      ;BR IF YES (NOT ALPHA)
5674  076176  121427  000132        CMPB    (R4),#132               ;SEE IF CHAR. GREATER THAN A "Z"
5675  076202  003003                BGT     2$                      ;BR IF YES (NOT ALPHA)
5676  076204  005204                INC     R4                      ;UPDATE CMD LINE PTR TO NEXT CHAR
5677  076206  005201                INC     R1                      ;INDICATE AN ALPHA FOUND
5678  076210  000756                BR      10$                     ;GO LOOK AT NEXT CHAR.
5679  076212  005701        2$:     TST     R1                      ;SEE IF ANY ALPHANUM'S WERE FOUND
5680  076214  001404                BEQ     3$                      ;BR IF NO
5681  076216  004737  075344        JSR     PC,TRVACT               ;IF ANY FOUND TAKE ACTION
5682  076222  000137  075374        JMP     TRVNOB                  ;THEN UPDATE R3 TO NEXT NODE -NO BRANCH
5683  076226  000137  075364  3$:   JMP     TRVBRC                  ;NONE FOUND, TAKE MISS BRANCH
5684
5685
5686
5687  076232                TRVSTR: PUSH    R5                      ;SAVE R5
5688  076234  010401                MOV     R4,R1                   ;POINT R1 TO CMD STRING
5689  076236  010305                MOV     R3,R5
5690  076240  062705  000006        ADD     #6,R5                   ;POUNT R5 TO MATCH STRING FROM CLI NODE
5691  076244  005037  003152        CLR     P$CNT                   ;CLEAR CHAR MATCH COUNT
5692  076250  105715        2$:     TSTB    (R5)                    ;SEE IF END OF MATCH STRING YET
5693  076252  001411                BEQ     10$                     ;BR IF YES
5694  076254  105711                TSTB    (R1)                    ;SEE IF END OF CMD LINE YET
5695  076256  001407                BEQ     10$                     ;BR IF YES
5696  076260  121115                CMPB    (R1),(R5)               ;SEE IF CHARACTERS MATCH
5697  076262  001005                BNE     10$                     ;BR IF NO
5698  076264  005237  003152        INC     P$CNT                   ;MATCH -INCREMENT MATCH COUNT
5699  076270  005201                INC     R1                      ;UPDATE STRING POINTERS
5700  076272  005205                INC     R5
5701  076274  000765                BR      2$                      ;BR TO CONTINUE CHECKING CHARS.
5702
5703  076276  005737  003152  10$:   TST     P$CNT                   ;WHEN DONE SEE IF ANY MATCHES FOUND
5704  076302  001407                BEQ     15$                     ;BR IF NO, GO TAKE THE MISS BRANCH
5705  076304  010104                MOV     R1,R4                   ;POINT CMD POINTER TO END OF STRING &
5706  076306                        POP     R5                      ;RESTORE R5
5707  076310  004737  075344        JSR     PC,TRVACT               ;IF A MATCH FOUND, GO DO MATCH ACTION
5708  076314  066303  000004        ADD     4(R3),R3                ;UPDATE R3 TO NEXT NODE (NO BRANCH)
5709  076320  000207                RTS     PC                      ; (NO RETURN THRU TRVNOB SINCE DIFFERENT
5710                                                                ;  DISPLACEMENT DUE TO MATCH STRING)
5711  076322                15$:    POP     R5                      ;RESTORE R5
5712  076324  000137  075364        JMP     TRVBRC                  ; GO TAKE BRANCH
5713                                                                ; (PARSED OK), -1 IF ILL CMD.....
5714
5715                                ;--+
5716                                ;   TRVADR                      TRAVERSE COMMAND LINE INPUT ADDRESS
5717                                ;
5718                                ;   THIS ROUTINE IS CALLED BY TWO DIFFERENT ACTION ROUTINES.  THE
5719                                ;   NODE ACTION ROUTINE CALLS IT TO PARSE THROUGH THE NODE
5720                                ;   ADDRESS INPUT BY THE OPERATOR.  THE OPRSEL ACTION ROUTINE
5721                                ;   CALLS TRVADR TO PARSE THROUGH THE "OPERATOR SELECTED" MESSAGE
5722                                ;   WHICH HAS BEEN INPUT IN THE COMMAND LINE.  FOR A NODE ADDRESS,
5723                                ;   THE ROUTINE LOOKS FOR A '/' AS A DELIMETER FOR THE ADDRESS,
5724                                ;   AND REPLACES THE / WITH A NULL BYTE FOR USE BY THE ADDRESS
5725                                ;   PACKING ROUTINE.  WHEN CALLED BY THE OPRSEL ROUTINE, A ":"
5726                                ;   IS EXPECTED AS THE DELIMETER FOR THE OPERATOR SELECTED MESSAGE.
5727                                ;   IF A NULL STRING IS ENTERED, AN ERROR MESSAGE IS PRINTED.
5728                                ;
```

```
5729                    ;       INPUTS -                R4 - POINTS TO THE BEGINING OF THE ADDRESS
5730                    ;                                    OR MESSAGE IN THE COMMAND LINE
5731                    ;       OUTPUTS -               SUMMARIZED IN TABLE BELOW
5732                    ;
5733                    ;       COMMAND LINE                        OUTPUTS
5734                    ;       INPUT CONDITION ! P$GDBD ! R4 POINTS TO  !  CFLAG CONTAINS  ! P$MERR
5735                    ;       ----------------+--------+--------------+-----------------+-------
5736                    ;       ILLEGAL CHAR.   !  -1    ! ILL. CHAR.   !                 ! N/A
5737                    ;       ADR./ASSIST     !   0    ! END OF LINE  !     CASIST      ! N/A
5738                    ;       ADR./TARGET     !        !              !                 !
5739                    ;       ADR./           !   0    ! END OF LINE  !     CTARGT      ! N/A
5740                    ;       ADR.            !        !              !                 !
5741                    ;       ADR./CHAR. OR   !        !              !                 !
5742                    ;       "OPR SEL/CHAR.  !        !              !                 !
5743                    ;         OTHER THAN "A"!  -1    ! /            !     CTARGT      ! N/A
5744                    ;         "T" OR BLANK  !        !              !                 !
5745                    ;       "."             !   0    ! CHAR. AFTER "!                 ! -1
5746                    ;       "OPR SEL"       !   0    ! CHAR. AFTER "!     OPRSEL      !  0
5747                    ;
5748                    ;       CALLING PROCEDURE -  JSR  PC,TRVADR
5749                    ;       REGISTER USAGE -     R1 IS USED AS A COUNTER TO REPORT ERROR MESSAGES
5750                    ;                            IF NULL STRINGS ARE ENTERED.
5751                    ;                            R4 POINTS TO THE NEXT CHAR. IN THE COMMAND LINE
5752                    ;
5753                    ;--+
5754
5755  076330  005001    TRVADR: CLR     R1                      ;CLEAR HEX DIGIT FOUND FLAG
5756  076332  121427  000000  1$:     CMPB    (R4),#0         ;SEE IF NUL CHAR.
5757  076336  001423            BEQ     20$                     ; IF YES, RETURN
5758  076340  121427  000040          CMPB    (R4),#40        ;SEE IF ILLEGAL CHARACTER
5759  076344  002414            BLT     10$                     ;IF YES; BRANCH TO ERROR ROUTINE
5760  076346  121427  000042          CMPB    (R4),#42        ;SEE IF CHAR. IS A '"'
5761  076352  001454            BEQ     40$                     :IF YES ,BRANCH TO 40$
5762  076354  121427  000057          CMPB    (R4),#57          SE IF CHAR. IS A "/"
5763  076360  001420            BEQ     30$                     ;BRANCH IF YES
5764  076362  121427  000132          CMPB    (R4),#132       ;SEE IF CHAR. GREATER THAN "Z"
5765  076366  003003            BGT     10$                     ; IF YES, ILLEGAL CHAR.
5766  076370  005204            INC     R4                      ;UPDATE CMD LINE POINTER TO NEXT CHAR.
5767  076372  005201            INC     R1                      ;INCIDATE A VALID CHAR. FOUND
5768  076374  000756            BR      1$                      ;LOOK AT NEXT CHAR.
5769  076376  112737  177777  003161  10$:    MOVB    #-1,P$GDBD      ;SET ERROR FLAG
5770  076404  000464            BR      50$                     ;RETURN
5771  076406  005701            20$:    TST     R1              ;SEE IF VALID CHARACTERS FOUND
5772  076410  001772            BEQ     10$                     ; IF NO, ILLEGAL CHAR.
5773  076412  012737  000000  003672  25$:    MOV     #CTARGT,CFLAG   ;SET TARGET FLAG
5774  076420  000456            BR      50$                     ;RETURN
5775  076422  005701            30$:    TST     R1              ;SEE IF VALID CHARACTERS FOUND
5776  076424  001764            BEQ     10$                     ; IF NO, ILLEGAL CHAR.
5777  076426  112714  000000          MOVB    #0,(R4)         ; IF YES, REPLACE "/" WITH NULL CHAR.
5778  076432  005204            INC     R4                      ;UPDATE CMD. LINE POINTER TO NEXT CHAR.
5779  076434  121427  000000          CMPB    (R4),#0         ;IS NEXT CHAR. NULL
5780  076440  001764            BEQ     25$                     ;  IF YES, TAKE DEFAULT OF TARGET
5781  076442  121427  000101          CMPB    (R4),#'A        ;IS NEXT CHAR. "A"
5782  076446  001412            BEQ     35$                     ;  IF YES, BR 35$
5783  076450  121427  000124          CMPB    (R4),#'T        ;IS NEXT CHAR. "T"
5784  076454  001756            BEQ     25$                     ;  IF YES, SET TARGET FLAG
```

```
5785  076456  112737  177777  003161          MOVB    #-1,P$GDBD              ;  ELSE, SET ERROR FLAG,
5786  076464  005304                          DEC     R4                      ;  READJUST COMMAND LINE POINTER
5787  076466  112714  000057                  MOVB    #'/,(R4)                ;  AND REPLACE / IN CMD LINE TO FIX ERROR
5788  076472  000747                          BR      25$                     ;  SET TARGET FLAG AND RETURN
5789  076474  012737  000001  003672  35$:    MOV     #CASIST,CFLAG           ;SET ASSIST FLAG
5790  076502  000425                          BR      50$
5791  076504  005701                  40$:    TST     R1                      ;SEE IF ANY CHARACTERS TYPED
5792  076506  001407                          BEQ     45$                     ;IF NO, BRANCH TO 45$
5793  076510  112714  000000                  MOVB    #0,(R4)                 ;ELSE, REPLACE "" WITH NULL
5794  076514  012737  000006  003672          MOV     #OPRSEL,CFLAG           ;SET OPERATOR SELECTED FLAG
5795  076522  005204                          INC     R4
5796  076524  000414                          BR      50$                     ;RETURN
5797  076526                          45$:    PRINTF  #NULSTR                 ;PRINT NULL STRING ERROR MESSAGE
5798  076526  012746  053362                                                          MOV     #NULSTR,-(SP)
5799  076532  012746  000001                                                          MOV     #1,-(SP)
5800  076536  010600                                                                 MOV     SP,R0
5801  076540  104417                                                                 TRAP    C$PNTF
5802  076542  062706  000004                                                          ADD     #4,SP
5803  076546  112737  177777  003163          MOVB    #-1,P$MERR              ;SET OPER. SELECTED MSG. ERROR FLAG
5804  076554  005204                          INC     R4                      ;MOVE CMD. LINE POINTER TO NEXT CHAR.
5805  076556  000207                  50$:    RTS     PC                      ;RETURN
5806
5807                                  ;-------------------------------------------------------------------------------
5808
5809                                  .SBTTL  REPORT CODING SECTION
5810
5811
5812                                  ;++
5813                                  ; THE REPORT CODING SECTION CONTAINS THE
5814                                  ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
5815                                  ;--
5816
5817  076560                                  BGNRPT
5818  076560                                                                  L$RPT::
5819
5820                                  ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5821                                  ;       THIS SECTION, WHICH IS OPTIONAL, CONTAINS THE CODE FOR PRINTING
5822                                  ;       STATISTICAL INFORMATION GATHERED BY THE DIAGNOSTIC.  IT IS
5823                                  ;       EXECUTED BY THE OPERATOR COMMAND "PRINT" OR BY THE MACRO CALL
5824                                  ;       "DORPT".  USE THE PRINTS MACRO TO PRINT THE INFORMATION.
5825                                  ;       USE FORMAT STATEMENTS AS IN THE PRINTB/PRINTX MACROS.  IT IS
5826                                  ;       THE PROGRAMMER'S RESPONSIBILTY TO DEVISE AND IMPLEMENT THE
5827                                  ;       FORM AND CONTENT OF THE STATISTICS.
5828                                  ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5829
5830  076560  004737  101710                  JSR     PC,ACTSUM
5831  076564                                  EXIT    RPT
5832  076564  000167                                                          .WORD   J$JMP
5833  076566  000000                                                          .WORD   L10006-2-.
5834
5835                                  ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5836                                  ;     INSERT LOCAL STORAGE THAT IS USED ONLY
5837                                  ;     DURING THE REPORT SECTION.
5838                                  ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5839
5840                                  ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
 5841                                 ;      INSERT MESSAGES THAT ARE USED ONLY
 5842                                 ;      DURING THE REPORT SECTION.
 5843                                 ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 5844
 5845                                         .EVEN
 5846
 5847  076570                                 ENDRPT
 5848  076570                                                                  L10006:
 5849  076570  104425                                                                  TRAP    C$RPT
```

```
                                .SBTTL  PROTECTION TABLE

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

076572                                  BGNPROT
076572                                                                                  L$PROT::

076572  177777                          -1                      ;OFFSET INTO P-TABLE FOR CSR ADDRESS
076574  177777                          -1                      ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
076576  177777                          -1                      ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

076600                                  ENDPROT

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;       INSERT BYTE OFFSET FOR DATA NOTED IN COMMENTS ABOVE.  (OFFSET
                                ;       REFERS TO THE NUMBER OF BYTES FROM THE BEGINNING OF A PTABLE
                                ;       ENTRY TO THE ITEM IN QUESTION.)  IF THE PARTICULAR
                                ;       ITEM DOES NOT APPLY, LEAVE ENTRY AS -1.  WHEN THE RUNTIME
                                ;       SERVICES EXECUTES A GPHARD, IT USES THESE OFFSETS (IF NOT
                                ;       SET TO -1) TO GET THE ITEMS AND COMPARE WITH THOSE SAVED
                                ;       IN THE XXDP+ MONITOR.  IF THE UNIT BEING REQUESTED MATCHES THE
                                ;       LOAD DEVICE, THE RUNTIME SERVICES RETURN AN INCOMPLETE FLAG ON
                                ;       THE GPHARD.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
                                        .SBTTL  INITIALIZE SECTION

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

076600                                          BGNINIT
076600                                                                                          L$INIT::

                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                        ;       THE INITIALIZE CODE IS EXECUTED UNDER FIVE CONDITIONS.  THERE
                                        ;       ARE SUPERVISOR EVENT FLAGS THAT ARE USED TO LET THE
                                        ;       DIAGNOSTIC KNOW UNDER WHICH CONDITION THE EXECUTION IS TAKING
                                        ;       PLACE.  THE EVENT FLAGS ARE READ USING THE ''READEF'' MACRO.
                                        ;       THE CONDITIONS UNDER WHICH THE INIT CODE IS EXECUTED AND THE
                                        ;       CORRRESPONDING EVENT FLAGS ARE:
                                        ;               START COMMAND           EF.START
                                        ;               RESTART COMMAND         EF.RESTART
                                        ;               CONTINUE COMMAND        EF.CONTINUE
                                        ;               POWERDOWN/POWERUP       EF.PWR
                                        ;               NEW PASS                EF.NEW
                                        ;       EXAMPLE OF EVENT FLAG USE:
                                        ;               READEF  #EF.START
                                        ;               BCOMPLETE       STARTCODE
                                        ;       DURING THE INIT CODE, USE THE ''GPHARD'' MACRO TO OBTAIN P-TABLE
                                        ;       INFORMATION FOR DEVICE TESTING.  GET ONE UNIT'S INFORMATION IF
                                        ;       THIS IS A SEQUENTIAL DIAGNOSTIC.  GET INFORMATION ON ALL
                                        ;       UNITS AVAILABLE FOR TESTING IF THIS IS AN EXERCISER.  THE NUMBER
                                        ;       OF UNITS AVAILABLE IS IN A HEADER LOCATION: 'L$UNIT''.
                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

076600  022737  000020  003672  INIT:   CMP     #CEXIT,CFLAG            ;SEE IF EXIT COMMAND TYPED
076606  001004                          BNE     INIT1                   ; IF NO, DO INIT CODE
076610  005037  003672                  CLR     CFLAG                   ; ELSE, CLEAR EXIT FLAG
076614  000137  100130                  JMP     INICLN                  ; AND DO CLEANUP
076620                          INIT1:  READEF  #EF.START               ;IF HERE BECAUSE OF ''START''. DO INIT
076620  012700  000040                                                                  MOV     #EF.START,R0
076624  104447                                                                          TRAP    C$REFG
076626                                  BCOMPLETE       START
076626  103424                                                                          BCS     START
076630                                  READEF  #EF.RESTART             ;IF HERE BECAUSE OF ''RESTART'', DO SOME INIT
076630  012700  000037                                                                  MOV     #EF.RESTART,R0
076634  104447                                                                          TRAP    C$REFG
076636                                  BNCOMPLETE      5$
076636  103002                                                                          BCC     5$
076640  000137  100046                  JMP     RESTRT
076644                          5$:     READEF  #EF.CONTINUE            ;IF HERE BECAUSE OF ''CONTINUE'', EXIT
076644  012700  000036                                                                  MOV     #EF.CONTINUE,R0
076650  104447                                                                          TRAP    C$REFG
076652                                  BNCOMPLETE      10$
076652  103002                                                                          BCC     10$
076654  000137  100046                  JMP     RESTRT
076660                          10$:    READEF  #EF.NEW                 ;IF HERE ON NEW PASS, SKIP SOME INIT
076660  012700  000035                                                                  MOV     #EF.NEW,R0
076664  104447                                                                          TRAP    C$REFG
```

```
5933  076666                                  BNCOMPLETE      15$
5934  076666  103002                                                          BCC     15$
5935  076670  000137  100112                  JMP     NEW
5936  076674  000137  100132          15$:    JMP     INIEXI                  ;IF DON'T KNOW WHY WE'RE HERE, EXIT
5937  076700                          START:  I$STACK #1000,SP                ;SET PARAMETER STACK POINTER
5938  076706                                  MEMORY  FRESIZ                  ;GET FREE MEMORY INFO
5939  076706  104431                                                          TRAP    C$MEM
5940  076710  010037  047660                                                  MOV     R0,FRESIZ
5941  076714  013737  047660  047662          MOV     FRESIZ,FREMEM           ;SIZE OF FREE MEMORY IN FRESIZ
5942  076722  062737  000002  047662          ADD     #2,FREMEM               ;START OF FREE MEMORY IN FREMEM
5943  076730  012702  003674                  MOV     #CLKCSR,R2              ;SETUP R2 AS A PRT. TO CLOCK INFO. BLOCK
5944  076734                                  CLOCK   L,R1                    ;GET LINE CLOCK INFO
5945  076734  012700  000114                                                  MOV     #'L,R0
5946  076740  104462                                                          TRAP    C$CLCK
5947  076742  010001                                                          MOV     R0,R1
5948  076744                                  BNCOMPLETE      1$              ;IF NONE, SEE IF P CLOCK PRESENT
5949  076744  103006                                                          BCC     1$
5950  076746  004737  067656                  JSR     PC,CLKSET               ;SET UP CLOCK INFO TABLE AND VECTOR
5951  076752  012737  000100  003704          MOV     #LCLKEN,CLKEN           ;SET UP THE ENABLE LINE CLOCK DATA
5952  076760  000436                          BR      3$
5953  076762                          1$:     CLOCK   P,R1                    ;GET P CLOCK INFO
5954  076762  012700  000120                                                  MOV     #'P,R0
5955  076766  104462                                                          TRAP    C$CLCK
5956  076770  010001                                                          MOV     R0,R1
5957  076772                                  BNCOMPLETE      2$              ;IF NO CLOCK, ERROR
5958  076772  103017                                                          BCC     2$
5959  076774  004737  067656                  JSR     PC,CLKSET               ; ELSE SET UP CLOCK INFO AND VECTOR
5960  077000  062737  000002  003674          ADD     #2,CLKCSR               ; POINT CLKCSR TO P-CLK COUNT SET REG.
5961  077006  012777  001600  104660          MOV     #PCLKCT,@CLKCSR         ;LOAD CLK SET REG. WITH COUNT VALUE
5962  077014  162737  000002  003674          SUB     #2,CLKCSR               ;POINT CLKCSR BACK TO P-CLK CSR
5963  077022  012737  000111  003704          MOV     #PCLKEN,CLKEN           ;SETUP TO ENABLE P-CLK DATA
5964  077030  000412                          BR      3$
5965  077032                          2$:     PRINTF  #NOCLK                  ;ERROR MESSAGE
5966  077032  012746  062622                                                  MOV     #NOCLK,-(SP)
5967  077036  012746  000001                                                  MOV     #1,-(SP)
5968  077042  010600                                                          MOV     SP,R0
5969  077044  104417                                                          TRAP    C$PNTF
5970  077046  062706  000004                                                  ADD     #4,SP
5971  077052  000137  100130                  JMP     INICLN                  ;CANNOT CONTINUE, DO CLEANUP
5972  077056                          3$:     GPHARD  #0,R1                   ;GET P-TAB POINTER FOR THIS UNIT
5973  077056  012700  000000                                                  MOV     #0,R0
5974  077062  104442                                                          TRAP    C$GPHRD
5975  077064  010001                                                          MOV     R0,R1
5976  077066                                  BCOMPLETE       4$              ;THIS ONE IS NOT AVAILABLE
5977  077066  103402                                                          BCS     4$
5978  077070  000137  100130                  JMP     INICLN
5979  077074  012137  047652          4$:     MOV     (R1)+,UNACSR            ;SAVE CSR
5980  077100  012137  047654                  MOV     (R1)+,UNAVEC            ;SAVE VECTOR
5981  077104  012137  047656                  MOV     (R1)+,UNAPRI            ;SAVE PRIORITY
5982  077110                                  SETVEC  UNAVEC,#UNAISR,UNAPRI   ;SETUP UNA INTERRUPT VECTOR
5983  077110  013746  047656                                                  MOV     UNAPRI,-(SP)
5984  077114  012746  071144                                                  MOV     #UNAISR,-(SP)
5985  077120  013746  047654                                                  MOV     UNAVEC,-(SP)
5986  077124  012746  000003                                                  MOV     #3,-(SP)
5987  077130  104437                                                          TRAP    C$SVEC
5988  077132  062706  000010                                                  ADD     #10,SP
```

```
 5989  077136  013737  047652  047632          MOV     UNACSR,PCSR0            ;PCSR0
 5990  077144  013737  047632  047634          MOV     PCSR0,PCSR1
 5991  077152  062737  000002  047634          ADD     #2,PCSR1                ;PCSR1
 5992  077160  013737  047634  047636          MOV     PCSR1,PCSR2
 5993  077166  062737  000002  047636          ADD     #2,PCSR2                ;PCSR2
 5994  077174  013737  047636  047640          MOV     PCSR2,PCSR3
 5995  077202  062737  000002  047640          ADD     #2,PCSR3                ;PCSR3
 5996  077210  012703  000050                  MOV     #TBLLEN,R3              ;CLEAR NODE TABLE
 5997  077214  012702  002404                  MOV     #NODTBL,R2
 5998  077220  005022                  5$:     CLR     (R2)+
 5999  077222  005303                          DEC     R3
 6000  077224  001375                          BNE     5$
 6001  077226  012703  000132                  MOV     #STBLEN,R3              ;CLEAR SUMMARY TABLE
 6002  077232  012702  002656                  MOV     #STATBL,R2
 6003  077236  005022                  6$:     CLR     (R2)+
 6004  077240  005303                          DEC     R3
 6005  077242  001375                          BNE     6$
 6006  077244  005037  050500                  CLR     S.NREC                  ; CLEAR SUMMARY DATA COUNTERS
 6007  077250  005037  050476                  CLR     S.REC
 6008  077254  005037  050502                  CLR     S.LEN
 6009  077260  005037  050504                  CLR     S.COMP
 6010  077264  005037  050506                  CLR     S.BYTE
 6011  077270  005037  050510                  CLR     S.XFER
 6012  077274  005037  003706                  CLR     TIMMIN                  ;CLEAR TIME SINCE-START-LOCATIONS
 6013  077300  005037  003710                  CLR     TIMSEC
 6014  077304  013737  003702  003712          MOV     CLKHZ,TIMTCK            ;LOAD TICKS/SEC
 6015  077312                                  SETVEC  CLKVEC,#CLKINT,CLKBR    ;SETUP CLOCK INTERRUPT VECTOR
 6016  077312  013746  003676                                                          MOV     CLKBR,-(SP)
 6017  077316  012746  067702                                                          MOV     #CLKINT,-(SP)
 6018  077322  013746  003700                                                          MOV     CLKVEC,-(SP)
 6019  077326  012746  000003                                                          MOV     #3,-(SP)
 6020  077332  104437                                                                  TRAP    C$SVEC
 6021  077334  062706  000010                                                          ADD     #10,SP
 6022  077340  013777  003704  104326          MOV     CLKEN,@CLKCSR           ;SET ENABLE BITS IN THE CLOCK TO START
 6023  077346                                  SETPRI  #PRI00                  ;SET PRIORITY=0 SO CLOCK CAN INTERRUPT
 6024  077346  012700  000000                                                          MOV     #PRI00,R0
 6025  077352  104441                                                                  TRAP    C$SPRI
 6026  077354                                  CALL    UNAINI                  ;INITIALIZE THE UNA
 6027  077362                                  CALL    FUNCT  #RDDEFA          ;READ UNA DEFAULT PHYSICAL ADDRESS
 6028  077374                                  P$POP   R2                      ;CHECK FOR ERROR
 6029  077376  001402                          BEQ     8$
 6030  077400  000137  100034                  JMP     20$
 6031  077404                          8$:     CALL    BINHEX #PCBB2,#6,#STRBUF        ;PUT ADDRESS INTO HEX FORMAT
 6032  077426                                  PRINTS  #HDMSG1,#STRBUF         ;PRINT ADDRESS
 6033  077426  012746  002322                                                          MOV     #STRBUF,-(SP)
 6034  077432  012746  055342                                                          MOV     #HDMSG1,-(SP)
 6035  077436  012746  000002                                                          MOV     #2,-(SP)
 6036  077442  010600                                                                  MOV     SP,R0
 6037  077444  104416                                                                  TRAP    C$PNTS
 6038  077446  062706  000006                                                          ADD     #6,SP
 6039  077452                                  CALL    FUNCT  #RDSTA           ;READ STATUS TO GET ROM VERSION
 6040  077464                                  P$POP   R2                      ;CHECK FOR ERROR
 6041  077466  001162                          BNE     20$
 6042  077470  113702  047670                  MOVB    PCBB2,R2                ;ONLY WANT LOWEST 6 BITS
 6043  077474  142702  000300                  BICB    #300,R2
 6044  077500                                  PRINTS  #HDMSG2,R2              ;PRINT ROM VERSION
```

```
 6045  077500  010246                                                          MOV     R2,-(SP)
 6046  077502  012746  055413                                                  MOV     #HDMSG2,-(SP)
 6047  077506  012746  000002                                                  MOV     #2,-(SP)
 6048  077512  010600                                                          MOV     SP,R0
 6049  077514  104416                                                          TRAP    C$PNTS
 6050  077516  062706  000006                                                  ADD     #6,SP
 6051  077522                      PRINTS  #HDMSG3             ;PRINT MORE HEADER INFO
 6052  077522  012746  055466                                                  MOV     #HDMSG3,-(SP)
 6053  077526  012746  000001                                                  MOV     #1,-(SP)
 6054  077532  010600                                                          MOV     SP,R0
 6055  077534  104416                                                          TRAP    C$PNTS
 6056  077536  062706  000004                                                  ADD     #4,SP
 6057  077542  012703  050340          MOV     #UCB20,R3          ;SET UP FUNCTION CONTROL BLOCK
 6058  077546  012723  000002          MOV     #2,(R3)+           ; MOVE 2 BYTES...
 6059  077552  012723  050550          MOV     #TEMP,(R3)+        ; INTO LOCATION TEMP...
 6060  077556  012723  000000          MOV     #0,(R3)+
 6061  077562  012713  021040          MOV     #21040,(R3)        ; FROM LOCATION 21040
 6062  077566                          CALL    FUNCT  #DMPMEM     ;DUMP INTERNAL MEMORY
 6063  077600                          P$POP   R2                 ;CHECK FOR ERROR
 6064  077602  001114                  BNE     20$
 6065  077604  013703  050550          MOV     TEMP,R3            ;PUT RESULT INTO R3
 6066  077610  032703  002000          BIT     #BIT10,R3          ;DETERMINE STATUS
 6067  077614  001414                  BEQ     11$
 6068  077616  032703  004000          BIT     #BIT11,R3
 6069  077622  001425                  BEQ     12$
 6070  077624                          PRINTS  #HDMSG4            ; BIT10!BIT11 = REMOTE AND POWER UP BOOT ENABLED
 6071  077624  012746  055522                                                  MOV     #HDMSG4,-(SP)
 6072  077630  012746  000001                                                  MOV     #1,-(SP)
 6073  077634  010600                                                          MOV     SP,R0
 6074  077636  104416                                                          TRAP    C$PNTS
 6075  077640  062706  000004                                                  ADD     #4,SP
 6076  077644  000435                  BR      15$
 6077  077646  032703  004000  11$:    BIT     #BIT11,R3
 6078  077652  001422                  BEQ     13$
 6079  077654                          PRINTS  #HDMSG5            ; BIT11 = REMOTE BOOT ENABLED WITH ROM
 6080  077654  012746  055577                                                  MOV     #HDMSG5,-(SP)
 6081  077660  012746  000001                                                  MOV     #1,-(SP)
 6082  077664  010600                                                          MOV     SP,R0
 6083  077666  104416                                                          TRAP    C$PNTS
 6084  077670  062706  000004                                                  ADD     #4,SP
 6085  077674  000421                  BR      15$
 6086  077676                  12$:    PRINTS  #HDMSG6            ; BIT10 = REMOTE BOOT ENABLED
 6087  077676  012746  055650                                                  MOV     #HDMSG6,-(SP)
 6088  077702  012746  000001                                                  MOV     #1,-(SP)
 6089  077706  010600                                                          MOV     SP,R0
 6090  077710  104416                                                          TRAP    C$PNTS
 6091  077712  062706  000004                                                  ADD     #4,SP
 6092  077716  000410                  BR      15$
 6093  077720                  13$:    PRINTS  #HDMSG7            ; REMOTE BOOT NOT ENABLED
 6094  077720  012746  055710                                                  MOV     #HDMSG7,-(SP)
 6095  077724  012746  000001                                                  MOV     #1,-(SP)
 6096  077730  010600                                                          MOV     SP,R0
 6097  077732  104416                                                          TRAP    C$PNTS
 6098  077734  062706  000004                                                  ADD     #4,SP
 6099  077740  032703  010000  15$:    BIT     #BIT12,R3
 6100  077744  001411                  BEQ     16$
```

```
077746                                          PRINTS  #HDMSG8                 ; BIT12 = SELF TEST ENABLED
077746  012746  055751                                                                                  MOV     #HDMSG8,-(SP)
077752  012746  000001                                                                                  MOV     #1,-(SP)
077756  010600                                                                                          MOV     SP,R0
077760  104416                                                                                          TRAP    C$PNTS
077762  062706  000004                                                                                  ADD     #4,SP
077766  000410                                  BR      17$
077770                                  16$:    PRINTS  #HDMSG9                 ; SELF TEST DISABLED
077770  012746  056014                                                                                  MOV     #HDMSG9,-(SP)
077774  012746  000001                                                                                  MOV     #1,-(SP)
100000  010600                                                                                          MOV     SP,R0
100002  104416                                                                                          TRAP    C$PNTS
100004  062706  000004                                                                                  ADD     #4,SP
100010  012737  000000  002370  17$:    MOV     #ALPHA,P$TYPE           ;SET MESSAGE DEFAULT VALUES
100016  012737  001000  002372          MOV     #512.,P$SIZE
100024  012737  000001  002374          MOV     #1,P$CPYS
100032  000405                          BR      RESTRT
100034                          20$:    ERRDF   18,EMSG18,ERR1
100034  104455                                                                                          TRAP    C$ERDF
100036  000022                                                                                          .WORD   18
100040  065127                                                                                          .WORD   EMSG18
100042  067460                                                                                          .WORD   ERR1
100044  000431                          BR      INICLN
100046  005037  003706          RESTRT: CLR     TIMMIN                  ;CLEAR TIME SINCE-START-LOCATIONS
100052  005037  003710                  CLR     TIMSEC
100056  013737  003702  003712          MOV     CLKHZ,TIMTCK            ;LOAD TICKS/SEC
100064                                  SETVEC  CLKVEC,#CLKINT,CLKBR    ;SETUP CLOCK INTERRUPT VECTOR
100064  013746  003676                                                                                  MOV     CLKBR,-(SP)
100070  012746  067702                                                                                  MOV     #CLKINT,-(SP)
100074  013746  003700                                                                                  MOV     CLKVEC,-(SP)
100100  012746  000003                                                                                  MOV     #3,-(SP)
100104  104437                                                                                          TRAP    C$SVEC
100106  062706  000010                                                                                  ADD     #10,SP
100112  013777  003704  103554  NEW:    MOV     CLKEN,@CLKCSR           ;SET ENABLE BITS IN THE CLOCK TO START
100120                                  SETPRI  #PRI00                  ;SET PRIORITY=0 SO CLOCK CAN INTERRUPT
100120  012700  000000                                                                                  MOV     #PRI00,R0
100124  104441                                                                                          TRAP    C$SPRI
100126  000401                          BR      INIEXI                  ;EXIT
100130                          INICLN: DOCLN                           ;ABORT PASS
100130  104444                                                                                          TRAP    C$DCLN
100132                          INIEXI: EXIT    INIT                    ;EXIT INIT SECTION
100132  104432                                                                                          TRAP    C$EXIT
100134  000002                                                                                          .WORD   L10010-.

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;       INSERT LOCAL STORAGE THAT IS USED ONLY
                                ;       DURING THE INITIALIZE SECTION.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;       INSERT MESSAGES THAT ARE USED ONLY
                                ;       DURING THE INITIALIZE SECTION.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                                        .EVEN
```

```
 6157  100136                          ENDINIT
 6158  100136
 6159  100136  104411                                  L10010:  TRAP    C$INIT
```

```
                                .SBTTL  AUTODROP SECTION

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

100140                                  BGNAUTO
100140                                                                      L$AUTO::

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;       INSERT CODE HERE TO CHECK DEVICE(S) TO SEE IF THEY RESPOND.
                                ;       ISSUE A ''DODU'' FOR THOSE THAT DON'T.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

100140                                  ENDAUTO
100140                                                                      L10011:
100140  104461                                                                      TRAP    C$AUTO
```

```
                                        .SBTTL  CLEANUP CODING SECTION

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

100142                                          BGNCLN
100142                                                                                          L$CLEAN::

                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                        ;       INSERT YOUR CLEANUP CODING.  THIS CODING SHOULD
                                        ;       RESTORE YOUR TEST-DEVICE TO A NEUTRAL STATE.
                                        ;       THIS CODE WILL BE EXECUTED AFTER EACH PASS AND AFTER THE
                                        ;       PROGRAM IS INTERRUPTED BY ''^C''.
                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

100142  012737  000000  050044                  MOV     #0,$WDMC+4              ;CLEAR MULTICAST ADDRESS LIST
100150                                          CALL    FUNCT   #WDMULA         ; WRITE 0 INTO LIST LENGTH
100162  012737  000400  050044                  MOV     #400,$WDMC+4            ; RESET FOR 1 ENTRY
100170                                          P$POP   R2                      ;CHECK FOR ERROR
100172  001404                                  BEQ     25$                     ; IF OK CONTINUE
100174                                          ERRHRD  38,EMSG25               ; ELSE, REPORT ERROR
100174  104456                                                                                  TRAP    C$ERHRD
100176  000046                                                                                  .WORD   38
100200  065442                                                                                  .WORD   EMSG25
100202  000000                                                                                  .WORD   0
100204  005737  050516                  25$:    TST     NIRCNT                  ;SEE IF RECEVE COUNT ZERO
100210  001413                                  BEQ     30$                     ; IF YES, EXIT
100212                                          CALL    RECEVE                  ; ELSE, CLEAR OUT UNWANTED ENTRIES
100220                                          P$POP   R2
100222  001770                                  BEQ     25$                     ; IF ANY WERE FOUND, UPDATE NEXT POINTER
100224                                          CALL    GETRNX  #RRGNXT
100236  000762                                  BR      25$                     ;SEE IF ANY MORE ENTRIES
100240  005077  103430                  30$:    CLR     @CLKCSR                 ;DISABLE CLOCK
100244                                          SETPRI  #PRI07                  ;SET PROCESSOR PRIORITY BACK TO 7
100244  012700  000340                                                                          MOV     #PRI07,R0
100250  104441                                                                                  TRAP    C$SPRI
100252                                          EXIT    CLN
100252  104432                                                                                  TRAP    C$EXIT
100254  000002                                                                                  .WORD   L10012-.

                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                        ;     INSERT LOCAL STORAGE THAT IS USED ONLY
                                        ;     DURING THE CLEANUP SECTION.
                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                        ;     INSERT MESSAGES THAT ARE USED ONLY
                                        ;     DURING THE CLEANUP SECTION.
                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                                                .EVEN

100256                                          ENDCLN
100256                                                                                          L10012:
```

```
  6236  100256  104412                                                    TRAP    C$CLEAN
```

```
                                .SBTTL  DROP UNIT SECTION

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

100260                                  BGNDU
100260                                                                      L$DU::

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;       INSERT DROP CODE HERE.  THIS CODE WILL BE EXECUTED AFTER
                                ;       A 'DROP'' COMMAND OR A 'DODU'' MACRO EXECUTION.  THE PURPOSE
                                ;       OF THIS CODE IS TO DO ANY NECESSARY HOUSEKEEPING AFTER A
                                ;       UNIT HAS BEEN DROPPED.  THIS SECTION IS OPTIONAL.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

100260                                  EXIT    DU
100260  000167                                                                      .WORD   J$JMP
100262  000000                                                                      .WORD   L10013-2-.

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;     INSERT LOCAL STORAGE THAT IS USED ONLY
                                ;     DURING THE DROP-UNIT SECTION.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;     INSERT MESSAGES THAT ARE USED ONLY
                                ;     DURING THE DROP-UNIT SECTION.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                                        .EVEN

100264                                  ENDDU
100264                                                                      L10013:
100264  104453                                                                      TRAP    C$DU
```

```
                                .SBTTL  ADD UNIT SECTION

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

 100266                                 BGNAU
 100266                                                                         L$AU::

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;       INSERT ADD CODE HERE.  THIS CODE WILL BE EXECUTED AFTER
                                ;       AN "ADD" COMMAND.  THE PURPOSE OF THIS CODE IS TO DO ANY
                                ;       HOUSEKEEPING THAT MAY BE NECESSARY AFTER A UNIT HAS BEEN ADDED.
                                ;       THIS SECTION IS OPTIONAL.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 100266                                 EXIT    AU
 100266 000167                                                                          .WORD   J$JMP
 100270 000000                                                                          .WORD   L10014-2-.

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;     INSERT LOCAL STORAGE THAT IS USED ONLY
                                ;     DURING THE ADD-UNIT SECTION.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;     INSERT MESSAGES THAT ARE USED ONLY
                                ;     DURING THE ADD-UNIT SECTION.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                                        .EVEN

 100272                                 ENDAU
 100272                                                                         L10014:
 100272 104452                                                                          TRAP    C$AU


                                .SBTTL  TEST 1:

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;     APPEND THE NAME OF THIS TEST TO THE .SBTTL, AS SHOWN IN THE
                                ;     FOLLOWING EXAMPLE.    .SBTTL  TEST 1:  NAME OF TEST
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


                                ;++
                                ; TEST TO ...
                                ;--

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;     CHANGE THE PHRASE "TEST TO ..." TO BE A FUNCTIONAL
                                ;     DESCRIPTION OF THE HARDWARE TEST WHICH FOLLOWS.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                        ;       INSERT PROGRAM EQUATES THAT ARE USED ONLY IN THIS TEST.
                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

100274                                          BGNTST
100274                                                                                  T1::

                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                        ;       INSERT THE CODING FOR THIS HARDWARE TEST.
                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                        .SBTTL  GETCL   COMMAND LINE FETCH & INTERPRETATION SECTION

100274  105037  003161                  GETCL:  CLRB    P$GDBD                  ;CLEAR CMD LINE PARSING ERROR FLAG
100300  105037  003160                          CLRB    P$NNUF
100304                                          GMANID  CLI$PM,CMDBUF,A,0,1,72.,NO      ;GET CMD LINE FROM OPERATOR
100304  104443                                                                                  TRAP    C$GMAN
100306  000406                                                                                  BR      10000$
100310  002200                                                                                  .WORD   CMDBUF
100312  000142                                                                                  .WORD   T$CODE
100314  052436                                                                                  .WORD   CLI$PM
100316  000000                                                                                  .WORD   0
100320  000001                                                                                  .WORD   T$LOLIM
100322  000110                                                                                  .WORD   T$HILIM
100324                                                                                  10000$:
100324  012737  002200  003144                  MOV     #CMDBUF,P$BUFA
100332  012737  051070  003146                  MOV     #CLITRE,P$TREE
100340  012737  100456  003150                  MOV     #CLIACT,P$ACT
100346  005037  003672                          CLR     CFLAG                   ;CLEAR QUALIFIER FLAG
100352  004737  075216                          JSR     PC,P$TRV                ;GO PARSE COMMAND TREE
100356  105737  003161                          TSTB    P$GDBD                  ;SEE IF PARSED OK, OR AN ERROR
100362  001412                                  BEQ     1$
100364                                          PRINTF  #CLIERM                 ;IF NOT PRINT ERROR MESSAGE
100364  012746  052445                                                                          MOV     #CLIERM,-(SP)
100370  012746  000001                                                                          MOV     #1,-(SP)
100374  010600                                                                                  MOV     SP,R0
100376  104417                                                                                  TRAP    C$PNTF
100400  062706  000004                                                                          ADD     #4,SP
100404  000137  100274                          JMP     GETCL
100410  105737  003160                  1$:     TSTB    P$NNUF                  ;SEE IF INCOMPLETE COMMAND TYPED
100414  001412                                  BEQ 10$
100416                                          PRINTF  #CLINUF                 ;IF NOT PRINT ERROR MESSAGE
100416  012746  052476                                                                          MOV     #CLINUF,-(SP)
100422  012746  000001                                                                          MOV     #1,-(SP)
100426  010600                                                                                  MOV     SP,R0
100430  104417                                                                                  TRAP    C$PNTF
100432  062706  000004                                                                          ADD     #4,SP
100436  000137  100274                          JMP     GETCL
100442  022737  000020  003672          10$:    CMP     #CEXIT,CFLAG            ;WAS EXIT COMMAND TYPED?
100450  001311                                  BNE     GETCL                   ;  IF NOT GET NEW COMMAND LINE
100452                                          EXIT    TST                     ;  ELSE EXIT
100452  104432                                                                                  TRAP    C$EXIT
100454  012272                                                                                  .WORD   L10015-.

                                        .SBTTL  CLI ACTION TABLE AND ROUTINES
                                        ;       USER MUST CLEAR/SET P$GDBD IF USE "CLIBIF" IN CONNECTION WITH ACTION
                                        ;       R2 WILL HOLD ACTION CODE FROM PARSING (CLI) NODE
```

```
 6385  100456                          CLIACT:
 6386  100456  006302                          ASL     R2              ;MULTIPLY ACTION CODE BY 2
 6387  100460  016202  100474                  MOV     10$(R2),R2      ;OFFSET VALUE
 6388  100464  062702  100474                  ADD     #10$,R2         ;ADD BASE VALUE
 6389  100470  004712                          JSR     PC,(R2)         ;GO DO ACTION
 6390  100472  000207                          RTS     PC              ;RETURN TO TRVACT
 6391
 6392                                                                  ;BRIEF DESCRIPTION OF ACTION TAKEN
 6393  100474  000122                  10$:    .WORD   ACTNUL-10$      ;0-NULL
 6394  100476  000124                          .WORD   ACTHLP-10$      ;1-HELP
 6395  100500  000162                          .WORD   ACTNOD-10$      ;2-NODE
 6396  100502  000472                          .WORD   ACTBLD-10$      ;3-BUILD
 6397  100504  003542                          .WORD   ACTRUN-10$      ;4-RUN SPECIFIED TEST
 6398  100506  006326                          .WORD   ACTPAT-10$      ;5-SET 'MESSAGE PATTERN' TEST FLAG
 6399  100510  010164                          .WORD   ACTSAV-10$      ;6-SAVE NODE TABLE
 6400  100512  001214                          .WORD   ACTSUM-10$      ;7-PRINT SUMMARY TABLE
 6401  100514  001570                          .WORD   ACTIDT-10$      ;10-REQUEST ID
 6402  100516  002654                          .WORD   ACTEXT-10$      ;11-EXIT
 6403  100520  000114                          .WORD   ACTNUF-10$      ;12-NOT ENOUGH INFO
 6404  100522  002664                          .WORD   ACTXAD-10$      ;13-EXTRACT NI NODE ADDRESS FROM INPUT LINE
 6405  100524  002746                          .WORD   ACTSR4-10$      ;14-SAVE POINTER TO BEGINING OF ADDRESS STRING
 6406  100526  007670                          .WORD   ACTSND-10$      ;15-SET 'NODE' FLAG FOR SHOW COMMAND
 6407  100530  002754                          .WORD   ACTALP-10$      ;16-SET 'ALPHA' FLAG
 6408  100532  002764                          .WORD   ACTONE-10$      ;17-SET 'ONES' FLAG
 6409  100534  002774                          .WORD   ACTZRO-10$      ;20-SET 'ZEROS' FLAG
 6410  100536  003004                          .WORD   ACT1AL-10$      ;21-SET '1ALT' FLAG
 6411  100540  003014                          .WORD   ACT0AL-10$      ;22-SET '0ALT' FLAG
 6412  100542  003024                          .WORD   ACTCTT-10$      ;23-SET 'CCITT' FLAG
 6413  100544  003034                          .WORD   ACTOPR-10$      ;24-SET 'OPER SEL' FLAG
 6414  100546  003162                          .WORD   ACTTYP-10$      ;25-DETERMINE MESSAGE TYPE
 6415  100550  003170                          .WORD   ACTSZE-10$      ;26-DETERMINE MESSAGE SIZE
 6416  100552  003246                          .WORD   ACTCPY-10$      ;27-DETERMINE MESSAGE COPIES
 6417  100554  003324                          .WORD   ACTNAD-10$      ;30-SET 'NODE/ADDRESS' FLAG
 6418  100556  003462                          .WORD   ACTNAL-10$      ;31-SET 'NODE/ALL' FLAG
 6419  100560  003662                          .WORD   ACTRNA-10$      ;32-SET 'ALL' FLAG FOR RUN COMMAND
 6420  100562  004724                          .WORD   ACTRNL-10$      ;33-SET 'LOOPPAIR' FLAG FOR RUN CMD
 6421  100564  006410                          .WORD   ACTSMS-10$      ;34-SHOW CURRENT MESSAGE PARAMETERS
 6422  100566  006502                          .WORD   ACTCMS-10$      ;35-RESET MESSAGE PARAMETERS TO DEFAULT
 6423  100570  006606                          .WORD   ACTCNT-10$      ;36-SET 'COUNTER' FLAG FOR SHOW COMMAND
 6424  100572  010044                          .WORD   ACTCNL-10$      ;37-CLEAR LOGICAL NODE NAMED FROM TABLE
 6425  100574  010130                          .WORD   ACTFCT-10$      ;40-INITIATE UNA PORT COMMAND FUNCTION
 6426  100576  010230                          .WORD   ACTUNS-10$      ;41-UNSAVE NODE TABLE
 6427  100600  010332                          .WORD   ACTCSU-10$      ;42-CLEAR SUMMARY TABLE
 6428  100602  004276                          .WORD   ACTDIR-10$      ;43-SET 'LOOP DIRECT' FLAG FOR RUN COMMAND
 6429  100604  010406                          .WORD   ACTDFT-10$      ;44-LOOK FOR PASS COUNT DEFAULT
 6430  100606  010454                          .WORD   ACTUSF-10$      ;45-UNSAVE NODE TABLE FROM A FILE
 6431
```

```
6432
6433                                    ;
6434                                    ;ACTION ROUTINE TO INDICATE THAT NOT ENOUGH COMMAND
6435                                    ;INFORMATION HAS BEEN ENTERED
6436                                    ;
6437
6438  100610  112737  177777  003160   ACTNUF: MOVB    #-1,P$NNUF              ;SET FLAG TO SAY NEED MORE OF COMMAND
6439
6440                                    ;
6441                                    ;ACTION ROUTINE TO DO NOTHING
6442                                    ;
6443
6444  100616  000207                    ACTNUL: RTS     PC                      ;RETURN TO PARSER
6445
6446
6447                                    ;
6448                                    ;ACTION ROUTINE TO PRINT OUT HELP FILE
6449                                    ;
6450
6451  100620                            ACTHLP: P$PUSH  R2                      ;SAVE R2
6452  100622  012702  003164                    MOV     #HLPTAB,R2              ;MOVE POINTER TO BEGINING OF HELP FILE
6453  100626                            10$:    PRINTF  (R2)+                   ;PRINT LINE AND INCREMENT POINTER
6454  100626  012246                                                                    MOV     (R2)+,-(SP)
6455  100630  012746  000001                                                           MOV     #1,-(SP)
6456  100634  010600                                                                    MOV     SP,R0
6457  100636  104417                                                                    TRAP    C$PNTF
6458  100640  062706  000004                                                           ADD     #4,SP
6459  100644  020227  003260                    CMP     R2,#HLPEND              ;SEE IF ENTIRE FILE PRINTED
6460  100650  001366                            BNE     10$                     ;IF NOT, PRINT MORE
6461  100652                                    P$POP   R2                      ;ELSE, RESTORE R2 AND RETURN
6462  100654  000207                            RTS     PC
6463
6464
6465                                    ;
6466                                    ;ACTION ROUTINE TO READ IN NODE PHY. ADDRESS, STORE IT IN ADRBUF
6467                                    ;AND ENTER IT INTO THE NODE TABLE
6468                                    ;
6469
6470  100656  105037  003160            ACTNOD: CLRB    P$NNUF                  ;CLEAR NOTNUF FLAG
6471  100662  004737  076330                    JSR     PC,TRVADR               ;TRAVERSE ADDRESS, CHECK IF TARGET OR ASSIST
6472  100666  105737  003161                    TSTB    P$GDBD                  ;CHECK IF RESULTS OK
6473  100672  001134                            BNE     50$                     ;IF NOT, RETURN WITH -1 IN P$GDBD
6474  100674                            10$:    CALL    EDPACK CBOADR,#ADRBUF,#6        ;GET ADDRESS INTO BUFFER
6475  100716                                    P$POP   R1                      ;CHECK RESULTS FOR NUMBER OF CHAR.S
6476  100720  001411                            BEQ     15$                     ;IF OK, BRANCH TO 15$
6477  100722                                    PRINTF  #CADRER                 ;ELSE PRINT ERROR MESSAGE
6478  100722  012746  053232                                                           MOV     #CADRER,-(SP)
6479  100726  012746  000001                                                           MOV     #1,-(SP)
6480  100732  010600                                                                    MOV     SP,R0
6481  100734  104417                                                                    TRAP    C$PNTF
6482  100736  062706  000004                                                           ADD     #4,SP
6483  100742  000510                            BR      50$                     ;AND RETURN
6484  100744                            15$:    CALL    CMPADR #ADRBUF,#ILLADR          ;SEE IF ILLEGAL ADDRESS
6485  100762                                    P$POP   R1
6486  100764  001021                            BNE     17$                     ;IF YES, PRINT ERROR MESSAGE
6487  100766                                    PRINTF  #ILADMS
```

```
 6488  100766  012746  053051                                                  MOV     #ILADMS,-(SP)
 6489  100772  012746  000001                                                  MOV     #1,-(SP)
 6490  100776  010600                                                          MOV     SP,R0
 6491  101000  104417                                                          TRAP    C$PNTF
 6492  101002  062706  000004                                                  ADD     #4,SP
 6493  101006                        PRINTF  #ILADM1
 6494  101006  012746  053135                                                  MOV     #ILADM1,-(SP)
 6495  101012  012746  000001                                                  MOV     #1,-(SP)
 6496  101016  010600                                                          MOV     SP,R0
 6497  101020  104417                                                          TRAP    C$PNTF
 6498  101022  062706  000004                                                  ADD     #4,SP
 6499  101026  000456                BR      50$
 6500  101030                  17$:  CALL    BINHEX #ADRBUF,#6,#STRBUF       ;CONVERT BINARY ADDRESS
 6501                                                                         ;INTO ASCII STRING
 6502  101052  022737  000001  003672        CMP     #CASIST,CFLAG           ;SEE IF TARGET OR ASSIST
 6503  101060  001407                        BEQ     20$
 6504  101062  012737  062613  002312        MOV     #ARGTY7,KEYWD2          ;MOVE 'TARGET' INTO KEYWD2
 6505  101070  012737  000000  002400        MOV     #CTARGT,NODTY           ;MOVE TARGET INTO NODE TYPE
 6506  101076  000406                        BR      25$
 6507  101100  012737  062603  002312  20$:  MOV     #ARGTY6,KEYWD2          ;MOVE 'ASSIST' INTO KEYWD2
 6508  101106  012737  000001  002400        MOV     #CASIST,NODTY
 6509  101114                        25$:    CALL    ENTRND                  ;CALL ROUTINE TO ENTER NODE ON TABLE
 6510  101122                                P$POP   R1                      ;CHECK RESULTS
 6511  101124  001017                        BNE     50$                     ;IF NODE TABLE FULL, RETURN
 6512  101126  012737  062510  002310        MOV     #CMDTY7,KEYWD1          ;ELSE, MOVE "NODE" INTO KEYWD1
 6513  101134                                PRINTS  #MSG2,KEYWD2,#STRBUF    ;INDICATE IF TARGET OR ASSIST
 6514  101134  012746  002322                                                  MOV     #STRBUF,-(SP)
 6515  101140  013746  002312                                                  MOV     KEYWD2,-(SP)
 6516  101144  012746  055177                                                  MOV     #MSG2,-(SP)
 6517  101150  012746  000003                                                  MOV     #3,-(SP)
 6518  101154  010600                                                          MOV     SP,R0
 6519  101156  104416                                                          TRAP    C$PNTS
 6520  101160  062706  000010                                                  ADD     #10,SP
 6521  101164  000207                  50$:  RTS     PC
 6522
 6523
 6524                                  ;
 6525                                  ;ACTION ROUTINE TO BUILD NODE TABLE USING THE PERIODIC ADDRESS
 6526                                  ;IDENTIFICATION MESSAGES SENT OUT BY NODE ON THE NI
 6527                                  ;
 6528
 6529  101166                  ACTBLD: PRINTS  #MSG1                   ; PRINT 'BUILD' COMMAND MESSAGE
 6530  101166  012746  054711                                                  MOV     #MSG1,-(SP)
 6531  101172  012746  000001                                                  MOV     #1,-(SP)
 6532  101176  010600                                                          MOV     SP,R0
 6533  101200  104416                                                          TRAP    C$PNTS
 6534  101202  062706  000004                                                  ADD     #4,SP
 6535  101206                        PRINTS  #MSG11
 6536  101206  012746  055024                                                  MOV     #MSG11,-(SP)
 6537  101212  012746  000001                                                  MOV     #1,-(SP)
 6538  101216  010600                                                          MOV     SP,R0
 6539  101220  104416                                                          TRAP    C$PNTS
 6540  101222  062706  000004                                                  ADD     #4,SP
 6541  101226                        PRINTS  #MSG12
 6542  101226  012746  055137                                                  MOV     #MSG12,-(SP)
 6543  101232  012746  000001                                                  MOV     #1,-(SP)
```

```
                                                                                    MOV     SP,R0
                                                                                    TRAP    C$PNTS
                                                                                    ADD     #4,SP
                        P$PUSH  R2,R3,R4                ; SAVE REGISTERS
                        CALL    FUNCT #WDMULA           ; WRITE MULTICAST ADDRESS LIST
                        P$POP   R2                      ; CHECK FOR ERROR
                        BEQ     10$                     ; IF OK, CONTINUE
                        ERRHRD  25,EMSG25               ; ELSE REPORT ERROR
                                                                                    TRAP    C$ERHRD
                                                                                    .WORD   25
                                                                                    .WORD   EMSG25
                                                                                    .WORD   0
10$:                    CLR     TEMP                    ; CLEAR 'NO. NODES IN LAST MIN.' COUNTER
                        CLR     TEMP1                   ; CLEAR NODE TYPE ARGUMENT (SET TO TARGET)
                        CLR     TEMP2                   ; SET INTERVAL COUNTER
                        MOV     #12,TEMP3               ; SET 'MINS. SINCE LAST NEW NODE' COUNTER
                        MOV     #NODTBL,SLOT            ; SET SLOT TO BEGINING OF NODE TABLE
                        MOV     #TIMERS,R4              ; SET UP FOR 1 MINUTE LOOP
19$:                    MOV     #60.,(R4)
20$:                    BREAK                           ; ALLOW FOR CONTROL C INTERRUPTION
                                                                                    TRAP    C$BRK
                        TST     (R4)                    ; SEE IF INTERVAL IS UP
                        BEQ     40$                     ; IF YES, BRANCH
                        CALL    RECEVE                  ;  ELSE, CHECK FOR RECEPTION OF ID MESSAGE
                        P$POP   R2                      ; R2 HOLDS NO OF MESSAGES RECEIVED
                        BEQ     20$                     ; IF NONE, KEEP LOOKING
                        MOV     RRGNXT,R3               ; IF ONE, GET RECEIVE RING POINTER
                        CALL    GETRNX  #RRGNXT         ; UPDATE POINTER
                        MOV     2(R3),R3                ; POINT R3 TO MESSAGE BUFFER
                        ADD     #SOURCE,R3              ; POINT R3 TO NODE ADDRESS
                        MOV     #NODTBL,R2              ; POINT R2 TO NODE TABLE
21$:                    CALL    CMPADR  R2,R3           ; SEE IF NODE ALREADY ON TABLE
                        P$POP   R1
                        BEQ     20$                     ; IF SAME, DON'T ADD TO TABLE
22$:                    ADD     #10,R2                  ;  ELSE CHECK NEXT TABLE ENTRY
                        CMP     #-1,(R2)                ;  SEE IF AT END OF TABLE
                        BNE     21$                     ;   IF NO, COMPARE NEXT ENTRY
                        CALL    FINDSL                  ;   IF YES, GET NEXT TABLE SLOT
                        P$POP   R2                      ; SEE IF TABLE FULL
                        BNE     35$                     ;  IF YES, BRANCH
                        MOV     SLOT,R2                 ;  IF NO, ADD NODE TO TABLE
                        MOV     (R3)+,(R2)+             ;
                        MOV     (R3)+,(R2)+             ;   SIX BYTES WORTH
                        MOV     (R3),(R2)               ;
                        MOVB    TEMP1,3(R2)             ; SET NODE TYPE (TARGET OR ASSIST)
                        COMB    TEMP1                   ;  SWITCH TYPE FOR NEXT TIME
                        BICB    #376,TEMP1              ;
                        INC     TEMP                    ; INCREMENT 'NODES IN LAST MIN.' COUNTER
                        MOV     #13,TEMP3               ; RESET 'MINS. SINCE LAST NODE' COUNTER
                        BR      20$                     ; CHECK FOR MORE INPUT
35$:                    PRINTB  #TABFUL,#NOD            ; PRINT "TABLE FULL" MESSAGE
                                                                                    MOV     #NOD,-(SP)
                                                                                    MOV     #TABFUL,-(SP)
                                                                                    MOV     #2,-(SP)
                                                                                    MOV     SP,R0
                                                                                    TRAP    C$PNTB
```

```
6600  101544  062706  000006                                                          ADD     #6,SP
6601  101550  000430                      BR      50$
6602  101552  005337  050556      40$:    DEC     TEMP3                   ; SEE IF 10 MINS SINCE LAST NODE
6603  101556  001425                      BEQ     50$                     ;  IF YES, EXIT
6604  101560  005237  050554              INC     TEMP2                   ; SEE IF TIME IS UP
6605  101564  023727  050554  000050      CMP     TEMP2,#40.
6606  101572  001417                      BEQ     50$                     ; IF YES, EXIT
6607  101574                              PRINTS  #BLDMSG,TEMP,TEMP2      ;  ELSE, PRINT "STILL WORKING" MESSAGE
6608  101574  013746  050554                                                          MOV     TEMP2,-(SP)
6609  101600  013746  050550                                                          MOV     TEMP,-(SP)
6610  101604  012746  052757                                                          MOV     #BLDMSG,-(SP)
6611  101610  012746  000003                                                          MOV     #3,-(SP)
6612  101614  010600                                                                  MOV     SP,R0
6613  101616  104416                                                                  TRAP    C$PNTS
6614  101620  062706  000010                                                          ADD     #10,SP
6615  101624  005037  050550              CLR     TEMP
6616  101630  000642                      BR      19$
6617  101632  012737  000000  050044  50$:    MOV     #0,$WDMC+4              ; CLEAR MULTICAST ADDRESS LIST
6618  101640                              CALL    FUNCT   #WDMULA         ;  WRITE 0 INTO LIST LENGTH
6619  101652                              P$POP   R2                      ; CHECK FOR ERROR
6620  101654  001404                      BEQ     54$                     ;  IF OK CONTINUE
6621  101656                              ERRHRD  34,EMSG25               ;  ELSE, REPORT ERROR
6622  101o56  104456                                                          TRAP    C$ERHRD
6623  101660  000042                                                          .WORD   34
6624  101662  065442                                                          .WORD   EMSG25
6625  101664  000000                                                          .WORD   0
6626  101666  004737  110364      54$:    JSR     PC,ACTSND               ; PRINT NODE TABLE
6627  101672  012737  000400  050044      MOV     #400,$WDMC+4            ; RESET MULTICAST LIST FOR 1 ENTRY
6628  101700                              P$POP   R2,R3,R4                ; RESTORE REGISTERS
6629  101706  000207                      RTS     PC
6630
6631
6632                                      ;
6633                                      ;ACTION ROUTINE TO PRINT OUT THE SUMMARY DATA
6634                                      ;
6635
6636  101710  105037  003160      ACTSUM: CLRB    P$NNUF                  ;CLEAR NOTNUF FLAG
6637  101714                              P$PUSH  R2,R3,R4
6638  101722  012701  002656              MOV     #STATBL,R1              ; MOVE ADDRESS OF TABLE TO R1
6639  101726  005711                      TST     (R1)                    ; SEE IF TABLE EMPTY
6640  101730  001013                      BNE     5$                      ;  IF NOT, CONT.
6641  101732                              PRINTF  #TABEMT,#SUMM           ;  ELSE PRINT 'TABLE EMPTY' MESSAGE
6642  101732  012746  053706                                                          MOV     #SUMM,-(SP)
6643  101736  012746  053632                                                          MOV     #TABEMT,-(SP)
6644  101742  012746  000002                                                          MOV     #2,-(SP)
6645  101746  010600                                                                  MOV     SP,R0
6646  101750  104417                                                                  TRAP    C$PNTF
6647  101752  062706  000006                                                          ADD     #6,SP
6648  101756  000536                      BR      30$                     ;  EXIT
6649  101760  021127  177777      5$:     CMP     (R1),#-1                ; SEE IF AT END OF TABLE
6650  101764  001533                      BEQ     30$                     ;  IF YES, EXIT
6651  101766  005711                      TST     (R1)                    ; SEE IF REST OF TABLE EMPTY
6652  101770  001531                      BEQ     30$                     ;  IF YES, EXIT
6653  101772                              CALL    BINHEX  R1,#6,#STRBUF   ; PRINT SUMMARY DATA
6654  102012                              PRINTF  #SUMMS1,#STRBUF         ;  NODE ADDRESS
6655  102012  012746  002322                                                          MOV     #STRBUF,-(SP)
```

```
6656  102016  012746  067215                                                  MOV     #SUMMS1,-(SP)
6657  102022  012746  000002                                                  MOV     #2,-(SP)
6658  102026  010600                                                          MOV     SP,R0
6659  102030  104417                                                          TRAP    C$PNTF
6660  102032  062706  000006                                                  ADD     #6,SP
6661  102036                        PRINTF  #SUMMS2                 ;  FIRST HEADER
6662  102036  012746  067236                                                  MOV     #SUMMS2,-(SP)
6663  102042  012746  000001                                                  MOV     #1,-(SP)
6664  102046  010600                                                          MOV     SP,R0
6665  102050  104417                                                          TRAP    C$PNTF
6666  102052  062706  000004                                                  ADD     #4,SP
6667  102056  016102  000006        MOV     6(R1),R2                ;   RX NOT COMPLETE
6668  102062  016103  000010        MOV     10(R1),R3               ;   RX COMPLETE
6669  102066  016104  000012        MOV     12(R1),R4               ;   LENGTH ERRORS
6670  102072                        PRINTF  #SUMMS3,R2,R3,R4        ;    PRINT THEM OUT
6671  102072  010446                                                          MOV     R4,-(SP)
6672  102074  010346                                                          MOV     R3,-(SP)
6673  102076  010246                                                          MOV     R2,-(SP)
6674  102100  012746  067322                                                  MOV     #SUMMS3,-(SP)
6675  102104  012746  000004                                                  MOV     #4,-(SP)
6676  102110  010600                                                          MOV     SP,R0
6677  102112  104417                                                          TRAP    C$PNTF
6678  102114  062706  000012                                                  ADD     #12,SP
6679  102120                        PRINTF  #SUMMS4                 ;  SECOND HEADER
6680  102120  012746  067351                                                  MOV     #SUMMS4,-(SP)
6681  102124  012746  000001                                                  MOV     #1,-(SP)
6682  102130  010600                                                          MOV     SP,R0
6683  102132  104417                                                          TRAP    C$PNTF
6684  102134  062706  000004                                                  ADD     #4,SP
6685  102140  016102  000014        MOV     14(R1),R2               ;   COMPARE ERRORS
6686  102144  062701  000016        ADD     #16,R1                  ;   BYTES COMPARED
6687  102150                        CALL    BINDEC  R1              ;    PUT INTO ASCII STRING
6688  102160                        PRINTF  #SUMMS5,R2,#DECSTR      ;    PRINT THEM OUT
6689  102160  012746  075200                                                  MOV     #DECSTR,-(SP)
6690  102164  010246                                                          MOV     R2,-(SP)
6691  102166  012746  067434                                                  MOV     #SUMMS5,-(SP)
6692  102172  012746  000003                                                  MOV     #3,-(SP)
6693  102176  010600                                                          MOV     SP,R0
6694  102200  104417                                                          TRAP    C$PNTF
6695  102202  062706  000010                                                  ADD     #10,SP
6696  102206  062701  000004        ADD     #4,R1                   ;   BYTES TRANSFERED
6697  102212                        CALL    BINDEC  R1              ;    PUT INTO ASCII STRING
6698  102222                        PRINTF  #SUMMS6,#DECSTR         ;    PRINT
6699  102222  012746  075200                                                  MOV     #DECSTR,-(SP)
6700  102226  012746  067452                                                  MOV     #SUMMS6,-(SP)
6701  102232  012746  000002                                                  MOV     #2,-(SP)
6702  102236  010600                                                          MOV     SP,R0
6703  102240  104417                                                          TRAP    C$PNTF
6704  102242  062706  000006                                                  ADD     #6,SP
6705  102246  062701  000004        ADD     #4,R1                   ; POINT R1 TO NEXT TABLE ENTRY
6706  102252  000642                BR      5$                      ; DO IT ALL AGAIN
6707  102254                30$:    P$POP   R2,R3,R4
6708  102262  000207                RTS     PC
6709
6710
6711                          ;
```

```
 6712                                   ;ACTION ROUTINE TO INITIATE THE REQUEST ID TEST TO THE SPECIFIED NODE
 6713                                   ;
 6714
 6715                                   ;--+
 6716                                   ; FUNCTIONAL DESCRIPTION
 6717                                   ;                THIS SUBROUTINE BUILDS AND TRANSMITS REQUEST ID PACKETS
 6718                                   ;                TO THE NODE SPECIFIED BY THE OPERATOR IN THE COMMAND LINE.
 6719                                   ;                THE SYSTEM ID INFO OF THE SPECIFIED NODE IS THEN DISPLAYED.
 6720                                   ;                IF THE NODE DOES NOT RESPOND BEFORE 60 SECONDS HAVE PASSED
 6721                                   ;                AN ERROR IS REPORTED TO THE OPERATOR.
 6722                                   ;
 6723                                   ; INPUTS -       IMPLICIT - THE SPECIFIED NODE ADDRESS IS LOCATED IN ADRBUF.
 6724                                   ;
 6725                                   ; OUTPUTS -      SYSTEM ID INFO OR ERROR MESSAGE PRINTED TO OPERATOR.
 6726                                   ;
 6727                                   ; CALLING PROCEDURE - JSR PC, ACTIDT
 6728                                   ;
 6729                                   ; SIDE EFFECTS - XRGNXT POINTER IS UPDATED BY A CALL TO BLDREQ SUB.
 6730                                   ;
 6731                                   ; REGISTER USAGE - R1 POINTS TO $WDMO FOR WRITE MODE OPERATIONS.
 6732                                   ;                  R2 IS SCRATCH.
 6733                                   ;                  R3 POINTS TO THE RECEIVED MESSAGE BUFFER.
 6734                                   ;                  R4 POINTS TO TIMOUT TIMER
 6735                                   ;
 6736                                   ;--+
 6737
 6738  102264  105737  003162           ACTIDT: TSTB    P$AERR                  ;SEE IF ADDRESS ENTERED WAS VALID
 6739  102270  001402                           BEQ     1$                      ; IF NOT, EXIT ACTION ROUTINE
 6740  102272  000137  103240                   JMP     55$
 6741  102276  105037  003160           1$:     CLRB    P$NNUF                  ;CLEAR NOTNUF FLAG
 6742  102302                                   CALL    CMPADR  #ADRBUF,#ILLADR ; SEE IF ILLEGAL ADDRESS
 6743  102320                                   P$POP   R1
 6744  102322  001012                           BNE     2$                      ;  IF NO, CONTINUE
 6745  102324                                   PRINTF  #ILADMS                 ;  ELSE PRINT ILLEGAL ADDRESS MESSAGE
 6746  102324  012746  053051                                                           MOV     #ILADMS,-(SP)
 6747  102330  012746  000001                                                           MOV     #1,-(SP)
 6748  102334  010600                                                                   MOV     SP,R0
 6749  102336  104417                                                                   TRAP    C$PNTF
 6750  102340  062706  000004                                                           ADD     #4,SP
 6751  102344  000137  103240                   JMP     55$
 6752  102350                           2$:     P$PUSH  R1,R2,R3,R4             ; SAVE REGISTERS
 6753  102360                                   CALL    CMPADR  #ADRBUF,#PHYADR ; SEE IF ADDRESS IS OWN (HOST NODE)
 6754  102376                                   P$POP   R1                      ;
 6755  102400  001535                           BEQ     27$
 6756  102402  012737  177776  050554         MOV     #-2,TEMP2               ; SET COUNTER FOR NO. OF TIMES TRIED
 6757  102410  012701  050300                 MOV     #$WDMO,R1               ; SET UP TO WRITE MODE
 6758  102414  012761  010000  000002         MOV     #10000,2(R1)            ;  10000: TPAD =1 (PAD TRANSMIT BUFFERS)
 6759  102422                                   CALL    FUNCT #WDMODE           ; WRITE MODE
 6760  102434                                   P$POP   R2                      ; CHECK FOR ERROR
 6761  102436  001402                           BEQ     3$                      ;  BR IF ERROR
 6762  102440  000137  103146                   JMP     40$
 6763  102444                           3$:     CALL    BLDREQ                  ; BUILD REQUEST ID MESSAGE PACKET
 6764  102452                                   CALL    XMIT                    ; TRANSMIT REQUEST
 6765  102460                                   P$POP   R2                      ; GET RESULTS, R2 = SUCCESS/FAILURE
 6766  102462  001402                           BEQ     4$                      ;  IF OK BRANCH
 6767  102464  000137  103160                   JMP     45$                     ;  ELSE JUMP TO 45$
```

```
6768  102470  005737  050534          4$:     TST     RETRYS                  ; SEE IF FAILED DUE TO EXCESSIVE COLLISIONS
6769  102474  001412                          BEQ     5$                      ;  IF NO, CONT.
6770  102476                                  PRINTF  #RTRYER                 ;  YES, PRINT 'EXCESSIVE COLLISIONS' MESSAGE
6771  102476  012746  052675                                                          MOV     #RTRYER,-(SP)
6772  102502  012746  000001                                                          MOV     #1,-(SP)
6773  102506  010600                                                                  MOV     SP,R0
6774  102510  104417                                                                  TRAP    C$PNTF
6775  102512  062706  000004                                                          ADD     #4,SP
6776  102516  000137  103206                  JMP     52$                     ;  EXIT
6777  102522  012704  003720          5$:     MOV     #TIMERS,R4              ; SET UP FOR 10 SECOND TIMOUT
6778  102526  012714  000012                  MOV     #10.,(R4)
6779  102532                          15$:    BREAK
6780  102532  104422                                                                  TRAP    C$BRK
6781  102534  005714                          TST     (R4)                    ; SEE IF TIME HAS EXPIRED
6782  102536  001425                          BEQ     20$                     ;  IF YES, BRANCH
6783  102540                                  CALL    RECEVE                  ; CHECK FOR ANSWER
6784  102546                                  P$POP   R2                      ;  R2 HOLDS NO. OF BUFFERS RECEIVED
6785  102550  001770                          BEQ     15$                     ;  IF NO BUFFERS RECIEVED, LOOP
6786  102552  013703  003764                  MOV     RRGNXT,R3               ; GET RECEIVE RING POINTER
6787  102556                                  CALL    GETRNX #RRGNXT          ; UPDATE POINTER
6788  102570  016303  000002                  MOV     2(R3),R3                ; POINT R3 TO MESSAGE BUFFER
6789  102574  026327  000022  051115          CMP     SIRCPT(R3),#'MR         ; SEE IF MESSAGE RECIEVED IS IN REPLY TO ONE SEN
6790  102602  001424                          BEQ     25$                     ;  IF YES, BRANCH TO 25$
6791  102604  005237  050554                  INC     TEMP2                   ;  INCREMENT RETRY COUNTER
6792  102610  001350                          BNE     15$                     ;  IF NO, LOOK FOR CORRECT REPLY MESSAGE
6793
6794  102612                          20$:    PRINTF  #EMSG22                 ;  ELSE, REPORT ERROR
6795  102612  012746  065266                                                          MOV     #EMSG22,-(SP)
6796  102616  012746  000001                                                          MOV     #1,-(SP)
6797  102622  010600                                                                  MOV     SP,R0
6798  102624  104417                                                                  TRAP    C$PNTF
6799  102626  062706  000004                                                          ADD     #4,SP
6800  102632  005237  050500                  INC     S.NREC                  ;  UPDATE SUMMARY DATA
6801  102636  000137  103170                  JMP     51$                     ;  AND EXIT
6802
6803  102642  012703  050413          22$:    MOV     #UCB22+13,R3            ; POINT R3 TO SAME PLACE IN MESSAGE AS
6804  102646  012702  050454                  MOV     #UCB22+54,R2            ;  IF IT WERE RECEIVED SYS ID, R2 TO
6805  102652  000410                          BR      27$                     ;  NODE ADDRESS
6806
6807  102654  005237  050476          25$:    INC     S.REC                   ; INCREMENT 'RECEIVED MESSAGES' COUNTER
6808  102660  062737  000056  050510          ADD     #46.,S.XFER             ; UPDATE 'BYTES TRANSFERED' COUNTER
6809  102666  010302                          MOV     R3,R2                   ; PUT POINTER INTO R2
6810  102670  062702  000042                  ADD     #SIADDR,R2              ; POINT R2 TO ADDRESS
6811  102674                          27$:    CALL    BINHEX R2,#6,#STRBUF    ; PUT ADDRESS INTO STRBUF
6812  102714                                  PRINTF  #SIMSG1,#STRBUF         ; PRINT REMOTE NODE ADDRESS
6813  102714  012746  002322                                                          MOV     #STRBUF,-(SP)
6814  102720  012746  066440                                                          MOV     #SIMSG1,-(SP)
6815  102724  012746  000002                                                          MOV     #2,-(SP)
6816  102730  010600                                                                  MOV     SP,R0
6817  102732  104417                                                                  TRAP    C$PNTF
6818  102734  062706  000006                                                          ADD     #6,SP
6819  102740  016302  000022                  MOV     SIRCPT(R3),R2           ; GET RECEIPT NUMBER
6820  102744                                  PRINTF  #SIMSG2,R2              ; PRINT RECEIPT NUMBER
6821  102744  010246                                                                  MOV     R2,-(SP)
6822  102746  012746  066477                                                          MOV     #SIMSG2,-(SP)
6823  102752  012746  000002                                                          MOV     #2,-(SP)
```

```
 6824  102756  010600                                                          MOV     SP,R0
 6825  102760  104417                                                          TRAP    C$PNTF
 6826  102762  062706  000006                                                  ADD     #6,SP
 6827  102766  116302  000027          MOVB    SIVERS(R3),R2          ; GET VERSION NO. AND PRINT
 6828  102772                         PRINTF  #SIMSG3,R2
 6829  102772  010246                                                          MOV     R2,-(SP)
 6830  102774  012746  066532                                                  MOV     #SIMSG3,-(SP)
 6831  103000  012746  000002                                                  MOV     #2,-(SP)
 6832  103004  010600                                                          MOV     SP,R0
 6833  103006  104417                                                          TRAP    C$PNTF
 6834  103010  062706  000006                                                  ADD     #6,SP
 6835  103014  116302  000030          MOVB    SIECO(R3),R2           ; GET ECO NO. AND PRINT
 6836  103020                         PRINTF  #SIMSG4,R2
 6837  103020  010246                                                          MOV     R2,-(SP)
 6838  103022  012746  066572                                                  MOV     #SIMSG4,-(SP)
 6839  103026  012746  000002                                                  MOV     #2,-(SP)
 6840  103032  010600                                                          MOV     SP,R0
 6841  103034  104417                                                          TRAP    C$PNTF
 6842  103036  062706  000006                                                  ADD     #6,SP
 6843  103042  116302  000031          MOVB    SIUECO(R3),R2          ; GET USER ECO NO. AND PRINT
 6844  103046                         PRINTF  #SIMSG5,R2
 6845  103046  010246                                                          MOV     R2,-(SP)
 6846  103050  012746  066613                                                  MOV     #SIMSG5,-(SP)
 6847  103054  012746  000002                                                  MOV     #2,-(SP)
 6848  103060  010600                                                          MOV     SP,R0
 6849  103062  104417                                                          TRAP    C$PNTF
 6850  103064  062706  000006                                                  ADD     #6,SP
 6851  103070  116302  000035          MOVB    SIFNCT(R3),R2          ; GET FUNCTON CODE AND PRINT
 6852  103074                         PRINTF  #SIMSG6,R2
 6853  103074  010246                                                          MOV     R2,-(SP)
 6854  103076  012746  066641                                                  MOV     #SIMSG6,-(SP)
 6855  103102  012746  000002                                                  MOV     #2,-(SP)
 6856  103106  010600                                                          MOV     SP,R0
 6857  103110  104417                                                          TRAP    C$PNTF
 6858  103112  062706  000006                                                  ADD     #6,SP
 6859  103116  116302  000053          MOVB    SIDEV(R3),R2           ; GET DEVICE TYPE AND PRINT
 6860  103122                         PRINTF  #SIMSG7,R2
 6861  103122  010246                                                          MOV     R2,-(SP)
 6862  103124  012746  066667                                                  MOV     #SIMSG7,-(SP)
 6863  103130  012746  000002                                                  MOV     #2,-(SP)
 6864  103134  010600                                                          MOV     SP,R0
 6865  103136  104417                                                          TRAP    C$PNTF
 6866  103140  062706  000006                                                  ADD     #6,SP
 6867  103144  000411                 BR      51$                    ; EXIT
 6868
 6869  103146                 40$:    ERRDF   23,EMSG23,ERR1         ; ERROR -- CAN'T WRITE MODE
 6870  103146  104455                                                          TRAP    C$ERDF
 6871  103150  000027                                                          .WORD   23
 6872  103152  065321                                                          .WORD   EMSG23
 6873  103154  067460                                                          .WORD   ERR1
 6874  103156  000424                 BR      54$
 6875
 6876  103160                 45$:    ERRDF   24,EMSG24,ERR1         ; ERROR -- CAN'T TRANSMIT PACKETS
 6877  103160  104455                                                          TRAP    C$ERDF
 6878  103162  000030                                                          .WORD   24
 6879  103164  065366                                                          .WORD   EMSG24
```

```
 6880  103166  067460                                                                    .WORD   ERR1
 6881
 6882  103170                        51$:    CALL    WRITES  #1,#ADRBUF      ; UPDATE SUMMARY TABLE
 6883  103206  005061  000002        52$:    CLR     2(R1)                   ; DISABLE TRANSMIT PADDING
 6884  103212                                CALL    FUNCT  #WDMODE
 6885  103224                                P$POP   R2                      ; CHECK FOR ERROR
 6886  103226  001347                        BNE     40$                     ;  IF ONE, REPORT IT
 6887  103230                        54$:    P$POP   R1,R2,R3,R4             ; RESTORE REGISTERS
 6888  103240  000207                55$:    RTS     PC
 6889
 6890
 6891                                ;
 6892                                ;ACTION ROUTINE TO CHECK FOR ADDITION PARAMETER CHANGE INPUTS
 6893                                ;AND PRINT OUT NEW PARAMETER INFO WHEN ALL INPUT ARE PROCESSED
 6894                                ;
 6895
 6896  103242  105714                ACTMSG: TSTB    (R4)                    ;CHECK FOR ADDITIONAL INPUT
 6897  103244  001401                        BEQ     12$                     ;BR IF NO
 6898  103246  000437                        BR      50$                     ;IF YES RETURN
 6899  103250  012737  062500  002310  12$:    MOV     #CMDTY6,KEYWD1
 6900  103256  013701  002370                MOV     P$TYPE,R1               ;GET MESSAGE TYPE ASCII STRING ADDRESS
 6901  103262  006301                        ASL     R1                      ;INTO R1
 6902  103264  062701  003262                ADD     #MSGTAB,R1
 6903  103270                                PRINTF  #MSGPRM                 ;PRINT 'MESSAGE' COMMAND MESSAGE
 6904  103270  012746  054641                                                        MOV     #MSGPRM,-(SP)
 6905  103274  012746  000001                                                        MOV     #1,-(SP)
 6906  103300  010600                                                                MOV     SP,R0
 6907  103302  104417                                                                TRAP    C$PNTF
 6908  103304  062706  000004                                                        ADD     #4,SP
 6909  103310                                PRINTF  #MSG4,(R1),P$SIZE,P$CPYS        ;PRINT MSG PARAMETERS
 6910  103310  013746  002374                                                        MOV     P$CPYS,-(SP)
 6911  103314  013746  002372                                                        MOV     P$SIZE,-(SP)
 6912  103320  011146                                                                MOV     (R1),-(SP)
 6913  103322  012746  055276                                                        MOV     #MSG4,-(SP)
 6914  103326  012746  000004                                                        MOV     #4,-(SP)
 6915  103332  010600                                                                MOV     SP,R0
 6916  103334  104417                                                                TRAP    C$PNTF
 6917  103336  062706  000012                                                        ADD     #12,SP
 6918  103342  105037  003160                CLRB    P$NNUF                  ;CLEAR NOTNUF FLAG
 6919  103346  000207                50$:    RTS     PC
 6920
 6921
 6922                                ;
 6923                                ;ACTION ROUTINE TO RETURN CONTROL TO THE SUPERVISOR
 6924                                ;
 6925
 6926  103350  012737  000020  003672  ACTEXT: MOV     #CEXIT,CFLAG            ;SET EXIT FLAG
 6927  103356  000207                        RTS     PC
 6928
 6929
 6930                                ;
 6931                                ;ACTION ROUTINE TO TAKE NI NODE ADDRESS FROM INPUT STRING BUFFER
 6932                                ;AND STORE IT IN THE BUFFER CALLED ADRBUF
 6933                                ;
 6934
 6935  103360                        ACTXAD: CALL    EDPACK CBOADR,#ADRBUF,#6        ;PUT NODE ADDRESS INTO ADRBUF
```

```
 6936  103402                                        P$POP   P$AERR                  ;SET ADDRESS=12 CHAR. GOOD/BAD FLAG
 6937  103406  105737  003162                        TSTB    P$AERR                  ;IF GOOD, RETURN
 6938  103412  001412                                BEQ     10$
 6939  103414                                        PRINTF  #CADRER                 ;ELSE, PRINT ERROR MESSAGE
 6940  103414  012746  053232                                                                MOV     #CADRER,-(SP)
 6941  103420  012746  000001                                                                MOV     #1,-(SP)
 6942  103424  010600                                                                        MOV     SP,R0
 6943  103426  104417                                                                        TRAP    C$PNTF
 6944  103430  062706  000004                                                                ADD     #4,SP
 6945  103434  105037  003160                        CLRB    P$NNUF                  ; AND CLEAR 'NOT ENOUGH' FLAG
 6946  103440  000207                        10$:    RTS     PC
 6947
 6948                                                ;
 6949                                                ;ACTION ROUTINE TO STORE POINTER TO BEGINING OF OPERATOR INPUT ADDRESS
 6950                                                ;IN COMMAND INPUT BUFFER
 6951                                                ;
 6952
 6953  103442  010437  002366                ACTSR4: MOV     R4,CBOADR               ;SAVE STRING POINTER
 6954  103446  000207                        10$:    RTS     PC
 6955
 6956
 6957                                                ;
 6958                                                ;ACTION ROUTINE TO SET MESSAGE TYPE = ALPHA FLAG
 6959                                                ;
 6960
 6961  103450  012737  000000  002370        ACTALP: MOV     #ALPHA,P$TYPE           ;SET MESSAGE TYPE
 6962  103456  000207                                RTS     PC
 6963
 6964
 6965                                                ;
 6966                                                ;ACTION ROUTINE TO SET MESSAGE TYPE = ALL ONES FLAG
 6967                                                ;
 6968
 6969  103460  012737  000001  002370        ACTONE: MOV     #ONES,P$TYPE            ;SET MESSAGE TYPE
 6970  103466  000207                                RTS     PC
 6971
 6972
 6973                                                ;
 6974                                                ;ACTION ROUTINE TO SET MESSAGE TYPE = ALL ZEROS FLAG
 6975                                                ;
 6976
 6977
 6978  103470  012737  000002  002370        ACTZRO: MOV     #ZEROS,P$TYPE           ;SET MESSAGE TYPE
 6979  103476  000207                                RTS     PC
 6980
 6981
 6982                                                ;
 6983                                                ;ACTION ROUTINE TO SET MESSAGE TYPE = ALTERNATING ONES FLAG
 6984                                                ;
 6985
 6986  103500  012737  000003  002370        ACT1AL: MOV     #ONEALT,P$TYPE          ;SET MESSAGE TYPE
 6987  103506  000207                                RTS     PC
 6988
 6989
 6990                                                ;
 6991                                                ;ACTION ROUTINE TO SET MESSAGE TYPE = ALTERNATING ZEROS FLAG
```

```
                                        ;
                                        ;
103510  012737  000004  002370  ACTOAL: MOV     #ZROALT,P$TYPE          ;SET MESSAGE TYPE
103516  000207                          RTS     PC
                                        ;
                                        ;
                                        ;ACTION ROUTINE TO SET MESSAGE TYPE = CCITT FLAG
                                        ;
103520  012737  000005  002370  ACTCTT: MOV     #CCITT,P$TYPE           ;SET MESSAGE TYPE
103526  000207                          RTS     PC
                                        ;
                                        ;
                                        ;ACTION ROUTINE TO SET MESSAGE TYPE = OPERATOR SELECTED INPUT
                                        ;
103530  105037  003163          ACTOPR: CLRB    P$MERR                  ;CLEAR MESSAGE ERROR FLAG
103534  004737  076330                  JSR     PC,TRVADR               ;PARSE THROUGH INPUT STRING
103540  105737  003161                  TSTB    P$GDBD                  ;TEST GOOD/BAD FLAG
103544  001403                          BEQ     10$                     ;IF GOOD, BR 10$
103546  105037  003161                  CLRB    P$GDBD                  ;CLEAR FLAG
103552  000415                          BR      20$                     ;SET CTARGT FLAG AND RETURN
103554  022737  000006  003672  10$:    CMP     #OPRSEL,CFLAG           ;CHECK TO SEE IF STRING VALID
103562  001011                          BNE     20$                     ;IF NOT OK, RETURN WITH ERROR FLAG SET
103564  012737  000006  002370          MOV     #OPRSEL,P$TYPE          ;SET MESSAGE TYPE
103572                                  CALL    SELMSG CBOADR           ;PUT OPERATOR SELECTED STRING INTO BUFFER
103604  000423                          BR      50$                     ;RETURN
103606  022737  000000  003672  20$:    CMP     #CTARGT,CFLAG           ;SEE IF CTARGT FLAG SET, IF YES ERROR
103614  001011                          BNE     30$                     ;IF NOT SET, BR 30$
103616                                  PRINTF  #UNBOND                 ;ELSE PRINT UNBOUNDED INPUT STRING ERR. MSG.
103616  012746  054252                                                          MOV     #UNBOND,-(SP)
103622  012746  000001                                                          MOV     #1,-(SP)
103626  010600                                                                  MOV     SP,R0
103630  104417                                                                  TRAP    C$PNTF
103632  062706  000004                                                          ADD     #4,SP
103636  000406                          BR      50$                     ;RETURN
103640  105737  003163          30$:    TSTB    P$MERR                  ;IF P$MERR FLAG SET, UNBOUNDED STRING
103644  001003                          BNE     50$                     ;WAS ENTERED, ERROR ALREADY HANDLED
103646  112737  177777  003161          MOVB    #-1,P$GDBD              ;SET ERROR FLAG AND RETURN
103654  000207          50$:    RTS     PC                              ;RETURN
                                        ;
                                        ;
                                        ;ACTION ROUTINE TO CHECK FOR MORE INPUT AFTER MESSAGE TYPE HAS BEEN
                                        ;ALTERED
                                        ;
103656  004737  103242          ACTTYP: JSR     PC,ACTMSG               ;CHECK FOR ADDITIONAL COMMANDS
103662  000207                          RTS     PC
                                        ;
                                        ;
                                        ;ACTION ROUTINE TO INPUT MESSAGE SIZE PARAMETER, CHECK TO SEE IF
                                        ;IT IS WITHIN LEGAL LIMITS, CHANGE PARAMETER AND THEN RETURN TO
```

```
 7048                                   ;SEE IF MORE INPUT EXISTA
 7049                                   ;
 7050
 7051  103664  023727  003154  000037  ACTSZE: CMP     P$NUM,#31.              ;CHECK FOR VALID SIZE RANGE
 7052  103672  003410                          BLE     10$
 7053  103674  022737  002673  003154          CMP     #1467.,P$NUM
 7054  103702  003404                          BLE     10$                     ;IF VALID CONTINUE
 7055  103704  013737  003154  002372          MOV     P$NUM,P$SIZE            ;SET MESSAGE SIZE
 7056  103712  000410                          BR      20$
 7057  103714                         10$:     PRINTF  #SIZLMT                 ;PRINT SIZE LIMITS EXCEEDED MESSAGE
 7058  103714  012746  054061                                                          MOV     #SIZLMT,-(SP)
 7059  103720  012746  000001                                                          MOV     #1,-(SP)
 7060  103724  010600                                                                 MOV     SP,R0
 7061  103726  104417                                                                 TRAP    C$PNTF
 7062  103730  062706  000004                                                          ADD     #4,SP
 7063  103734  004737  103242          20$:     JSR     PC,ACTMSG               ;CHECK FOR ADDITIONAL COMMANDS
 7064  103740  000207                          RTS     PC
 7065
 7066
 7067                                   ;
 7068                                   ;ACTION ROUTINE TO INPUT COPIES PARAMETER, CHECK TO SEE IF IT IS
 7069                                   ;WITHIN LEGAL LIMITS, CHANGE PARAMETER AND THEN RETURN TO SEE IF
 7070                                   ;MORE INPUT PARAMETERS EXIST
 7071                                   ;
 7072
 7073  103742  023727  003154  000000  ACTCPY: CMP     P$NUM,#0                ;CHECK FOR VALID COPIES RANGE
 7074  103750  003410                          BLE     10$
 7075  103752  022737  000400  003154          CMP     #256.,P$NUM
 7076  103760  003404                          BLE     10$                     ;IF VALID, CONTINUE
 7077  103762  013737  003154  002374          MOV     P$NUM,P$CPYS            ;SET MESSAGE COPIES
 7078  103770  000410                          BR      20$
 7079  103772                         10$:     PRINTF  #CPYLMT                 ;PRINT COPY LIMIT EXCEEDED MESSAGE
 7080  103772  012746  053775                                                          MOV     #CPYLMT,-(SP)
 7081  103776  012746  000001                                                          MOV     #1,-(SP)
 7082  104002  010600                                                                 MOV     SP,R0
 7083  104004  104417                                                                 TRAP    C$PNTF
 7084  104006  062706  000004                                                          ADD     #4,SP
 7085  104012  004737  103242          20$:     JSR     PC,ACTMSG               ;CHECK FOR ADDITIONAL COMMANDS
 7086  104016  000207                          RTS     PC
 7087
 7088
 7089                                   ;
 7090                                   ;ACTION ROUTINE TO CLEAR NODE SPECIFIED BY PHYSICAL ADDRESS FROM NODE TABLE
 7091                                   ;
 7092
 7093  104020  105037  003160          ACTNAD: CLRB    P$NNUF                  ;CLEAR NOTNUF FLAG
 7094  104024  105737  003162                  TSTB    P$AERR                  ;SEE IF ADDRESS ENTERED WAS VALID
 7095  104030  001051                          BNE     35$                     ; IF NOT, EXIT ACTION ROUTINE
 7096  104032                                  P$PUSH  R2,R3                   ;SAVE R2 AND R3
 7097  104036  012702  002314                  MOV     #ADRBUF,R2              ;MOVE ADDRESS OF ADDRESS INTO R2
 7098  104042  012703  002404                  MOV     #NODTBL,R3              ;MOVE ADDRESS OF NODE TABLE INTO R3
 7099  104046                         21$:     CALL    CMPADR R2,R3            ;SEE IF ADDRESSES MATCH
 7100  104060                                  P$POP   R1
 7101  104062  001416                          BEQ     25$                     ;IF YES, BR 25$
 7102  104064  062703  000010                  ADD     #10,R3                  ;ELSE POINT R3 TO NEXT ENTRY
 7103  104070  022713  177777                  CMP     #-1,(R3)                ;SEE IF END OF TABLE
```

```
 7104  104074  001364                        BNE     21$                     ;IF NOT, COMPARE NEXT ENTRY
 7105  104076                                PRINTF  #NOCMPR                 ;ELSE, PRINT ADDRESS DOESN'T COMPARE MSG.
 7106  104076  012746  054160                                                        MOV     #NOCMPR,-(SP)
 7107  104102  012746  000001                                                        MOV     #1,-(SP)
 7108  104106  010600                                                              MOV     SP,R0
 7109  104110  104417                                                              TRAP    C$PNTF
 7110  104112  062706  000004                                                      ADD     #4,SP
 7111  104116  000414                        BR      30$                     ;RETURN
 7112  104120  005023                 25$:   CLR     (R3)+                   ;ELSE, CLEAR NODE FROM TABLE
 7113  104122  005023                        CLR     (R3)+
 7114  104124  005023                        CLR     (R3)+
 7115  104126  005013                        CLR     (R3)
 7116  104130                                PRINTF  #ADRDEL                 ;PRINT NODE DELETED FROM TABLE MESSAGE
 7117  104130  012746  054340                                                        MOV     #ADRDEL,-(SP)
 7118  104134  012746  000001                                                        MOV     #1,-(SP)
 7119  104140  010600                                                              MOV     SP,R0
 7120  104142  104417                                                              TRAP    C$PNTF
 7121  104144  062706  000004                                                      ADD     #4,SP
 7122  104150                         30$:   P$POP   R2,R3                   ;RESTORE R2 AND R3
 7123  104154  000207                 35$:   RTS     PC                      ;RETURN
 7124
 7125
 7126                                        ;
 7127                                        ;ACTION ROUTINE TO CLEAR NODE TABLE
 7128                                        ;
 7129
 7130  104156                         ACTNAL: P$PUSH  R2,R3                  ;SAVE R2,R3
 7131  104162  012703  000050                MOV     #TBLLEN,R3              ;SET INCR. COUNTER TO 40
 7132  104166  012702  002404                MOV     #NODTBL,R2              ;MOVE NODE TABLE ADDRESS INTO R2
 7133  104172  005022                 10$:   CLR     (R2)+                   ;CLEAR BYTE IN NODE TABEL
 7134  104174  005303                        DEC     R3                      ;DECRIMENT COUNTER
 7135  104176  001375                        BNE     10$                     ;CONTINUE UNTIL DONE
 7136  104200                                PRINTF  #TABCLR,#NOD            ;PRINT NODE TABLE CLEARED MESSAGE
 7137  104200  012746  053701                                                        MOV     #NOD,-(SP)
 7138  104204  012746  054514                                                        MOV     #TABCLR,-(SP)
 7139  104210  012746  000002                                                        MOV     #2,-(SP)
 7140  104214  010600                                                              MOV     SP,R0
 7141  104216  104417                                                              TRAP    C$PNTF
 7142  104220  062706  000006                                                      ADD     #6,SP
 7143  104224  105037  003160                        CLRB    P$NNUF                  ;CLEAR NOTNUF FLAG
 7144  104230                                P$POP   R2,R3                   ;RESTORE R2 AND R3
 7145  104234  000207                        RTS     PC
 7146
 7147
 7148                                        ;
 7149                                        ;ACTION ROUTINE TO RUN SPECIFIED TEST
 7150                                        ;
 7151
 7152  104236  105037  003160         ACTRUN: CLRB    P$NNUF                  ; CLEAR 'NOT ENOUGH' FLAG
 7153  104242  013737  003154  002376        MOV     P$NUM,P$PASS
 7154  104250  022737  000032  002310  5$:   CMP     #CRNALL,KEYWD1          ; SEE IF 'ALL' TEST
 7155  104256  001004                        BNE     10$                     ;  IF NO, CONTINUE
 7156  104260                                CALL    RUNALL                  ;  IF YES, DO ALLNODE
 7157  104266  000423                        BR      30$
 7158  104270  022737  000033  002310  10$:  CMP     #CLUPPR,KEYWD1          ; IS IT 'LOOPPAIR' TEST
 7159  104276  001004                        BNE     15$                     ;  IF NO, CONTINUE
```

```
7160  104300                                    CALL    RUNLUP                  ;  IF YES, DO LOOPPAIR
7161  104306  000413                            BR      30$
7162  104310  022737  000043  002310  15$:     CMP     #CDIR,KEYWD1            ; IS IT 'DIRECT' TEST
7163  104316  001004                            BNE     20$                     ;  IF NO, CONTINUE
7164  104320                                    CALL    RUNDIR                  ;  IF YES, DO DIRECT
7165  104326  000403                            BR      30$
7166  104330                            20$:    CALL    RUNPAT                  ; ELSE, ITS 'PATTERN' TEST
7167  104336  023727  002376  177777  30$:     CMP     P$PASS,#-1              ; SEE IF PASS SET FOR INDEFINATE
7168  104344  001741                            BEQ     5$                      ;  IF YES, LOOP
7169  104346  005337  002376                    DEC     P$PASS                  ; HAVE WE DONE ALL PASSES?
7170  104352  001336                            BNE     5$                      ;  IF NO, LOOP
7171  104354  000207                            RTS     PC
7172
7173                                    ;
7174                                    ;ACTION ROUTINE TO SET 'RUN ALL' FLAG
7175                                    ;
7176
7177  104356  012737  000032  002310  ACTRNA: MOV     #CRNALL,KEYWD1          ; SET FLAG
7178  104364  000207                            RTS     PC
7179
7180  104366                            RUNALL: CALL    DIRCOM                  ; RUN LOOPDIRECT TEST
7181  104374                                    P$POP   R1                      ;  CHECK RESULTS
7182  104376  001415                            BEQ     5$                      ;  IF OK, BRANCH
7183  104400  022701  000001                    CMP     #1,R1                   ;   ELSE, WAS TABLE EMPTY?
7184  104404  001410                            BEQ     3$                      ;   IF YES, DON'T PRINT ABORT MESSAGE
7185  104406                                    PRINTS  #PASABT                 ;  ELSE ABORT TEST AND PRINT MESSAGE
7186  104406  012746  061526                                                            MOV     #PASABT,-(SP)
7187  104412  012746  000001                                                            MOV     #1,-(SP)
7188  104416  010600                                                                    MOV     SP,R0
7189  104420  104416                                                                    TRAP    C$PNTS
7190  104422  062706  000004                                                            ADD     #4,SP
7191  104426  000137  104770            3$:     JMP     32$
7192  104432  012737  002404  002402  5$:     MOV     #NODTBL,SLOT            ; MOVE NODE TABLE ADDRESS TO SLOT
7193  104440                                    CALL    FULSLT                  ; FIND FIRST ENTRY
7194  104446  013701  002402                    MOV     SLOT,R1                 ;  AND PUT TARGET ADDRESS INTO R1
7195  104452  013737  002374  050562  10$:     MOV     P$CPYS,CPYCNT           ; SET UP LOOP FOR NO. OF COPIES
7196  104460  062737  000010  002402            ADD     #10,SLOT                ; UPDATE SLOT
7197  104466  013702  002402                    MOV     SLOT,R2
7198  104472                                    CALL    FULSLT                  ; GET NEXT ASSIST NODE FROM TABLE
7199  104500  022737  177777  002402            CMP     #-1,SLOT                ;  SEE IF AT END OF TABLE
7200  104506  001515                            BEQ     25$                     ;  IF YES, BR
7201  104510                            15$:    CALL    BLDFAS  R1,SLOT         ; BUILD FULL ASSIST MESSAGE
7202  104524                                    CALL    XMIT                    ; TRANSMIT MESSAGE
7203  104532                                    P$POP   R3                      ;  CHECK RESULTS
7204  104534  001404                            BEQ     17$                     ;  IF OK, CONTINUE
7205  104536                                    ERRHRD  37,EMSG24               ;  PRINT ERROR MESSAGE
7206  104536  104456                                                                    TRAP    C$ERHRD
7207  104540  000045                                                                    .WORD   37
7208  104542  065366                                                                    .WORD   EMSG24
7209  104544  000000                                                                    .WORD   0
7210  104546                            17$:    CALL    BINHEX  R1,#6,#STRBUF   ;  PRINT ERROR MESSAGE
7211  104566                                    CALL    BINHEX  R2,#6,#STRBU1   ;
7212  104606                                    PRINTB  #TSTMS4,#ARGTY7,#STRBUF,#ARGTY6,#STRBU1 ;   ASSIST NODE =
7213  104606  012746  002344                                                            MOV     #STRBU1,-(SP)
7214  104612  012746  062603                                                            MOV     #ARGTY6,-(SP)
7215  104616  012746  002322                                                            MOV     #STRBUF,-(SP)
```

```
7216  104622  012746  062613                                                  MOV     #ARGTY7,-(SP)
7217  104626  012746  061624                                                  MOV     #TSTMS4,-(SP)
7218  104632  012746  000005                                                  MOV     #5,-(SP)
7219  104636  010600                                                          MOV     SP,R0
7220  104640  104414                                                          TRAP    C$PNTB
7221  104642  062706  000014                                                  ADD     #14,SP
7222  104646                        CALL    RUNCOM          ; DO RECEIVE LOOP
7223  104654                        P$POP   R4              ; CHECK RESULTS
7224  104656  001405                BEQ     21$             ;  IF OK, LOOP SOME MORE
7225  104660                20$:    ERRHRD  28,EMSG42,ERR3  ;  ELSE PRINT ERROR MESSAGE
7226  104660  104456                                                          TRAP    C$ERHRD
7227  104662  000034                                                          .WORD   28
7228  104664  066353                                                          .WORD   EMSG42
7229  104666  067604                                                          .WORD   ERR3
7230  104670  000410                BR      101$
7231  104672                21$:    PRINTB  #OKFU
7232  104672  012746  062013                                                  MOV     #OKFU,-(SP)
7233  104676  012746  000001                                                  MOV     #1,-(SP)
7234  104702  010600                                                          MOV     SP,R0
7235  104704  104414                                                          TRAP    C$PNTB
7236  104706  062706  000004                                                  ADD     #4,SP
7237  104712  005337  050562        101$:   DEC     CPYCNT          ; DECREMENT 'COPIES' COUNTER
7238  104716  001274                BNE     15$             ;  IF MORE TO DO, LOOP
7239  104720                        CALL    WRITES #2,R1,SLOT       ; ELSE, UPDATE SUMMARY TABLE
7240  104740  000644                BR      10$
7241  104742  062701  000010        25$:    ADD     #10,R1          ; POINT R1 TO NEXT TARGET NODE
7242  104746  010137  002402                MOV     R1,SLOT         ; UPDATE SLOT
7243  104752                        CALL    FULSLT          ; GET ADDRESS FROM TABLE
7244  104760  022737  177777  002402        CMP     #-1,SLOT        ; SEE IF END OF TABLE
7245  104766  001231                BNE     10$             ;  IF NO, CONTINUE ELSE, FINISHED
7246  104770                32$:    RETURN
7247
7248                                ;
7249                                ;ACTION ROUTINE TO SET 'RUN LOOP DIRECT' FLAG
7250                                ;
7251
7252  104772  012737  000043  002310  ACTDIR: MOV     #CDIR,KEYWD1    ; SET FLAG
7253  105000  000207                RTS     PC
7254
7255  105002                        RUNDIR: CALL    DIRCOM          ; CALL COMMON CODE
7256  105010                        P$POP   R1
7257  105012  022701  000001        CMP     #1,R1           ; WAS TABLE EMPTY?
7258  105016  001400                BEQ     10$             ;  IF YES, DON'T PRINT
7259  105020                10$:    RETURN
7260
7261  105022  005001                DIRCOM: CLR     R1              ; CLEAR RESULTS REGISTER
7262  105024  012737  002404  002402        MOV     #NODTBL,SLOT    ; MOVE NODE TABLE ADDRESS TO SLOT
7263  105032                        CALL    FULSLT          ; SEE IF TABLE EMPTY
7264  105040  022737  177777  002402        CMP     #-1,SLOT
7265  105046  001015                BNE     9$              ;  IF NO CONTINUE
7266  105050                        PRINTF  #TABEMT,#NOD    ;  ELSE, PRINT "TABLE EMPTY" MESSAGE
7267  105050  012746  053701                                                  MOV     #NOD,-(SP)
7268  105054  012746  053632                                                  MOV     #TABEMT,-(SP)
7269  105060  012746  000002                                                  MOV     #2,-(SP)
7270  105064  010600                                                          MOV     SP,R0
7271  105066  104417                                                          TRAP    C$PNTF
```

```
105070 062706 000006                                                                          ADD     #6,SP
105074 012701 000001                  MOV     #1,R1                     ;  PUT 'TABLE EMPTY' INDICATOR IN R1
105100 000545                         BR      32$
105102 012737 002404 002402   9$:     MOV     #NODTBL,SLOT
105110 013737 002374 050562   10$:    MOV     P$CPYS,CPYCNT             ; SET UP FOR NO. OF COPIES
105116                                CALL    FULSLT                    ; GET NEXT NODE IN TABLE
105124 022737 177777 002402           CMP     #-1,SLOT                  ; SEE IF AT END OF TABLE
105132 001530                         BEQ     32$                       ;  IF YES, EXIT
105134                                CALL    BINHEX  SLOT,#6,#STRBUF   ; PRINT ADDRESS BEING TESTED
105156                        15$:    PRINTB  #TSTMS2,#DIRECT,#STRBUF   ;  NODE ADDRESS
105156 012746 002322                                                                          MOV     #STRBUF,-(SP)
105162 012746 062173                                                                          MOV     #DIRECT,-(SP)
105166 012746 061571                                                                          MOV     #TSTMS2,-(SP)
105172 012746 000003                                                                          MOV     #3,-(SP)
105176 010600                                                                                 MOV     SP,R0
105200 104414                                                                                 TRAP    C$PNTB
105202 062706 000010                                                                          ADD     #10,SP
105206 022737 000005 002310           CMP     #CPATRN,KEYWD1
105214 001016                         BNE     16$
105216 013701 002370                  MOV     P$TYPE,R1
105222 006301                         ASL     R1
105224 062701 003262                  ADD     #MSGTAB,R1
105230                                PRINTB  #MESPA1,(R1)
105230 011146                                                                                 MOV     (R1),-(SP)
105232 012746 062125                                                                          MOV     #MESPA1,-(SP)
105236 012746 000002                                                                          MOV     #2,-(SP)
105242 010600                                                                                 MOV     SP,R0
105244 104414                                                                                 TRAP    C$PNTB
105246 062706 000006                                                                          ADD     #6,SP
105252                        16$:    CALL    BLDLD   SLOT              ; CALL BUILD LOOPDIRECT SUBROUTINE
105264                                CALL    XMIT                      ; TRANSMIT LOOPDIRECT MESSAGES
105272                                P$POP   R2                        ; GET RESULTS, R2 = SUCCESS/FAILURE
105274 001405                         BEQ     26$                       ; IF OK, EXIT
105276                        25$:    ERRHRD  26,EMSG24                 ;  ELSE PRINT ERROR MESSAGE
105276 104456                                                                                 TRAP    C$ERHRD
105300 000032                                                                                 .WORD   26
105302 065366                                                                                 .WORD   EMSG24
105304 000000                                                                                 .WORD   0
105306 000700                         BR      10$
105310                        26$:    CALL    RUNCOM                    ; DO RECIEVE LOOP
105316                                P$POP   R4                        ; GET RESULTS
105320 001407                         BEQ     29$                       ;  IF NO ERRORS, CONTINUE
105322                                ERRHRD  27,EMSG34,ERR2
105322 104456                                                                                 TRAP    C$ERHRD
105324 000033                                                                                 .WORD   27
105326 065732                                                                                 .WORD   EMSG34
105330 067516                                                                                 .WORD   ERR2
105332 012701 177777                  MOV     #-1,P1                    ; PUT ERROR INDICATOR INTO RI
105336 000410                         BR      101$
105340                        29$:    PRINTB  #OK                       ; RESPONSE OK
105340 012746 061661                                                                          MOV     #OK,-(SP)
105344 012746 000001                                                                          MOV     #1,-(SP)
105350 010600                                                                                 MOV     SP,R0
105352 104414                                                                                 TRAP    C$PNTB
105354 062706 000004                                                                          ADD     #4,SP
105360 005337 050562          101$:   DEC     CPYCNT                    ; DECREMENT 'COPIES' COUNTER
```

```
 7328  105364  001274                          BNE     15$                     ;  IF MORE TO DO, LOOP
 7329  105366                                  CALL    WRITES #1,SLOT          ; ELSE,UPDATE SUMMARY TABLE
 7330  105404  062737  000010  002402  30$:    ADD     #10,SLOT                ; INCREMENT TO NEXT NODE TABLE ENTRY
 7331  105412  000636                          BR      10$
 7332  105414                          32$:    RETURN  R1
 7333
 7334
 7335                                  ;
 7336                                  ;ACTION ROUTINE TO SET 'RUN LOOPPAIR' FLAG
 7337                                  ;
 7338
 7339  105420  012737  000033  002310  ACTRNL: MOV     #CLUPPR,KEYWD1          ; SET FLAG
 7340  105426  000207                          RTS     PC
 7341
 7342  105430  005037  050552          RUNLUP: CLR     TEMP1                   ; CLEAR 'HEADER PRINTED' FLAG
 7343  105434  012737  002404  002402          MOV     #NODTBL,SLOT            ; MOVE NODE TABLE ADDRESS TO SLOT
 7344  105442                                  CALL    FULSLT                  ; SEE IF TABLE EMPTY
 7345  105450  022737  177777  002402          CMP     #-1,SLOT                ;
 7346  105456  001014                          BNE     9$                      ;  IF NO, CONTINUE
 7347  105460                                  PRINTF  #TABEMT,#NOD            ;  ELSE, PRINT ''TABLE EMPTY'' MESSAGE
 7348  105460  012746  053701                                                          MOV     #NOD,-(SP)
 7349  105464  012746  053632                                                          MOV     #TABEMT,-(SP)
 7350  105470  012746  000002                                                          MOV     #2,-(SP)
 7351  105474  010600                                                                  MOV     SP,R0
 7352  105476  104417                                                                  TRAP    C$PNTF
 7353  105500  062706  000006                                                          ADD     #6,SP
 7354  105504  000137  106514                  JMP     30$
 7355  105510  012737  002404  002402  9$:     MOV     #NODTBL,SLOT            ; MOVE NODE TABLE ADDRESS TO SLOT
 7356  105516  013737  002374  050562  10$:    MOV     P$CPYS,CPYCNT           ; SET UP FOR NO. OF COPIES
 7357  105524                                  CALL    FULSLT                  ; GET NEXT NODE IN TABLE
 7358  105532  022737  177777  002402          CMP     #-1,SLOT                ; SEE IF AT END OF TABLE
 7359  105540  001002                          BNE     11$                     ;  IF YES, EXIT
 7360  105542  000137  106514                  JMP     30$
 7361  105546  013701  002402          11$:    MOV     SLOT,R1                 ; MOVE SLOT TO R1
 7362  105552  126127  000007  000000          CMPB    7(R1),#CTARGT           ; SEE IF TARGET NODE
 7363  105560  001431                          BEQ     18$                     ;  IF YES, BRANCH
 7364  105562  010102                          MOV     R1,R2                   ;  ELSE, POINT R1 TO TARGET NODE
 7365  105564  062701  000010                  ADD     #10,R1                  ;   AND R2 TO ASSIST NODE
 7366                                  ;       BR      19$
 7367  105570  126127  000007  000000  15$:    CMPB    7(R1),#CTARGT           ; SEE IF TARGET NODE
 7368  105576  001425                          BEQ     19$                     ;  IF YES, OK
 7369  105600                          17$:    PRINTF  #EMSG32                 ;  ELSE PRINT ERROR MESSAGE
 7370  105600  012746  065637                                                          MOV     #EMSG32,-(SP)
 7371  105604  012746  000001                                                          MOV     #1,-(SP)
 7372  105610  010600                                                                  MOV     SP,R0
 7373  105612  104417                                                                  TRAP    C$PNTF
 7374  105614  062706  000004                                                          ADD     #4,SP
 7375  105620                                  PRINTF  #PASABT
 7376  105620  012746  061526                                                          MOV     #PASABT,-(SP)
 7377  105624  012746  000001                                                          MOV     #1,-(SP)
 7378  105630  010600                                                                  MOV     SP,R0
 7379  105632  104417                                                                  TRAP    C$PNTF
 7380  105634  062706  000004                                                          ADD     #4,SP
 7381  105640  000137  106514                  JMP     30$                     ;   EXIT
 7382  105644  010102                  18$:    MOV     R1,R2                   ; POINT R1 TO TARGET NODE
 7383  105646  062702  000010                  ADD     #10,R2                  ;  AND R2 TO ASSIST NODE
```

```
 7384  105652  126227  000007  000001  19$:    CMPB    7(R2),#CASIST           ; IS R2 POINTING TO AN ASSIST NODE?
 7385  105660  001347                          BNE     17$                     ;  IF NOT, ERROR, ELSE CONTINUE
 7386  105662                          20$:    CALL    BLDAST  R2,R1           ; BUILD TRANSMIT ASSIST MESSAGE
 7387  105674                                  CALL    XMIT                    ; TRANSMIT MESSAGE
 7388  105702                                  P$POP   R4                      ; GET RESULTS, R2 = SUCCESS/FAILURE
 7389  105704  0C1406                          BEQ     22$                     ; IF OK, EXIT
 7390  105706                          21$:    ERRHRD  26,EMSG24               ;  ELSE PRINT ERROR MESSAGE
 7391  105706  104456                                                                  TRAP    C$ERHRD
 7392  105710  000032                                                                  .WORD   26
 7393  105712  065366                                                                  .WORD   EMSG24
 7394  105714  000000                                                                  .WORD   0
 7395  105716  000137  106432                  JMP     28$
 7396  105722                          22$:    CALL    BINHEX  R1,#6,#STRBUF   ;  PRINT ERROR MESSAGE
 7397  105742                                  CALL    BINHEX  R2,#6,#STRBU1   ;
 7398  105762                                  PRINTB  #TSTMS4,#ARGTY7,#STRBUF,#ARGTY6,#STRBU1 ;   ASSIST NODE =
 7399  105762  012746  002344                                                          MOV     #STRBU1,-(SP)
 7400  105766  012746  062603                                                          MOV     #ARGTY6,-(SP)
 7401  105772  012746  002322                                                          MOV     #STRBUF,-(SP)
 7402  105776  012746  062613                                                          MOV     #ARGTY7,-(SP)
 7403  106002  012746  061624                                                          MOV     #TSTMS4,-(SP)
 7404  106006  012746  000005                                                          MOV     #5,-(SP)
 7405  106012  010600                                                                  MOV     SP,R0
 7406  106014  104414                                                                  TRAP    C$PNTB
 7407  106016  062706  000014                                                          ADD     #14,SP
 7408  106022                                  CALL    RUNCOM                  ; DO RECIEVE LOOP
 7409  106030                                  P$POP   R3                      ; CHECK RESULTS
 7410  106032  001405                          BEQ     23$                     ;  IF OK, CONT.
 7411  106034                                  ERRHRD  28,EMSG40,ERR3
 7412  106034  104456                                                                  TRAP    C$ERHRD
 7413  106036  000034                                                                  .WORD   28
 7414  106040  066174                                                                  .WORD   EMSG40
 7415  106042  067604                                                                  .WORD   ERR3
 7416  106044  000410                          BR      101$
 7417  106046                          23$:    PRINTB  #OKRE
 7418  106046  012746  061702                                                          MOV     #OKRE,-(SP)
 7419  106052  012746  000001                                                          MOV     #1,-(SP)
 7420  106056  010600                                                                  MOV     SP,R0
 7421  106060  104414                                                                  TRAP    C$PNTB
 7422  106062  062706  000004                                                          ADD     #4,SP
 7423  106066                          101$:   CALL    BLDAST  R1,R2           ; BUILD RECEIVE ASSIST MESSAGE
 7424  106100                                  CALL    XMIT                    ; TRANSMIT MESSAGE
 7425  106106                                  P$POP   R4                      ; CHECK RESULTS
 7426  106110  001276                          BNE     21$                     ; IF OK CONTINUE, ELSE REPORT ERROR
 7427  106112                                  CALL    BINHEX  R1,#6,#STRBUF   ;  PRINT ERROR MESSAGE
 7428  106132                                  CALL    BINHEX  R2,#6,#STRBU1   ;
 7429  106152                                  PRINTB  #TSTMS4,#ARGTY7,#STRBUF,#ARGTY6,#STRBU1 ;   ASSIST NODE =
 7430  106152  012746  002344                                                          MOV     #STRBU1,-(SP)
 7431  106156  012746  062603                                                          MOV     #ARGTY6,-(SP)
 7432  106162  012746  002322                                                          MOV     #STRBUF,-(SP)
 7433  106166  012746  062613                                                          MOV     #ARGTY7,-(SP)
 7434  106172  012746  061624                                                          MOV     #TSTMS4,-(SP)
 7435  106176  012746  000005                                                          MOV     #5,-(SP)
 7436  106202  010600                                                                  MOV     SP,R0
 7437  106204  104414                                                                  TRAP    C$PNTB
 7438  106206  062706  000014                                                          ADD     #14,SP
 7439  106212                                  CALL    RUNCOM                  ; DO RECEIVE LOOP
```

```
                        P$POP   R3                      ;  GET RESULTS
                        BEQ     25$                     ;  IF OK, CONT.
                        ERRHRD  28,EMSG41,ERR3
                                                                TRAP    C$ERHRD
                                                                .WORD   28
                                                                .WORD   EMSG41
                                                                .WORD   ERR3
                        BR      102$
                25$:    PRINTB  #OKTR
                                                                MOV     #OKTR,-(SP)
                                                                MOV     #1,-(SP)
                                                                MOV     SP,R0
                                                                TRAP    C$PNTB
                                                                ADD     #4,SP
                102$:   CALL    BLDFAS  R1,R2           ; BUILD FULL ASSIST MESSAGE
                        CALL    XMIT                    ; TRANSMIT MESSAGE
                        P$POP   R4                      ; CHECK RESULTS
                        BEQ     26$                     ; IF OK CONTINUE, ELSE REPORT ERROR
                        JMP     21$
                26$:    CALL    BINHEX  R1,#6,#STRBUF   ;  PRINT ERROR MESSAGE
                        CALL    BINHEX  R2,#6,#STRBU1   ;
                        PRINTB  #TSTMS4,#ARGTY7,#STRBUF,#ARGTY6,#STRBU1 ;   ASSIST NODE =
                                                                MOV     #STRBU1,-(SP)
                                                                MOV     #ARGTY6,-(SP)
                                                                MOV     #STRBUF,-(SP)
                                                                MOV     #ARGTY7,-(SP)
                                                                MOV     #TSTMS4,-(SP)
                                                                MOV     #5,-(SP)
                                                                MOV     SP,R0
                                                                TRAP    C$PNTB
                                                                ADD     #14,SP
                        CALL    RUNCOM                  ; DO RECEIVE LOOP
                        P$POP   R3                      ; CHECK RESULTS
                        BEQ     28$                     ;  IF NO ERRORS, CONT
                        ERRHRD  28,EMSG42,ERR3
                                                                TRAP    C$ERHRD
                                                                .WORD   28
                                                                .WORD   EMSG42
                                                                .WORD   ERR3
                        BR      103$
                28$:    PRINTB  #OKFU
                                                                MOV     #OKFU,-(SP)
                                                                MOV     #1,-(SP)
                                                                MOV     SP,R0
                                                                TRAP    C$PNTB
                                                                ADD     #4,SP
                103$:   DEC     CPYCNT                  ; DECREMENT 'COPIES' COUNTER
                        BEQ     29$                     ;  IF MORE TO DO, LOOP
                        JMP     20$
                29$:    CALL    WRITES #2,R1,R2         ; ELSE,UPDATE SUMMARY TABLE
                        ADD     #20,SLOT        002402  ; INCREMENT TO NEXT NODE TABLE ENTRY
                        JMP     10$
                30$:    RETURN

                RUNCOM: TST     RETRYS                  ; SEE IF FAILED DUE TO EXCESSIVE COLLISIONS
                        BNE     38$                     ;  IF YES, BR, ELSE CONT.
```

```
7496  106524  012704  003720            MOV     #TIMERS,R4              ; SET UP FOR 10 SECOND TIMOUT
7497  106530  012714  000012            MOV     #10.,(R4)
7498  106534  005002                    CLR     R2                      ; CLEAR RESULTS REGISTER
7499  106536                    35$:    BREAK
7500  106536  104422                                                            TRAP    C$BRK
7501  106540  005714                    TST     (R4)                    ; SEE IF TIME HAS EXPIRED
7502  106542  001516                    BEQ     40$                     ;  IF YES, BRANCH
7503  106544                            CALL    RECEVE                  ; CHECK FOR ANSWER
7504  106552                            P$POP   R1                      ;  R2 HOLDS NO. OF BUFFERS RECEIVED
7505  106554  001770                    BEQ     35$                     ;  IF NO BUFFERS RECIEVED, LOOP
7506  106556  063737  050560  050510    ADD     XFER,S.XFER             ; UPDATE BYTES TRANSFERED SUM. COUNTER
7507  106564  005237  050476            INC     S.REC                   ; UPDATE PACKETS RECEIVED SUM. COUNTER
7508  106570  013703  003764            MOV     RRGNXT,R3               ; GET RECEIVE RING POINTER
7509  106574                            CALL    GETRNX #RRGNXT          ; UPDATE POINTER
7510  106606  016301  000006            MOV     6(R3),R1                ; GET PACKET LENGTH FROM DISCRIPTOR
7511  106612  042701  170000            BIC     #170000,R1              ;  ZERO OUT EXCESS INFOR
7512  106616  162701  000004            SUB     #4,R1                   ;  SUBTRACT CRC B  S
7513  106622  020137  050566            CMP     R1,BUFLEN               ; CHECK FOR LENGTH ERROR
7514  106626  001423                    BEQ     37$                     ;  IF OK, BR
7515  106630  005237  050502            INC     S.LEN                   ;  ELSE, UPDATE LENGTH ERRORS COUNTER
7516  106634  012737  062342  002312    MOV     #LENGTH,KEYWD2          ;  MOVE 'LENGTH' TO ERROR INDICATOR
7517  106642  012702  177777            MOV     #-1,R2                  ;  INDICATE ERROR TO R2
7518  106646                            PRINTX  #LGERMS,BUFLEN,R1       ;  PRINT LENGTH ERROR MESSAGE
7519  106646  010146                                                            MOV     R1,-(SP)
7520  106650  013746  050566                                                    MOV     BUFLEN,-(SP)
7521  106654  012746  067117                                                    MOV     #LGERMS,-(SP)
7522  106660  012746  000003                                                    MOV     #3,-(SP)
7523  106664  010600                                                            MOV     SP,R0
7524  106666  104415                                                            TRAP    C$PNTX
7525  106670  062706  000010                                                    ADD     #10,SP
7526  106674  000450                    BR      50$                     ;  AND EXIT
7527  106676  016303  000002    37$:    MOV     2(R3),R3                ; POINT R3 TO MESSAGE BUFFER
7528  106702  066303  000016            ADD     16(R3),R3               ; POINT R3 TO DATA AFTER SKIP COUNT
7529  106706  062703  000030            ADD     #30,R3                  ; POINT R3 TO FIRST DATA BYTE
7530  106712  063737  002372  050506    ADD     P$SIZE,S.BYTE           ; UPDATE BYTES COMPARED SUMMARY COUNTER
7531  106720                            CALL    DATCMP P$SIZE,CMPBUF,R3 ; CHECK FOR DATA COMPARE ERRORS
7532  106740                            P$POP   R3                      ;  CHECK RESULTS
7533  106742  001425                    BEQ     50$                     ;   IF ERRORS,
7534  106744  060337  050504            ADD     R3,S.COMP               ;   UPDATE COMPARE ERRORS SUMMARY COUNTER
7535  106750  012737  062351  002312    MOV     #COMPAR,KEYWD2          ;   MOVE 'COMPARE' TO ERROR INDICATOR
7536  106756  012702  177777            MOV     #-1,R2                  ;   INDICATE ERROR TO R2
7537  106762  000415                    BR      50$
7538
7539  106764  012737  062316  002312    MOV     #RETRY,KEYWD2           ; MOVE 'EXCESSIVE COLLISIONS' TO ERROR INCICATOR
7540  106772  012702  177777            MOV     #-1,R2                  ;  INDICATE ERROR IN R2
7541  106776  000407                    BR      50$
7542
7543  107000  005237  050500    40$:    INC     S.NREC                  ; UPDATE MESSAGES NOT RECEIVED COUNTER
7544  107004  012737  062302  002312    MOV     #NORESP,KEYWD2          ; MOVE 'NO RESPONCE' TO ERROR INDICATOR
7545  107012  012702  177777            MOV     #-1,R2                  ; INDICATE ERROR TO R2
7546
7547  107016                    50$:    RETURN  R2                      ; RETURN
7548
7549
7550                            ;
7551                            ;ACTION ROUTINE TO SET 'RUN PATTERN' FLAG
```

```
7552                                        ;
7553
7554  107022  012737  000005  002310  ACTPAT: MOV     #CPATRN,KEYWD1          ;SET FLAG
7555  107030  000207                          RTS     PC
7556
7557
7558  107032                          RUNPAT: P$PUSH  P$TYPE                  ; SAVE TYPE PARAMETER
7559  107036  005037  002370                  CLR     P$TYPE                  ; SET TYPE TO FIRST TYPE
7560  107042                          5$:     CALL    DIRCOM                  ; SEND MESSAGES
7561  107050                                  P$POP   R1                      ; GET RESULTS TO KEEP STACK IN ORDER
7562  107052  001403                          BEQ     10$                     ; IF OK, CONT
7563  107054  022701  000001                  CMP     #1,R1                   ;  ELSE, WAS TABLE EMPTY
7564  107060  001406                          BEQ     15$                     ;  IF YES, RETURN
7565  107062  005237  002370          10$:    INC     P$TYPE                  ; SET TO NEXT TYPE
7566  107066  022737  000005  002370          CMP     #5,P$TYPE               ; SEE IF DONE ALL OF THEM
7567  107074  002362                          BGE     5$                      ;  IF NOT, DO MORE
7568  107076                          15$:    P$POP   P$TYPE                  ; RESTORE MESSAGE TYPE
7569  107102                                  RETURN
7570
7571                                        ;
7572                                        ;ACTION ROUTINE TO SHOW THE CURRENT MESSAGE PARAMETERS
7573                                        ;
7574
7575  107104  013701  002370  ACTSMS: MOV     P$TYPE,R1               ;GET MESSAGE TYPE INTO R1
7576  107110  006301                          ASL     R1                      ;MULTIPLY BY 2
7577  107112  062701  003262                  ADD     #MSGTAB,R1              ;ADD MESSAGE TABLE OFFSET
7578  107116                                  PRINTF  #MSGPRM                 ;PRINT MESSAGE PARAMETER MESSAGE
7579  107116  012746  054641                                                                  MOV     #MSGPRM,-(SP)
7580  107122  012746  000001                                                                  MOV     #1,-(SP)
7581  107126  010600                                                                          MOV     SP,R0
7582  107130  104417                                                                          TRAP    C$PNTF
7583  107132  062706  000004                                                                  ADD     #4,SP
7584  107136                                  PRINTF  #MSG4,(R1),P$SIZE,P$CPYS        ;PRINT PARAMETERS
7585  107136  013746  002374                                                                  MOV     P$CPYS,-(SP)
7586  107142  013746  002372                                                                  MOV     P$SIZE,-(SP)
7587  107146  011146                                                                          MOV     (R1),-(SP)
7588  107150  012746  055276                                                                  MOV     #MSG4,-(SP)
7589  107154  012746  000004                                                                  MOV     #4,-(SP)
7590  107160  010600                                                                          MOV     SP,R0
7591  107162  104417                                                                          TRAP    C$PNTF
7592  107164  062706  000012                                                                  ADD     #12,SP
7593  107170  105037  003160                  CLRB    P$NNUF
7594  107174  000207                          RTS     PC
7595
7596
7597                                        ;
7598                                        ;ACTION ROUTINE TO CLEAR THE CURRENT MESSAGE PARAMETERS AND
7599                                        ;RESET THEM TO THE DEFAULT VALUE
7600                                        ;
7601
7602  107176  012737  000000  002370  ACTCMS: MOV     #ALPHA,P$TYPE           ;RESET TYPE
7603  107204  012737  001000  002372          MOV     #512.,P$SIZE            ;RESET SIZE
7604  107212  012737  000001  002374          MOV     #1,P$CPYS               ;RESET COPIES
7605  107220                                  PRINTF  #CLRMSG                 ;PRINT MESSAGE PARAMETERS RESET MESSAGE
7606  107220  012746  053716                                                                  MOV     #CLRMSG,-(SP)
7607  107224  012746  000001                                                                  MOV     #1,-(SP)
```

```
 7608  107230  010600                                                          MOV     SP,R0
 7609  107232  104417                                                          TRAP    C$PNTF
 7610  107234  062706  000004                                                  ADD     #4,SP
 7611  107240                        PRINTF  #MSG4,MSGTAB,P$SIZE,P$CPYS      ;PRINT PARAMETERS
 7612  107240  013746  002374                                                  MOV     P$CPYS,-(SP)
 7613  107244  013746  002372                                                  MOV     P$SIZE,-(SP)
 7614  107250  013746  003262                                                  MOV     MSGTAB,-(SP)
 7615  107254  012746  055276                                                  MOV     #MSG4,-(SP)
 7616  107260  012746  000004                                                  MOV     #4,-(SP)
 7617  107264  010600                                                          MOV     SP,R0
 7618  107266  104417                                                          TRAP    C$PNTF
 7619  107270  062706  000012                                                  ADD     #12,SP
 7620  107274  105037  003160              CLRB    P$NNUF                    ;CLEAR NOTNUF FLAG
 7621  107300  000207                      RTS     PC
 7622
 7623
 7624                                  ;
 7625                                  ;ACTION ROUTINE TO SET SHOW COUNTERS FLAG
 7626                                  ;
 7627
 7628  107302                        ACTCNT: CALL    FUNCT #RDCNTS             ;READ COUNTERS
 7629  107314                                P$POP   R1                        ;CHECK RESULT
 7630  107316  001402                        BEQ     21$                       ;BRANCH IF ERROR
 7631  107320  000137  110346               JMP     40$
 7632                                                                          ;PRINT COUNTER INFO
 7633
 7634  107324                        21$:    CALL    BINHEX  #PHYADR,#6,#STRBUF      ;GET ADDRESS INTO ASCII
 7635  107346                                PRINTF  #CNTR00,#STRBUF
 7636  107346  012746  002322                                                  MOV     #STRBUF,-(SP)
 7637  107352  012746  062703                                                  MOV     #CNTR00,-(SP)
 7638  107356  012746  000002                                                  MOV     #2,-(SP)
 7639  107362  010600                                                          MOV     SP,R0
 7640  107364  104417                                                          TRAP    C$PNTF
 7641  107366  062706  000006                                                  ADD     #6,SP
 7642  107372                                PRINTF  #CNTR01,UCB12+2
 7643  107372  013746  050162                                                  MOV     UCB12+2,-(SP)
 7644  107376  012746  062763                                                  MOV     #CNTR01,-(SP)
 7645  107402  012746  000002                                                  MOV     #2,-(SP)
 7646  107406  010600                                                          MOV     SP,R0
 7647  107410  104417                                                          TRAP    C$PNTF
 7648  107412  062706  000006                                                  ADD     #6,SP
 7649  107416                                CALL    BINDEC  #UCB12+4
 7650  107430                                PRINTF  #CNTR02,#DECSTR
 7651  107430  012746  075200                                                  MOV     #DECSTR,-(SP)
 7652  107434  012746  063032                                                  MOV     #CNTR02,-(SP)
 7653  107440  012746  000002                                                  MOV     #2,-(SP)
 7654  107444  010600                                                          MOV     SP,R0
 7655  107446  104417                                                          TRAP    C$PNTF
 7656  107450  062706  000006                                                  ADD     #6,SP
 7657  107454                                CALL    BINDEC  #UCB12+10
 7658  107466                                PRINTF  #CNTR03,#DECSTR
 7659  107466  012746  075200                                                  MOV     #DECSTR,-(SP)
 7660  107472  012746  063066                                                  MOV     #CNTR03,-(SP)
 7661  107476  012746  000002                                                  MOV     #2,-(SP)
 7662  107502  010600                                                          MOV     SP,R0
 7663  107504  104417                                                          TRAP    C$PNTF
```

```
7664  107506  062706  000006                                         ADD     #6,SP
7665  107512                      PRINTF  #CNTR04,UCB12+14
7666  107512  013746  050174                                         MOV     UCB12+14,-(SP)
7667  107516  012746  063133                                         MOV     #CNTR04,-(SP)
7668  107522  012746  000002                                         MOV     #2,-(SP)
7669  107526  010600                                                 MOV     SP,R0
7670  107530  104417                                                 TRAP    C$PNTF
7671  107532  062706  000006                                         ADD     #6,SP
7672  107536                      PRINTF  #CNTR05,UCB12+16
7673  107536  013746  050176                                         MOV     UCB12+16,-(SP)
7674  107542  012746  063210                                         MOV     #CNTR05,-(SP)
7675  107546  012746  000002                                         MOV     #2,-(SP)
7676  107552  010600                                                 MOV     SP,R0
7677  107554  104417                                                 TRAP    C$PNTF
7678  107556  062706  000006                                         ADD     #6,SP
7679  107562                      CALL    BINDEC  #UCB12+20
7680  107574                      PRINTF  #CNTR06,#DECSTR
7681  107574  012746  075200                                         MOV     #DECSTR,-(SP)
7682  107600  012746  063260                                         MOV     #CNTR06,-(SP)
7683  107604  012746  000002                                         MOV     #2,-(SP)
7684  107610  010600                                                 MOV     SP,R0
7685  107612  104417                                                 TRAP    C$PNTF
7686  107614  062706  000006                                         ADD     #6,SP
7687  107620                      CALL    BINDEC  #UCB12+24
7688  107632                      PRINTF  #CNTR07,#DECSTR
7689  107632  012746  075200                                         MOV     #DECSTR,-(SP)
7690  107636  012746  063317                                         MOV     #CNTR07,-(SP)
7691  107642  012746  000002                                         MOV     #2,-(SP)
7692  107646  010600                                                 MOV     SP,R0
7693  107650  104417                                                 TRAP    C$PNTF
7694  107652  062706  000006                                         ADD     #6,SP
7695  107656                      PRINTF  #CNTR08,UCB12+30
7696  107656  013746  050210                                         MOV     UCB12+30,-(SP)
7697  107662  012746  063367                                         MOV     #CNTR08,-(SP)
7698  107666  012746  000002                                         MOV     #2,-(SP)
7699  107672  010600                                                 MOV     SP,R0
7700  107674  104417                                                 TRAP    C$PNTF
7701  107676  062706  000006                                         ADD     #6,SP
7702  107702                      PRINTF  #CNTR09,UCB12+32
7703  107702  013746  050212                                         MOV     UCB12+32,-(SP)
7704  107706  012746  063442                                         MOV     #CNTR09,-(SP)
7705  107712  012746  000002                                         MOV     #2,-(SP)
7706  107716  010600                                                 MOV     SP,R0
7707  107720  104417                                                 TRAP    C$PNTF
7708  107722  062706  000006                                         ADD     #6,SP
7709  107726                      CALL    BINDEC  #UCB12+34
7710  107740                      PRINTF  #CNTR10,#DECSTR
7711  107740  012746  075200                                         MOV     #DECSTR,-(SP)
7712  107744  012746  063513                                         MOV     #CNTR10,-(SP)
7713  107750  012746  000002                                         MOV     #2,-(SP)
7714  107754  010600                                                 MOV     SP,R0
7715  107756  104417                                                 TRAP    C$PNTF
7716  107760  062706  000006                                         ADD     #6,SP
7717  107764                      CALL    BINDEC  #UCB12+40
7718  107776                      PRINTF  #CNTR11,#DECSTR
7719  107776  012746  075200                                         MOV     #DECSTR,-(SP)
```

```
 7720  110002  012746  063552                                    MOV     #CNTR11,-(SP)
 7721  110006  012746  000002                                    MOV     #2,-(SP)
 7722  110012  010600                                            MOV     SP,R0
 7723  110014  104417                                            TRAP    C$PNTF
 7724  110016  062706  000006                                    ADD     #6,SP
 7725  110022                    CALL    BINDEC  #UCB12+44
 7726  110034                    PRINTF  #CNTR12,#DECSTR
 7727  110034  012746  075200                                    MOV     #DECSTR,-(SP)
 7728  110040  012746  063622                                    MOV     #CNTR12,-(SP)
 7729  110044  012746  000002                                    MOV     #2,-(SP)
 7730  110050  010600                                            MOV     SP,R0
 7731  110052  104417                                            TRAP    C$PNTF
 7732  110054  062706  000006                                    ADD     #6,SP
 7733  110060                    CALL    BINDEC  #UCB12+50
 7734  110072                    PRINTF  #CNTR13,#DECSTR
 7735  110072  012746  075200                                    MOV     #DECSTR,-(SP)
 7736  110076  012746  063670                                    MOV     #CNTR13,-(SP)
 7737  110102  012746  000002                                    MOV     #2,-(SP)
 7738  110106  010600                                            MOV     SP,R0
 7739  110110  104417                                            TRAP    C$PNTF
 7740  110112  062706  000006                                    ADD     #6,SP
 7741  110116                    CALL    BINDEC  #UCB12+54
 7742  110130                    PRINTF  #CNTR14,#DECSTR
 7743  110130  012746  075200                                    MOV     #DECSTR,-(SP)
 7744  110134  012746  063735                                    MOV     #CNTR14,-(SP)
 7745  110140  012746  000002                                    MOV     #2,-(SP)
 7746  110144  010600                                            MOV     SP,R0
 7747  110146  104417                                            TRAP    C$PNTF
 7748  110150  062706  000006                                    ADD     #6,SP
 7749  110154                    CALL    BINDEC  #UCB12+60
 7750  110166                    PRINTF  #CNTR15,#DECSTR
 7751  110166  012746  075200                                    MOV     #DECSTR,-(SP)
 7752  110172  012746  063771                                    MOV     #CNTR15,-(SP)
 7753  110176  012746  000002                                    MOV     #2,-(SP)
 7754  110202  010600                                            MOV     SP,R0
 7755  110204  104417                                            TRAP    C$PNTF
 7756  110206  062706  000006                                    ADD     #6,SP
 7757  110212                    CALL    BINDEC  #UCB12+64
 7758  110224                    PRINTF  #CNTR16,#DECSTR
 7759  110224  012746  075200                                    MOV     #DECSTR,-(SP)
 7760  110230  012746  064033                                    MOV     #CNTR16,-(SP)
 7761  110234  012746  000002                                    MOV     #2,-(SP)
 7762  110240  010600                                            MOV     SP,R0
 7763  110242  104417                                            TRAP    C$PNTF
 7764  110244  062706  000006                                    ADD     #6,SP
 7765  110250                    PRINTF  #CNTR17,UCB12+70
 7766  110250  013746  050250                                    MOV     UCB12+70,-(SP)
 7767  110254  012746  064101                                    MOV     #CNTR17,-(SP)
 7768  110260  012746  000002                                    MOV     #2,-(SP)
 7769  110264  010600                                            MOV     SP,R0
 7770  110266  104417                                            TRAP    C$PNTF
 7771  110270  062706  000006                                    ADD     #6,SP
 7772  110274                    PRINTF  #CNTR18,UCB12+72
 7773  110274  013746  050252                                    MOV     UCB12+72,-(SP)
 7774  110300  012746  064154                                    MOV     #CNTR18,-(SP)
 7775  110304  012746  000002                                    MOV     #2,-(SP)
```

```
 7776  110310  010600                                                          MOV     SP,R0
 7777  110312  104417                                                          TRAP    C$PNTF
 7778  110314  062706  000006                                                  ADD     #6,SP
 7779  110320                        PRINTF  #CNTR19,UCB12+74
 7780  110320  013746  050254                                                  MOV     UCB12+74,-(SP)
 7781  110324  012746  064221                                                  MOV     #CNTR19,-(SP)
 7782  110330  012746  000002                                                  MOV     #2,-(SP)
 7783  110334  010600                                                          MOV     SP,R0
 7784  110336  104417                                                          TRAP    C$PNTF
 7785  110340  062706  000006                                                  ADD     #6,SP
 7786  110344  000404                BR      50$
 7787
 7788  110346                  40$:  ERRHRD  31,EMSG31
 7789  110346  104456                                                          TRAP    C$ERHRD
 7790  110350  000037                                                          .WORD   31
 7791  110352  065576                                                          .WORD   EMSG31
 7792  110354  000000                                                          .WORD   0
 7793
 7794  110356  105037  003160    50$:  CLRB    P$NNUF
 7795  110362  000207                RTS     PC
 7796
 7797
 7798                            ;
 7799                            ;ACTION ROUTINE TO PRINT OUT THE NODE TABLE
 7800                            ;
 7801
 7802  110364  105037  003160    ACTSND: CLRB    P$NNUF
 7803  110370  012737  002404  002402    MOV     #NODTBL,SLOT            ;MOVE NODE TABLE ADDRESS INTO SLOT
 7804  110376                    CALL    FULSLT                          ;SEE IF TABLE EMPTY
 7805  110404  022737  177777  002402    CMP     #-1,SLOT                ;IF YES, DON'T PRINT HEADER
 7806  110412  001437            BEQ     15$
 7807  110414                    PRINTF  #NTBHDR                         ;PRINT NODE TABLE HEADER
 7808  110414  012746  053462                                                  MOV     #NTBHDR,-(SP)
 7809  110420  012746  000001                                                  MOV     #1,-(SP)
 7810  110424  010600                                                          MOV     SP,R0
 7811  110426  104417                                                          TRAP    C$PNTF
 7812  110430  062706  000004                                                  ADD     #4,SP
 7813  110434                  10$:  CALL    FULSLT                          ;FIND LOCATION IN TABLE WITH AN ADDRESS
 7814  110442  022737  177777  002402    CMP     #-1,SLOT                ;CHECK IF AT END OF TABLE
 7815  110450  001432            BEQ     20$                             ;IF YES, RETURN
 7816  110452                    CALL    BINHEX SLOT,#6,#STRBUF          ;ELSE, PUT ASCII ADDRESS INTO BUFFER
 7817  110474                    CALL    PRTNOD                          ;PRINT NODE TABLE ENTRY
 7818  110502  062737  000010  002402    ADD     #8.,SLOT                ;INCR. SLOT TO POINT TO NEXT TABLE ENTRY
 7819  110510  000751            BR      10$                             ;CONTINUE UNTIL ALL ENTRIES PRINTED
 7820  110512                    PRINTF  #TABEMT,#NOD
 7821  110512  012746  053701                                                  MOV     #NOD,-(SP)
 7822  110516  012746  053632                                                  MOV     #TABEMT,-(SP)
 7823  110522  012746  000002                                                  MOV     #2,-(SP)
 7824  110526  010600                                                          MOV     SP,R0
 7825  110530  104417                                                          TRAP    C$PNTF
 7826  110532  062706  000006                                                  ADD     #6,SP
 7827  110536  000207            20$:  RTS     PC                              ;RETURN
 7828
 7829
 7830                            ;
 7831                            ;ACTION ROUTINE TO CLEAR A NODE SPECIFIED BY NODE LOGICAL NAME
```

```
                                        ;FROM THE NODE TABLE
                                        ;
                                        ;
110540                                  ACTCNL: P$PUSH  R2                      ;SAVE R2
110542  013702  003154                          MOV     P$NUM,R2                ;PUT NODE LOGICAL NUMBER INTO R2
110546  006302                                  ASL     R2                      ;MULTIPLY BY 8
110550  006302                                  ASL     R2                      ;NODE TABLE ADDRESS =
110552  006302                                  ASL     R2                      ;  (LOG. NO. X 8) + #NODTBL
110554  062702  002404                          ADD     #NODTBL,R2              ;ADD OFFSET
110560  005022                                  CLR     (R2)+                   ;CLEAR ENTRY (8 BYTES)
110562  005022                                  CLR     (R2)+
110564  005022                                  CLR     (R2)+
110566  005012                                  CLR     (R2)
110570                                          P$POP   R2                      ;RESTORE R2
110572  105037  003160                          CLRB    P$NNUF                  ;CLEAR NOTNUF FLAG
110576                                          PRINTF  #LOGDEL,P$NUM           ;PRINT MESSAGE INDICATING DELETION
110576  013746  003154                                                                  MOV     P$NUM,-(SP)
110602  012746  054426                                                                  MOV     #LOGDEL,-(SP)
110606  012746  000002                                                                  MOV     #2,-(SP)
110612  010600                                                                          MOV     SP,R0
110614  104417                                                                          TRAP    C$PNTF
110616  062706  000006                                                                  ADD     #6,SP
110622  000207                                  RTS     PC                      ;RETURN
                                        ;
                                        ;ACTION ROUTINE TO INITIATE A UNA PORT COMMAND
                                        ;
110624  105037  003160                  ACTFCT: CLRB    P$NNUF                  ;CLEAR NOTNUF FLAG
110630                                          CALL    FUNCT P$NUM             ;CALL FUNCTION ROUTINE WITH FUNCTION CODE
110642                                          P$POP   R1                      ;CHECK RESULTS
110644  001404                                  BEQ     1$                      ; IF OK EXIT
110646                                          ERRHRD  30,EMSG30               ; ELSE REPORT ERROR
110646  104456                                                                  TRAP    C$ERHRD
110650  000036                                                                  .WORD   30
110652  065531                                                                  .WORD   EMSG30
110654  000000                                                                  .WORD   0
110656  000207                          1$:     RTS     PC
                                        ;
                                        ;ACTION ROUTINE TO SAVE NODE TABLE
                                        ;
110660                                  ACTSAV: P$PUSH  R2,R3                   ;SAVE R2 AND R3
110664  012702  002404                          MOV     #NODTBL,R2              ;SET REGISTERS FOR COPYING
110670  012703  002530                          MOV     #SAVTBL,R3              ;R2 = FROM, R3 = TO
110674                                          PRINTF  #UNSMSG,#SAVED          ;PRINT 'TABLE SAVED' MESSAGE
110674  012746  054620                                                                  MOV     #SAVED,-(SP)
110700  012746  054561                                                                  MOV     #UNSMSG,-(SP)
110704  012746  000002                                                                  MOV     #2,-(SP)
110710  010600                                                                          MOV     SP,R0
110712  104417                                                                          TRAP    C$PNTF
110714  062706  000006                                                                  ADD     #6,SP
110720  000137  111002                          JMP     SAVCOM
                                        ;
```

```
                                        ;ACTION ROUTINE TO UNSAVE NODE TABLE
                                        ;
                                        ;
110724  105037  003160    ACTUNS: CLRB    P$NNUF                  ;CLEAR 'NOT ENOUGH' FLAG
110730                            P$PUSH  R2,R3                   ;SAVE R2 AND R3
110734  121427  000057            CMPB    (R4),#57
110740  001002                    BNE     5$
110742  000137  111014            JMP     QUIT
110746  012703  002404    5$:     MOV     #NODTBL,R3              ;SET REGISTERS FOR COPYING
110752  012702  002530            MOV     #SAVTBL,R2              ;R2 = FROM, R3 = TO
110756                            PRINTF  #UNSMSG,#RESTOR         ;PRINT 'TABLE RESTORED' MESSAGE
110756  012746  054627                                                    MOV     #RESTOR,-(SP)
110762  012746  054561                                                    MOV     #UNSMSG,-(SP)
110766  012746  000002                                                    MOV     #2,-(SP)
110772  010600                                                            MOV     SP,R0
110774  104417                                                            TRAP    C$PNTF
110776  062706  000006                                                    ADD     #6,SP
111002  012701  000050    SAVCOM: MOV     #TBLLEN,R1              ;MOVE TABLE LENGTH TO R1
111006  012223            10$:    MOV     (R2)+,(R3)+             ;MOVE WORD
111010  005301                    DEC     R1                      ;DECREMENT COUNTER
111012  001375                    BNE     10$                     ;IF MORE, LOOP
111014                    QUIT:   P$POP   R2,R3                   ; ELSE, RESTORE COUNTERS
111020  105037  003160            CLRB    P$NNUF                  ; CLEAR 'NOT ENOUGH' FLAG
111024  000207                    RTS     PC
                                        ;
                                        ;ACTION ROUTINE TO CLEAR SUMMARY TABLE
                                        ;
111026  105037  003160    ACTCSU: CLRB    P$NNUF                  ;CLEAR 'NOT ENOUGH' COUNTER
111032                            P$PUSH  R2                      ;SAVE R2
111034  012701  000132            MOV     #STBLEN,R1              ;MOVE TABLE LENGTH TO R1
111040  012702  002656            MOV     #STATBL,R2              ;MOVE SUMMARY TABLE ADDRESS TO R2
111044  005022            5$:     CLR     (R2)+                   ;CLEAR FIRST WORD
111046  005301                    DEC     R1                      ;SEE IF FINISHED
111050  001375                    BNE     5$                      ; IF NO, DO MORE
111052                            PRINTF  #TABCLR,#SUMM           ; ELSE, PRINT 'TABLE CLEARED' MESSAGE
111052  012746  053706                                                    MOV     #SUMM,-(SP)
111056  012746  054514                                                    MOV     #TABCLR,-(SP)
111062  012746  000002                                                    MOV     #2,-(SP)
111066  010600                                                            MOV     SP,R0
111070  104417                                                            TRAP    C$PNTF
111072  062706  000006                                                    ADD     #6,SP
111076                            P$POP   R2                      ; AND RESTORE R2
111100  000207                    RTS     PC
                                        ;
                                        ;ACTION ROUTINE TO CHECK FOR PASS DEFAULT VALUE
                                        ;
111102                    ACTDFT:
111102  121427  000040    1$:     CMPB    (R4),#40                ;SEE IF SPACES
111106  001002                    BNE     2$                      ; IF NO, CONT.
111110  005204                    INC     R4                      ; ELSE, POINT TO NEXT CHAR
111112  000773                    BR      1$                      ; AND CHECK AGAIN
111114  121427  000000    2$:     CMPB    (R4),#0                 ;SEE IF DEFAULT VALUE
```

```
7944  111120  001007                          BNE     10$                     ; IF NO, BR
7945  111122  012763  000030  000002          MOV     #30,2(R3)               ; IF YES, POINT R3 TO SKIP CHECK PASS COUNT
7946  111130  012737  000001  003154          MOV     #1,P$NUM                ;SET DEFAULT TO 1
7947  111136  000403                          BR      15$                     ;RETURN
7948  111140  012763  000004  000002  10$:    MOV     #4,2(R3)                ;POINT R3 TO CHECK FOR PASS COUNT
7949  111146  000207                  15$:    RTS     PC
7950
7951                                          ;
7952                                          ;ACTION ROUTINE TO READ A FILE FROM EXTERNAL MEDIA ONTO THE NODE TABLE
7953                                          ;
7954
7955  111150                          ACTUSF:
7956  111150                                  P$PUSH  R2                      ;SAVE R2
7957  111152  005002                          CLR     R2                      ;INITIALIZE R2 TO NODE TYPE 'TARGET'
7958  111154                                  OPEN    CBOADR                  ;OPEN FILE, NAME=ASCIZ STRING
7959  111154  013700  002366                                                          MOV     CBOADR,R0
7960  111160  104434                                                                  TRAP    C$OPEN
7961  111162                                  BCOMPLETE       1$              ;RETURN IF SUCCESSFUL
7962  111162  103413                                                                  BCS     1$
7963  111164                                  PRINTF  #OPNERR,CBOADR          ; ELSE PRINT "OPEN ERROR"
7964  111164  013746  002366                                                          MOV     CBOADR,-(SP)
7965  111170  012746  112110                                                          MOV     #OPNERR,-(SP)
7966  111174  012746  000002                                                          MOV     #2,-(SP)
7967  111200  010600                                                                  MOV     SP,R0
7968  111202  104417                                                                  TRAP    C$PNTF
7969  111204  062706  000006                                                          ADD     #6,SP
7970  111210                                  CLOSE                           ; CLOSE FILE
7971  111210  104435                                                                  TRAP    C$CLOS
7972  111212                          1$:     CALL    RDLIN                   ;READ A LINE AT A TIME
7973  111220  005737  112104                  TST     BAD                     ;SEE IF AN ERROR DURING READ
7974  111224  001064                          BNE     25$                     ; BR ON ERROR TO LEAVE
7975  111226  005737  112106                  TST     EOFF                    ;SEE IF EOF BEFORE PROCESS
7976  111232  001411                          BEQ     10$                     ;IF VALID, PROCESS
7977  111234                                  PRINTF  #EOFFND                 ; ELSE SAY 'END OF FILE' AND LEAVE
7978  111234  012746  112223                                                          MOV     #EOFFND,-(SP)
7979  111240  012746  000001                                                          MOV     #1,-(SP)
7980  111244  010600                                                                  MOV     SP,R0
7981  111246  104417                                                                  TRAP    C$PNTF
7982  111250  062706  000004                                                          ADD     #4,SP
7983  111254  000450                          BR      25$
7984  111256                          10$:    PRINTF  #PLINE,#FILLIN          ;PRINT LINE READ FROM FILE
7985  111256  012746  111560                                                          MOV     #FILLIN,-(SP)
7986  111262  012746  112172                                                          MOV     #PLINE,-(SP)
7987  111266  012746  000002                                                          MOV     #2,-(SP)
7988  111272  010600                                                                  MOV     SP,R0
7989  111274  104417                                                                  TRAP    C$PNTF
7990  111276  062706  000006                                                          ADD     #6,SP
7991  111302                                  CALL    EDPACK #FILLIN,#ADRBUF,#6       ;PUT ADDRESS INTO BINARY
7992  111324                                  P$POP   R1                      ;CHECK RESULTS
7993  111326  001411                          BEQ     12$                     ; IF OK, BR
7994  111330                                  PRINTF  #CADERR                 ; ELSE PRINT ERROR MESSAGE
7995  111330  012746  053306                                                          MOV     #CADERR,-(SP)
7996  111334  012746  000001                                                          MOV     #1,-(SP)
7997  111340  010600                                                                  MOV     SP,R0
7998  111342  104417                                                                  TRAP    C$PNTF
7999  111344  062706  000004                                                          ADD     #4,SP
```

```
                        BR      25$                     ; AND EXIT
                12$:    MOVB    R2,NODTY                ;SET UP NODE TYPE
                        COMB    R2                      ;SWITCH TYPE FOR NEXT TIME
                        BICB    #376,R2
                        CALL    ENTRND                  ;ENTER IN NODE TABLE
                        P$POP   R1                      ;GET RESULTS
                        BR      1$                      ;READ MORE ADDRESS
                25$:    P$POP   R2                      ;RESTORE R2
                        CLRB    P$NNUF                  ;CLEAR 'NOT ENOUGH' FLAG
                        PRINTF  #UNSMSG,#RESTOR         ;PRINT 'TALBE RESTORED' MESSAGE
                                                                        MOV     #RESTOR,-(SP)
                                                                        MOV     #UNSMSG,-(SP)
                                                                        MOV     #2,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTF
                                                                        ADD     #6,SP
                        RETURN



                .SBTTL  READ LINE OF OPENED FILE
                ;
                ; THIS ROUTINE GETS BYTES FROM AN OPENED FILE UNTIL A CR IS ENCOUNTERED
                ; "EOF" AND "BAD" FLAGS ARE SET IF END-OF-FILE OR ERRORS ARE ENCOUNTERED
                ;
                ; NOTE: ASSUMING A ASCII TEXT FILE IS BEING READ, FOR EXAMPLE:
                ;       AA-00-03-00-01-AB<CR><LF>
                ;                        ''
                ;       AA-00-03-00-01-AB<CR><LF>
                ;
                ;       WHAT YOU SEE READ BYTE-BY-BYTE IS:
                ;            "A..-AB<CR><LF>A..-AB<CR><LF>..<0><0><0>.....???
                ;       SO I MADE ASSUMPTION THAT SINCE SEE "0-PADDING" AFTER LAST CHAR TO
                ;       END-OF-FILEBLOCK, ANY CHARACTER THAT IS NOT "SPACE OR GREATER" OR A
                ;       <CR> OR <LF> THEN I'LL TAKE THAT AS END-OF-FILE(TEXT), SET EOF-FLAG
                ;       AND LEAVE.
                ;
                ;       INPUTS:
                ;               FILLIN  BUFFER TO HOLD LINE OF BYTES READ FROM OPENED FILE
                ;                       (CR NOT INCLUDED, 0-BYTE TERMINATED)
                ;       OUTPUTS:
                ;               BAD     IF NON-ZERO, ERROR IN READING A BYTE FROM FILE
                ;               EOFF    IF NON-ZERO, END OF FILE WAS ENCOUNTERED
                ;               FILLIN  ASCIZ STRING THAT WAS READ AS CHAR-CR-LF STRING
                ;                        (CR-LF REMOVED)
                ;
                RDLIN:  MOV     #FILLIN,R2              ;POINT R2 TO A LINE BUFFER
                        CLR     BAD                     ;CLEAR FLAGS
                        CLR     EOFF
                1$:     GETBYT  (R2)                    ;GO GET A BYTE FROM INPUT FILE
                                                                        TRAP    C$GETB
                                                                        MOVB    R0,(R2)
                        BCOMPLETE       2$              ;BR IF READ-BYTE SUCESSFUL
                                                                        BCS     2$
                4$:     PRINTF  #RDERR                  ; ELSE PRINT "READ-ERROR"
                                                                        MOV     #RDERR,-(SP)
```

```
 8056  111460  012746  000001                                                          MOV     #1,-(SP)
 8057  111464  010600                                                                  MOV     SP,R0
 8058  111466  104417                                                                  TRAP    C$PNTF
 8059  111470  062706  000004                                                          ADD     #4,SP
 8060  111474  012737  177777  112104          MOV     #-1,BAD                 ; SET BAD-TRY FLAG AND LEAVE
 8061  111502  000416                          BR      5$
 8062
 8063  111504  122712  000015          2$:     CMPB    #15,(R2)                ;IS THE CHARACTER A <CR>
 8064  111510  001756                          BEQ     1$                      ; BR IF YES (GO BACK TO GET <LF>)
 8065  111512  122712  000012                  CMPB    #12,(R2)                ;IS THE CHARACTER A <LF>
 8066  111516  001410                          BEQ     5$                      ; BR IF YES (TERMINATE AND LEAVE)
 8067  111520  122712  000040                  CMPB    #40,(R2)                ;IS IT A "EOF" (END-OF-FILE(TEXT))
 8068                                                                          ; (EOF=ANY NON-CHAR>37 EXCEPT CR,LF)
 8069  111524  101002                          BHI     6$                      ; BR IF YES
 8070  111526  005202                          INC     R2                      ; IF NO, LEAVE CHAR IN BUFFER
 8071  111530  000746                          BR      1$                      ; AND GO GET MORE CHARS
 8072
 8073  111532  012737  177777  112106  6$:     MOV     #-1,EOFF                ;IF YES, TERMINATE INPUT BUFF
 8074                                                                          ; AND SET EOF-FLAG
 8075  111540  105012                  5$:     CLRB    (R2)
 8076  111542                                  RETURN
 8077
 8078  111544  000014                  FILENM: .BLKB   12.             ;BUFFER FOR FILE NAME
 8079  111560  000204                  FILLIN: .BLKB   132.            ;BUFFER FOR SINGLE LINE READ FROM FILE
 8080  111764  000120                  MATCH:  .BLKB   80.             ;BUFFER FOR WORD TO MATCH FROM FILE
 8081  112104  000000                  BAD:    .WORD   0               ;ERROR/NOT-FOUND FLAG WORD
 8082  112106  000000                  EOFF:   .WORD   0               ;END-OF-FILE FLAG (<>0 = EOF)
 8083
 8084  112110  047045  040445  052477  OPNERR: .ASCIZ  /%N%A?UNABLE TO OPEN '"%T%A"?/
 8085  112116  040516  046102  020105
 8086  112124  047524  047440  042520
 8087  112132  020116  022442  022524
 8088  112140  021101  000077
 8089  112144  047045  040445  043077  RDERR:  .ASCIZ  /%N%A?FILE READ ERROR?/
 8090  112152  046111  020105  042522
 8091  112160  042101  042440  051122
 8092  112166  051117  000077
 8093  112172  047045  040445  044506  PLINE:  .ASCIZ  /%N%AFILE LINE WAS:%N%T%N/
 8094  112200  042514  046040  047111
 8095  112206  020105  040527  035123
 8096  112214  047045  052045  047045
 8097  112222     000
 8098  112223     045  022516  042501  EOFFND: .ASCIZ  /%N%AEND-OF-FILE FOUND, FILE READ/
 8099  112230  042116  047455  026506
 8100  112236  044506  042514  043040
 8101  112244  052517  042116  020054
 8102  112252  044506  042514  051040
 8103  112260  040505  000104
 8104
 8105                                    ;--+
 8106                                    ;  |     SELMSG                          OPERATOR SELECTED MESSAGE STORAGE
 8107                                    ;  |
 8108                                    ;  |     THIS ROUTINE WILL TAKE THE OPERATOR SELECTED MESSAGE FROM THE COMMAND
 8109                                    ;  |     LINE INPUT STRING BUFFER AND PUT IT INTO A BUFFER AT LOCATION OPSLBF.
 8110                                    ;  |
 8111                                    ;  |     INPUTS  -               P1 - ADDRESS OF OPERATOR SELECTED MESSAGE IN
```

```
8112                                    ;                                       INPUT STRING
8113                                    ;       EXPLICIT OUTPUTS -      NONE
8114                                    ;       IMPLICIT OUTPUTS -      THE BUFFER AT OPSLBF WILL CONTAIN THE ASCII
8115                                    ;                               OPERATOR SELECTED INPUT STRING FOLLOWED BY
8116                                    ;                               A NULL CHARACTER
8117                                    ;       SUBORDINATE ROUTINES -  NONE
8118                                    ;       CALLING PROCEEDURE -    CALL SELMSG P1  ;INPUT ADDRESS OF ASCII STRING
8119                                    ;       REGISTER USAGE -        R1 CONTAINS ADDRESS OF INPUT STRING
8120                                    ;                               R2 CONTAINS ADDRESS OF OUTPUT STRING
8121                                    ;
8122                                    ;--+
8123
8124  112264                            SELMSG: P$POP   R1              ;PUT ADDRESS OF OPR. SEL ASCII STRING INTO R1
8125  112266  012702  003570                    MOV     #OPSLBF,R2      ;PUT ADDRESS OF OUTPUT BUFFER INTO R2
8126  112272  005003                            CLR     R3              ;CLEAR CHARACTER COUNTER
8127  112274  105711                    5$:     TSTB    (R1)            ;CHECK FOR END OF STRING
8128  112276  001404                            BEQ     10$             ;GO TO 10$ IF END
8129  112300  112122                            MOVB    (R1)+,(R2)+     ;ELSE, MOVE BYTE TO OUTPUT BUFFER
8130  112302  005203                            INC     R3              ;COUNT NUMBER OF CHARACTERS IN INPUT BUFFER
8131  112304  000137  112274                    JMP     5$              ;GO DO MORE CHARACTERS
8132  112310  112712  000000            10$:    MOVB    #0,(R2)         ;PUT ZERO AT END OF OUTPUT BUFFER
8133  112314  010337  003314                    MOV     R3,MSG6C        ;STORE NUMBER OF CHARACTERS FOR USE IN BUF. BUILDING
8134  112320                                    RETURN
8135
8136                                    ;--+
8137                                    ;       ENTRND                                          ENTER NODE IN TABLE
8138                                    ;
8139                                    ;       THIS ROUTINE ENTERS A NODE INTO THE NODE TABLE
8140                                    ;
8141                                    ;       INPUTS                  NONE
8142                                    ;       EXPLICIT OUTPUTS        P1 - ZERO IF SUCCESSFUL, -1 IF TABLE FULL
8143                                    ;       IMPLICIT OUTPUTS        THE ADDRESS CONTAINED IN ADRBUF WILL BE
8144                                    ;                               ADDED TO THE NODE TABLE IN THE FIRST
8145                                    ;                               AVAILABLE SLOT WITH THE NODE TYPE CONTAINED
8146                                    ;                               IN NODTY (TARGET OR ASSIST)
8147                                    ;       SUBORDINATE ROUTINES    FINDSL - FIND EMPTY SLOT IN TABLE
8148                                    ;       CALLING PROCEEDURE      CALL ENTRND
8149                                    ;                               P$POP P1        ;OUTPUT GOOD/BAD RESULT
8150                                    ;
8151                                    ;--+
8152
8153  112322                            ENTRND: CALL FINDSL             ;FIND AVAILABLE SLOT IN TABLE
8154  112330                                    P$POP   R1              ;CHECK IF TABLE FULL
8155  112332  001403                            BEQ     5$              ;IF NOT FULL BR TO 5$
8156  112334                                    P$PUSH  #-1             ;ELSE PUT FULL INDICATION ON STACK
8157  112340  000416                            BR      20$             ;RETURN
8158  112342  012703  000006            5$:     MOV     #6,R3           ;SET INCR. COUNTER TO 6 (BYTES)
8159  112346  013701  002402                    MOV     SLOT,R1         ;MOV ADDRESS OF AVAILABLE SLOT TO R1
8160  112352  012702  002314                    MOV     #ADRBUF,R2      ;MOV ADDRESS OF NODE ADDRESS TO R2
8161  112356  112221                    10$:    MOVB    (R2)+,(R1)+     ;MOV BYTE OF ADDRESS
8162  112360  005303                            DEC     R3              ;DECR. COUNTER
8163  112362  001375                            BNE     10$             ;CONTINUE UNTIL 6 BYTES TRANSFERED
8164  112364  005201                            INC     R1              ;SET POINTER TO NODE TYPE LOCATION
8165  112366  113711  002400                    MOVB    NODTY,(R1)      ;MOVE NODE TYPE INTO TABLE
8166  112372                                    P$PUSH  #0              ;PUT ADDRESS ADDED INDICATION ON STACK
8167  112376                            20$:    RETURN                  ;RETURN
```

```
 8168
 8169                                   ;--+
 8170                                   ;       FINDSL                          FIND EMPTY SLOT IN NODE TABLE
 8171                                   ;
 8172                                   ;       INPUTS                  NONE
 8173                                   ;       EXPLICIT OUTPUTS        NONE
 8174                                   ;       IMPLICIT OUTPUTS        THE ADDRESS OF THE FIRST AVAILABLE SLOT IN THE
 8175                                   ;                               NODE TABLE WILL BE LOCATED IN SLOT.  THE
 8176                                   ;                               PARAMETER STACK WILL CONTAIN -1 IF THE NODE
 8177                                   ;                               TABLE IS FULL AND 0 IF AN EMPTY SLOT WAS FOUND
 8178                                   ;       SUBORDINATE ROUTINES    NONE
 8179                                   ;       CALLING PROCEDURE       CALL FINDSL
 8180                                   ;                               P$POP  P1       ;-1 IF FULL/ 0 IF SLOT AVAILABLE
 8181                                   ;
 8182                                   ;--+
 8183
 8184  112400  012702  002404  FINDSL: MOV     #NODTBL,R2              ;MOVE ADDRESS OF NODE TABLE TO R2
 8185  112404  022712  000000  10$:    CMP     #0,(R2)                 ;SEE IF SLOT EMPTY
 8186  112410  001422                  BEQ     20$                     ;IF YES, BR 20$
 8187  112412  062702  000010          ADD     #8.,R2                  ;ELSE MOVE POINTER TO NEXT ENTRY LOC.
 8188  112416  022712  177777          CMP     #-1,(R2)                ;SEE IF AT END OF TABLE
 8189  112422  001370                  BNE     10$                     ;IF NOT, CONTINUE LOOKING
 8190  112424                          PRINTF  #TABFUL,#NOD            ;ELSE, PRINT TABLE FULL MESSAGE
 8191  112424  012746  053701                                                  MOV     #NOD,-(SP)
 8192  112430  012746  053560                                                  MOV     #TABFUL,-(SP)
 8193  112434  012746  000002                                                  MOV     #2,-(SP)
 8194  112440  010600                                                          MOV     SP,R0
 8195  112442  104417                                                          TRAP    C$PNTF
 8196  112444  062706  000006                                                  ADD     #6,SP
 8197  112450                          P$PUSH  #-1                     ;PUT TABLE FULL INDICATION ON STACK
 8198  112454  000404                  BR      30$                     ;RETURN
 8199  112456  010237  002402  20$:    MOV     R2,SLOT                 ;MOVE ADDRESS OF EMPTY LOC. INTO SLOT
 8200  112462                          P$PUSH  #0                      ;PUT LOC. FOUND INDICATION ON STACK
 8201  112466                  30$:    RETURN                          ;RETURN
 8202
 8203                                   ;--+
 8204                                   ;       FULSLT                                  FULL SLOT ROUTINE
 8205                                   ;
 8206                                   ;       THIS ROUTINE FINDS A LOCATION IN THE TABLE WHERE A NODE PHYSICAL
 8207                                   ;       ADDRESS EXISTS.  IT IS USED WHEN PRINTING OUT THE NODE TABLE.
 8208                                   ;
 8209                                   ;       INPUTS                  NONE
 8210                                   ;       EXPLICIT OUTPUTS        NONE
 8211                                   ;       IMPLICIT OUTPUTS        THE LOCATION SLOT WILL CONTAIN THE PHYSICAL
 8212                                   ;                               ADDRESS OF A NODE TABLE ENTRY.  SLOT WILL
 8213                                   ;                               CONTAIN -1 WHEN POINTING TO THE END OF THE
 8214                                   ;                               NODE TABLE
 8215                                   ;       SUBORDINATE ROUTINES    NONE
 8216                                   ;       CALLING PROCEDURE       CALL FULSLT
 8217                                   ;
 8218                                   ;--+
 8219
 8220  112470  013701  002402  FULSLT: MOV     SLOT,R1                 ;MOVE SLOT LOCATION TO R1
 8221  112474  022711  000000  10$:    CMP     #0,(R1)                 ;CHECK IF EMPTY
 8222  112500  001412                  BEQ     20$                     ;IF YES, BR 20$
 8223  112502  022711  177777          CMP     #-1,(R1)                ;SEE IF END OF NODE TABLE
```

```
 8224  112506  001403                          BEQ     15$             ;IF YES, BR 15$
 8225  112510  010137  002402                  MOV     R1,SLOT         ;ELSE PUT EMPTY LOC. ADDRESS INTO SLOT
 8226  112514  000407                          BR      30$             ;RETURN
 8227  112516  012737  177777  002402  15$:    MOV     #-1,SLOT        ;PUT -1 INTO SLOT TO SHOW END OF TABLE
 8228  112524  000403                          BR      30$             ;RETURN
 8229  112526  062701  000010          20$:    ADD     #8.,R1          ;INCR. POINTER TO NEXT LOCATION
 8230  112532  000760                          BR      10$             ;CHECK NEXT LOC.
 8231  112534                          30$:    RETURN                  ;RETURN
 8232
 8233                                          ;--+
 8234                                          ;       CMPADR                          COMPAIR TWO ADDRESSES
 8235                                          ;
 8236                                          ;       THIS ROUTINE COMPAIRS TWO SIX BYTE STRINGS
 8237                                          ;
 8238                                          ;       INPUTS                  P1 - ADDRESS OF FIRST STRING
 8239                                          ;                               P2 - ADDRESS OF SECOND STRING
 8240                                          ;       OUTPUTS                 P3 - 0 = COMPARISON/-1 = NO COMPARISON
 8241                                          ;
 8242                                          ;       CALLING PROCEDURE       CALL    CMPADR P1,P2
 8243                                          ;                               P$POP   P3
 8244                                          ;
 8245                                          ;--+
 8246
 8247  112536                          CMPADR: P$POP   R2,R3           ;PUT ADDRESS OF STRING TO BE COMPARED IN R2 AND R3
 8248  112542  022223                          CMP     (R2)+,(R3)+     ;DO FIRST TWO BYTES COMPARE
 8249  112544  001006                          BNE     10$             ; IF NO, EXIT
 8250  112546  022223                          CMP     (R2)+,(R3)+     ;DO SECOND TWO BYTES COMPARE
 8251  112550  001004                          BNE     10$             ; IF NO, EXIT
 8252  112552  021213                          CMP     (R2),(R3)       ;DO LAST TWO BYTES COMPARE
 8253  112554  001002                          BNE     10$             ; IF NO, EXIT
 8254  112556  005001                          CLR     R1              ;PUT COMPARISON OK INDICATOR IN R1
 8255  112560  000402                          BR      15$
 8256  112562  012701  177777          10$:    MOV     #-1,R1          ;PUT NO COMPARISON INDICATOR IN R1
 8257  112566                          15$:    RETURN  R1
 8258
 8259                                          ;--+
 8260                                          ;       PRTNOD                          PRINT NODE TABLE
 8261                                          ;
 8262                                          ;       INPUTS                  NONE
 8263                                          ;       EXPLICIT OUTPUTS        NONE
 8264                                          ;       IMPLICIT OUTPUTS        ONE ENTRY IN THE NODE TABLE WILL BE PRINTED
 8265                                          ;       SUBORDINATE ROUTINES    NONE
 8266                                          ;       CALLING SEQUENCE        CALL PRTNOD
 8267                                          ;
 8268  112572                          PRTNOD: PRINTF  #NODADR,#STRBUF         ;PRINT NODE ADDRESS
 8269  112572  012746  002322                                                                  MOV     #STRBUF,-(SP)
 8270  112576  012746  053430                                                                  MOV     #NODADR,-(SP)
 8271  112602  012746  000002                                                                  MOV     #2,-(SP)
 8272  112606  010600                                                                          MOV     SP,R0
 8273  112610  104417                                                                          TRAP    C$PNTF
 8274  112612  062706  000006                                                                  ADD     #6,SP
 8275  112616  013702  002402                  MOV     SLOT,R2         ;MOVE SLOT ADDRESS INTO R2
 8276  112622  162702  002404                  SUB     #NODTBL,R2      ;CALCULATE NODE LOGICAL NAME
 8277  112626  006202                          ASR     R2              ;USING:    LOG. NO. =
 8278  112630  006202                          ASR     R2              ;(SLOT-#NODTAB)/8
 8279  112632  006202                          ASR     R2
```

```
112634                                 PRINTF  #LOGNAM,R2              ;PRINT LOGICAL NAME
112634  010246                                                                                 MCV     R2,-(SP)
112636  012746  053440                                                                         MOV     #LOGNAM,-(SP)
112642  012746  000002                                                                         MOV     #2,-(SP)
112646  010600                                                                                 MOV     SP,R0
112650  104417                                                                                 TRAP    C$PNTF
112652  062706  000006                                                                         ADD     #6,SP
112656  013702  002402                 MOV     SLOT,R2                 ;SEE IF TARGET OR ASSIST NODE
112662  062702  000007                 ADD     #7,R2                   ; INFO CONTAINED IN 7TH BYTE OF ENTRY
112666  111203                         MOVB    (R2),R3                 ;MOVE INTO R3
112670  020327  000001                 CMP     R3,#CASIST              ;SEE IF ASSIST NODE
112674  001404                         BEQ     10$                     ; IF YES, BR 10$
112676  012737  062613  002312         MOV     #ARGTY7,KEYWD2          ; ELSE MOVE 'TARGET' INTO KEYWD2
112704  000403                         BR      20$                     ; CONTINUE
112706  012737  062603  002312  10$:   MOV     #ARGTY6,KEYWD2          ; MOVE 'ASSIST' INTO KEYWD2
112714                          20$:   PRINTF  #NODTYP,KEYWD2          ;PRINT NODE TYPE
112714  013746  002312                                                                         MOV     KEYWD2,-(SP)
112720  012746  053453                                                                         MOV     #NODTYP,-(SP)
112724  012746  000002                                                                         MOV     #2,-(SP)
112730  010600                                                                                 MOV     SP,R0
112732  104417                                                                                 TRAP    C$PNTF
112734  062706  000006                                                                         ADD     #6,SP
112740                                 RETURN                          ;RETURN
112742                                 EXIT    TST
112742  104432                                                                                 TRAP    C$EXIT
112744  000002                                                                                 .WORD   L10015-.

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;       INSERT LOCAL STORAGE THAT IS USED ONLY
                                ;       DURING THIS TEST.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;       INSERT MESSAGES THAT ARE USED ONLY
                                ;       DURING THIS TEST.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                                       .EVEN

112746                                 ENDTST
112746                                                                         L10015:
112746  104401                                                                                 TRAP    C$ETST

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;       BEGIN THE REMAINING TESTS ON NEW PAGES.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

                                        .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.
                                        ;--

 112750                                         BGNHRD
 112750  000015                                                                         .WORD L10016-L$HARD/2
 112752                                                                         L$HARD::

                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                        ;       INSERT HARDWARE PARAMETER IN:ERPRETIVE CODE HERE.  THIS CODE
                                        ;       IS USED BY THE SUPERVISOR TO INTERROGATE THE OPERATOR FOR
                                        ;       DEVICE INFORMATION TO PUT IN THE P-TABLE.  THIS CODE IS USED
                                        ;       IN CONJUNCTION WITH THE DEFAULT P-TABLE TEMPLATE.  THE MACROS
                                        ;       US'D IN THIS SECTION ARE ''GPRMD'', ''GPRMA'' AND ''GPRML''.
                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 112752                                         GPRMA   ASKCSR,0,0,160000,177776,YES            ; GET CSR ADDRESS
 112752  000031                                                                         .WORD   T$CODE
 112754  113004                                                                         .WORD   ASKCSR
 112756  160000                                                                         .WORD   T$LOLIM
 112760  177776                                                                         .WORD   T$HILIM
 112762                                         GPRMA   ASKVEC,2,0,0,776,YES                    ; GET VECTOR ADDRESS
 112762  001031                                                                         .WORD   T$CODE
 112764  113037                                                                         .WORD   ASKVEC
 112766  000000                                                                         .WORD   T$LOLIM
 112770  000776                                                                         .WORD   T$HILIM
 112772                                         GPRMD   ASKPRI,4,0,340,0,7,YES                  ; GET PRIORITY LEVEL
 112772  002032                                                                         .WORD   T$CODE
 112774  113073                                                                         .WORD   ASKPRI
 112776  000340                                                                         .WORD   340
 113000  000000                                                                         .WORD   T$LOLIM
 113002  000007                                                                         .WORD   T$HILIM

 113004                                         ENDHRD
                                                                                        .EVEN
 113004                                                                         L10016:

                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                        ;     INSERT MESSAGES THAT ARE USED ONLY
                                        ;     DURING THE HARDWARE PARAMETER CODING SECTION.
                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 113004  044127  052101  044440  ASKCSR: .ASCIZ  /WHAT IS THE PCSRO ADDRESS?/
 113012  020123  044124  020105
 113020  041520  051123  020117
 113026  042101  051104  051505
 113034  037523     000
 113037     127  040510  020124  ASKVEC: .ASCIZ  /WHAT IS THE VECTOR ADDRESS?/
```

```
CZUACAO DEUNA NI EXERCISER DIAGNOSTIC   MACY11 30A(1052)  20-JUL-83  13:27  PAGE 163
CZUACA.P11    19-JUL-83 17:13           HARDWARE PARAMETER CODING SECTION

  8382  113044  051511  052040  042510
  8383  113052  053040  041505  047524
  8384  113060  020122  042101  051104
  8385  113066  051505  037523     000
  8386  113073     127  040510  020124  ASKPRI: .ASCIZ  /WHAT IS THE PRIORITY LEVEL?/
  8387  113100  051511  052040  042510
  8388  113106  050040  044522  051117
  8389  113114  052111  020131  042514
  8390  113122  042526  037514     000
  8391          113130                          .EVEN
  8392
```

```
                        .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.
                        ;--

113130                          BGNSFT
113130  000000                                                          .WORD L10017-L$SOFT/2
113132                                                          L$SOFT::

                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        ;       INSERT SOFTWARE PARAMETER INTERPRETIVE CODING HERE.  THIS CODE
                        ;       IS USED BY THE SUPERVISOR TO INTERROGATE THE OPERATOR FOR
                        ;       SOFTWARE INFORMATION WHICH WILL BE PLACED IN THE SOFTWARE
                        ;       TABLE.  THIS SECTION IS OPTIONAL.
                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                                .EVEN

113132                          ENDSFT
                                                                        .EVEN
113132                                                          L10017:


                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        ;     INSERT MESSAGES THAT ARE USED ONLY
                        ;     DURING THE SOFTWARE PARAMETER CODING SECTION.
                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

113132                  $PATCH::
113132  000010                  .BLKW   10

                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        ;       THIS IS A PATCH AREA THAT SHOULD BE INCLUDED IN ALL DIAGNOSTICS.
                        ;       ADJUST THE SIZE TO FIT YOUR OWN PREFERENCES.
                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

113152                          LASTAD
                                                                        .EVEN
113152  000000                                                          .WORD   0
113154  000000                                                          .WORD   0
113156                  L$LAST::
```

```
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;       HARDCODED P-TABLES MAY BE PLACED HERE BY USING THE SETUP MACROS.
                                ;       THIS SECTION IS OPTIONAL AND SHOULD BE REMOVED IF IT IS NOT BEING
                                ;       USED.  CHANGE THE POINTER MACRO ARGUMENT TO REFLECT THE REMOVAL.
                                ;
                                ;       THE P-TABLES ARE DELIMITED BY THE ''BGNSETUP'' AND ''ENDSETUP'' MACROS.
                                ;       THE ''BGNSETUP'' MACRO HAS ONE ARGUMENT WHICH IS THE NUMBER OF
                                ;       P-TABLE ENTRIES.   EACH ENTRY IS DELIMITED BY THE ''BGNPTAB'' AND
                                ;       ''ENDPTAB'' MACROS.  NEITHER OF THESE MACROS REQUIRE AN ARGUMENT.
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                                ;       BGNSETUP        1
                                ;       BGNPTAB
                                ;       .WORD   0
                                ;       ENDPTAB
                                ;       ENDSETUP

         000001                 .END
```

```
ACTALP  103450          6407    6961#
ACTBLD  101166          6396    6529#
ACTCMS  107176          6422    7602#
ACTCNL  110540          6424    7835#
ACTCNT  107302          6423    7628#
ACTCPY  103742          6416    7073#
ACTCSU  111026          6427    7917#
ACTCTT  103520          6412    7002#
ACTDFT  111102          6429    7938#
ACTDIR  104772          6428    7252#
ACTEXT  103350          6402    6926#
ACTFCT  110624          6425    7860#
ACTHLP  100620          6394    6451#
ACTIDT  102264          6401    6738#
ACTMSG  103242          6896#   7041    7063    7085
ACTNAD  104020          6417    7093#
ACTNAL  104156          6418    7130#
ACTNOD  100656          6395    6470#
ACTNUF  100610          6403    6438#
ACTNUL  100616          6393    6444#
ACTONE  103460          6408    6949#
ACTOPR  103530          6413    7010#
ACTPAT  107022          6398    7554#
ACTRNA  104356          6419    7177#
ACTRNL  105420          6420    7339#
ACTRUN  104236          6397    7152#
ACTSAV  110660          6399    7875#
ACTSMS  107104          6421    7575#
ACTSND  110364          6406    6626    7802#
ACTSR4  103442          6405    6953#
ACTSUM  101710          5830    6400    6636#
ACTSZE  103664          6415    7051#
ACTTYP  103656          6414    7041#
ACTUNS  110724          6426    7891#
ACTUSF  111150          6430    7955#
ACTXAD  103360          6404    6935#
ACTZRO  103470          6409    6978#
ACTOAL  103510          6411    6994#
ACT1AL  103500          6410    6986#
ADR   = 000020  G       1199#
ADRBUF  002314          1567#   5117    6475    6485    6501    6743    6754    6883    6936    7097    7992    8160
ADRDEL  054340          2698#   7117
ALLNOD  062151          3226#
ALPHA = 000000  G       1282#   6114    6961    7602
ANCHOR  070042          4045#   4097    4214    4244    4251    4322    4331    4340    4457    4496    4532    4546    4558
                        4559    4632    4633    4649    4694    4701    4932    4933    5015    5016    5070    5071    5130
                        5334    6027    6028    6032    6040    6063    6199    6210    6213    6475    6485    6501    6510
                        6549    6568    6572    6576    6582    6619    6654    6688    6698    6743    6754    6760    6764
                        6765    6784    6788    6812    6883    6885    6936    7020    7100    7157    7161    7165    7167
                        7181    7194    7199    7202    7203    7211    7212    7223    7240    7244    7256    7264    7278
                        7281    7302    7303    7312    7330    7345    7358    7387    7388    7397    7398    7409    7424
                        7425    7428    7429    7440    7455    7456    7460    7461    7472    7490    7504    7510    7532
                        7561    7629    7635    7650    7658    7680    7688    7710    7718    7726    7734    7742    7750
                        7758    7805    7814    7817    7818    7862    7973    7992    8005    8154
ARGTY1  062536          3282#
ARGTY2  062544          3283#
```

```
ARGTY3  062555            3285#
ARGTY4  062566            3287#
ARGTY5  062577            3289#
ARGTY6  062603            3290#   6507    7214    7400    7431    7463    8294
ARGTY7  062613            3292#   6504    7216    7402    7433    7465    8292
ASKCSR  113004            8352    8376#
ASKPRI  113073            8362    8386#
ASKVEC  113037            8357    8381#
ASSEMB= 000010             628
B     = 000007            1758#   1765    1766#   1771    1772#   1777    1778#   1783    1784#   1789    1790#   1795    1796#
                          1801#   1808    1809#   1814    1815#   1820    1821#   1826    1827#   1832    1833#   1838    1839#
BAD     112104            7973    8047*   8060*   8081#
BCOUNT  050526 G          2253#   4124    4151*   4421*
BINDEC  075014 G          5396#   6688    6698    7650    7658    7680    7688    7710    7718    7726    7734    7742    7750
                          7758
BINHEX  072672            4846#   6032    6501    6654    6812    7211    7212    7281    7397    7398    7428    7429    7460
                          7461    7635    7817
BIT0  = 000001 G          1172#
BIT00 = 000001 G          1161#   1172    1311    1313    1315    1319    1321    1322    1340    1342    1344    1346    1368
                          1371    1374    1377    1380    1383    1387    1390
BIT01 = 000002 G          1160#   1171    1312    1313    1320    1322    1341    1342    1345    1346    1365    1370    1371
                          1376    1377    1382    1383    1389    1390
BIT02 = 000004 G          1159#   1170    1314    1315    1321    1322    1344    1345    1346    1367    1368    1370    1371
                          1379    1380    1382    1383
BIT03 = 000010 G          1158#   1169    1318    1319    1320    1321    1322    1335    1373    1374    1376    1377    1379
                          1380    1382    1383
BIT04 = 000020 G          1157#   1168    1386    1387    1389    1390
BIT05 = 000040 G          1156#   1167    1306
BIT06 = 000100 G          1155#   1166    1305
BIT07 = 000200 G          1154#   1165    1304    1333
BIT08 = 000400 G          1153#   1164    1303    1439
BIT09 = 001000 G          1152#   1163    1440
BIT1  = 000002 G          1171#
BIT10 = 002000 G          1151#   1301    1441    1450    6066
BIT11 = 004000 G          1150#   1300    1442    1451    1478    6068    6077
BIT12 = 010000 G          1149#   1299    1443    1452    1479    6099
BIT13 = 020000 G          1148#   1298    1480    1487
BIT14 = 040000 G          1147#   1297    1330    1445    1454
BIT15 = 100000 G          1146#   1296    1329    1446    1455
BIT2  = 000004 G          1170#
BIT3  = 000010 G          1169#
BIT4  = 000020 G          1168#
BIT5  = 000040 G          1167#
BIT6  = 000100 G          1166#
BIT7  = 000200 G          1165#
BIT8  = 000400 G          1164#
BIT9  = 001000 G          1163#
BLDAST  073606 G          5033#   7387    7424
BLDBUF  074324 G          4932    5015    5070    5207#
BLDFAS  073242 G          4974#   7202    7455
BLDLD   072760 G          4899#   7302
BLDMSG  052757            2564#   6610
BLDREQ  074124 G          5111#   6764
BOE   = 000400 G          1203#
BOOT  = 000005 G          1315#
BUFL  = 100000 G          1455#
```

```
BUFLEN  050566 G        2269#   4527    4562    4911*   4987*   5046*   5116*   7513    7520
BUILD = 000003          1246#   2363
CADERR  053306          2602#   7995
CADRER  053232          2594#   6478    6940
CALPHA= 000016          1257#   2447
CASIST= 000001 G        1215#   5789    6502    6503    7384    8290
CBOADR  002366          1572#   6475    6936    6953*   7020    7959    7964
CCCITT= 000023          1262#   2457
CCITT = 000005 G        1287#   7002
CCLMSG= 000035          1272#   2426
CCLNAD= 000004 G        1217#
CCLNAL= 000010 G        1218#
CCLSUM= 000042          1277#   2428
CCNTR = 000036          1273#   2407
CCPYS = 000027          1266#   2468
CDEFLT= 000044          1279#   2482
CDIR  = 000043          1278#   2479    7162    7252
CEXADR= 000013          1254#   2424    2438
CEXIT = 000020 G        1219#   3771    5909    6376    6926
CFLAG   003672          1701#   3771*   5773*   5789*   5794*   5909    5911*   6356*   6376    6502    6926*   7016    7021
CFUNCT= 000040          1275#   2491
CLIACT  100456          6355    6385#
CLIALN= 000007          1236#   2437
CLIALP= 000006          1235#
CLIBIF= 000003          1232#   2397
CLIBR = 000002          1231#   2362    2366    2369    2377    2379    2381    2383    2387    2394    2395    2404    2406
                        2408    2409    2420    2423    2424    2425    2427    2429    2436    2438    2448    2450    2452
                        2454    2456    2458    2460    2461    2465    2469    2470    2476    2478    2480    2482
CLIBRX  052554          2540#   5596
CLIDEC= 000011          1238#   2422    2464    2468    2485
CLIERM  052445          2526#   6361
CLIERR= 000000          1229#   2374    2388    2430
CLIEXI= 000001          1230#   2358    2360    2364    2371    2373    2375    2385    2389    2398    2410    2431    2439
                        2486    2492    2498    2499
CLINBG  052527          2536#   5640
CLINUF  052476          2531#   6370
CLINUM= 000005          1234#
CLIOCT= 000010          1237#   2491
CLISPA= 000004          1233#   2356    2393    2402    2414    2416    2435    2443    2474    2490    2496
CLISTR= 000012          1239#   2359    2361    2363    2365    2368    2370    2372    2376    2378    2380    2382    2384
                        2386    2403    2405    2407    2415    2419    2426    2428    2445    2447    2449    2451    2453
                        2455    2457    2462    2466    2475    2477    2479    2481    2483
CLITRE  051070          2352#   6354
CLI$PM  052436          2524#   6348
CLKBR   003676          1705#   6016    6128
CLKCSR  003674          1704#   3988*   4009*   5943    5960*   5961*   5962*   6022*   6134*   6214*
CLKEN   003704          1708#   4009    5951*   5963*   6022    6134
CLKHZ   003702          1707#   3991    4006    6014    6126
CLKINT  067702 G        3986#   6017    6129
CLKSET  067656          3941#   5950    5959
CLKVEC  003700          1706#   6018    6130
CLRCNT= 000013 G        1377#
CLRMSG  053716          2651#   7606
CLRSTA= 000017 G        1383#
CLUPPR= 000033          1270#   2475    7158    7339
CMDBUF  002200          1564#   6346    6353
```

```
CMDTY1  062444            3268#
CMDTY2  062451            3269#
CMDTY3  062461            3271#
CMDTY4  062467            3273#
CMDTY5  062474            3274#
CMDTY6  062500            3275#   6899
CMDTY7  062510            3277#   6512
CMDTY8  062515            3278#
CMDTY9  062523            3280#
CMPADR  112536            5334    6485    6576    6743    6754    7100    8247#
CMPBUF  050570 G          2270#   4930*   5013*   5068*   7532
CMPER1  066754            3699#   5270
CMPER2  067051            3710#   5283
CMPSTR  072530            4784#   4795    4799    4808    4812
CNDADR= 000030            1267#   2425
CNDLOG= 000037            1274#   2423
CNODAL= 000031            1268#   2419
CNODE = 000015            1256#   2403
CNTR00  062703            3309#   7637
CNTR01  062763            3318#   7644
CNTR02  063032            3325#   7652
CNTR03  063066            3330#   7660
CNTR04  063133            3337#   7667
CNTR05  063210            3345#   7674
CNTR06  063260            3352#   7682
CNTR07  063317            3358#   7690
CNTR08  063367            3365#   7697
CNTR09  063442            3373#   7704
CNTR10  063513            3380#   7712
CNTR11  063552            3386#   7720
CNTR12  063622            3393#   7728
CNTR13  063670            3400#   7736
CNTR14  063735            3407#   7744
CNTR15  063771            3412#   7752
CNTR16  064033            3418#   7760
CNTR17  064101            3425#   7767
CNTR18  064154            3433#   7774
CNTR19  064221            3440#   7781
COMAND  071350 G          4214    4331    4452#   4496    4532
COMPAR  062351            3253#   7535
CONES = 000017            1258#   2449
COPRSL= 000024            1263#   2460
COUNT   050542 G          2259#   5214*   5215*   5217    5220*
CPATRN= 000005            1248#   2481    7289    7554
CPYCNT  050562 G          2267#   7195*   7237*   7276*   7327*   7356*   7486*
CPYLMT  053775            2659#   7080
CRC   = 004000 G          1478#
CRNALL= 000032            1269#   2477    7154    7177
CRUN  = 000004            1247#   2486
CSAVE = 000006            1249#   2372
CSAVR4= 000014            1255#   2394    2417    2436    2459
CSHCTR= 000002 G          1216#
CSHMSG= 000034            1271#   2405
CSIZE = 000026            1265#   2464
CTARGT= 000000 G          1214#   5773    6505    7021    7362    7367
CTYPE = 000025            1264#   2461
```

CZUACA0 DEUNA NI EXERCISER DIAGNOSTIC MACY11 30A(1052) 20-JUL-83 13:27 PAGE 171
CZUACA.P11 19-JUL-83 17:13 CROSS REFERENCE TABLE -- USER SYMBOLS

```
CUNSAV= 000041          1276#  2382
CUNSVF= 000045          1280#  2498
CZEROS= 000020          1259#  2451
C$AU  = 000052           628#  6309
C$AUTO= 000061           628#  6179
C$BRK = 000022           628#  6564  6780  7500
C$BSEG= 000004           628#
C$BSUB= 000002           628#
C$CEFG= 000045           628#
C$CLCK= 000062           628#  5946  5955
C$CLEA= 000012           628#  6236
C$CLOS= 000035           628#  7971
C$CLP1= 000006           628#
C$CVEC= 000036           628#
C$DCLN= 000044           628#  3780  6140
C$DODU= 000051           628#
C$DRPT= 000024           628#
C$DU  = 000053           628#  6272
C$EDIT= 000003           628#  1010
C$ERDF= 000055           628#  4127  4133  4140  4147  4191  4217  4325  4334  4343  4568  4574  4580
                        4651   4935  4941  5018  5024  5073  5079  5132  6119  6870  6877
C$ERHR= 000056           628#  4587  6203  6552  6622  7206  7226  7306  7315  7391  7412  7443  7475
                        7789   7865
C$ERRO= 000060           628#
C$ERSF= 000054           628#
C$ERSO= 000057           628#
C$ESCA= 000010           628#
C$ESEG= 000005           628#
C$ESUB= 000003           628#
C$ETST= 000001           628#  8321
C$EXIT= 000032           628#  6142  6219  6379  8304
C$GETB= 000026           628#  8050
C$GETW= 000027           628#
C$GMAN= 000043           628#  6344
C$GPHR= 000042           628#  5974
C$GPLO= 000030           628#
C$GPRI= 000040           628#
C$INIT= 000011           628#  6159
C$INLP= 000020           628#
C$MANI= 000050           628#
C$MEM = 000031           628#  5939
C$MSG = 000023           628#  3783  3808  3828
C$OPEN= 000034           628#  7960
C$PNTB= 000014           628#  6599  7220  7235  7287  7299  7325  7406  7421  7437  7452  7469  7484
C$PNTF= 000017           628#  4642  5363  5599  5643  5801  5969  6364  6373  6457  6481  6491  6497
                        6646   6659  6665  6677  6683  6694  6703  6749  6774  6798  6817  6825  6833
                        6841   6849  6857  6865  6907  6916  6943  7027  7061  7083  7109  7120  7141
                        7271   7352  7373  7379  7582  7591  7609  7618  7640  7647  7655  7663  7670
                        7677   7685  7693  7700  7707  7715  7723  7731  7739  7747  7755  7763  7770
                        7777   7784  7811  7825  7852  7883  7903  7929  7968  7981  7989  7998  8014
                        8058   8195  8273  8285  8300
C$PNTS= 000016           628#  6037  6049  6055  6074  6083  6090  6097  6105  6112  6519  6533  6539
                        6545   6613  7189
C$PNTX= 000015           628#  3777  3796  3803  3817  3824  5273  5286  7524
C$QIO = 000377           628#
C$RDBU= 000007           628#
```

```
C$REFG= 000047            628#  5915    5920    5926    5932
C$RESE= 000033            628#
C$REVI= 000003            628#  1009
C$RFLA= 000021            628#
C$RPT = 000025            628#  5849
C$SEFG= 000046            628#
C$SPRI= 000041            628#  6025    6137    6217
C$SVEC= 000037            628#  5987    6020    6132
C$TPRI= 000013            628#
C.COLL= 000074 G         1554#
C.MREC= 000010 G         1538#
C.MXMT= 000040 G         1546#
C.PREC= 000004 G         1537#
C.PXMD= 000054 G         1549#
C.PXMT= 000034 G         1545#
C.PXM2= 000050 G         1548#
C.PXM3= 000044 G         1547#
C.RDAT= 000020 G         1541#
C.RERB= 000014 G         1539#
C.RERR= 000016 G         1540#
C.RLEX= 000032 G         1544#
C.RLIN= 000030 G         1543#
C.RMDB= 000024 G         1542#
C.SECS= 000002 G         1536#
C.XABB= 000066 G         1552#
C.XABT= 000070 G         1553#
C.XDAT= 000060 G         1550#
C.XMDB= 000064 G         1551#
COALT = 000022           1261#  2455
C1ALT = 000021           1260#  2453
DATCMP  074406 G         5253#  7532
DECSTR  075200 G         5401   5446#  6689  6699  7651  7659  7681  7689  7711  7719  7727  7735  7743
                         7751   7759
DEF   = 002000 G         1441#  4550
DEPADR  047752 G         1931#
DESTIN= 000000 G         1399#  4912*  4913*  4914*  4988*  4989*  4990*  5047*  5048*  5049*  5118*  5119*  5120*
DFPTBL  002170 G         1104#
DIAGMC= 000000            628
DIRCOM  105022           7181   7256   7261#  7561
DIRECT  062173           3230#  7283
DMPMEM= 000020 G         1386#  6063
DNI   = 004000 G         1300#  4186   4197   4409   4417
DNIFLG  050522 G         2251#  4089   4091*  4411*
EDPACK  072322           4689#  6475   6936   7992
EF.CON= 000036 G         1179#  5925
EF.NEW= 000035 G         1180#  5931
EF.PWR= 000034 G         1181#
EF.RES= 000037 G         1178#  5919
EF.STA= 000040 G         1177#  5914
EMSG0   003464 G         1644   1676#
EMSG01  064272           3452#  4135
EMSG02  064321           3456#  4142
EMSG03  064341           3459#  4149
EMSG04  064373           3464#  4129
EMSG05  064415           3468#  4219
EMSG06  064450           3473#  4345
```

CZUACA0 DEUNA NI EXERCISER DIAGNOSTIC MACY11 30A(1052) 20-JUL-83 13:27 PAGE 173
CZUACA.P11 19-JUL-83 17:13 CROSS REFERENCE TABLE -- USER SYMBOLS

```
EMSG07  064510            3479#   4336
EMSG08  064553            3485#   4589
EMSG09  064613            3491#   4570
EMSG1   003465 G          1645    1678#
EMSG10  064656            3497#   4576    4582    4937    5020    5075    5134
EMSG11  064716            3503#   4653
EMSG14  064755            3509#   4943    5026    5081
EMSG15  065031            3517#   4193
EMSG16  065064            3522#   4327
EMSG18  065127            3528#   6121
EMSG19  065176            3535#
EMSG2   003466 G          1646    1680#
EMSG20  065234            3540#
EMSG22  065266            3545#   6795
EMSG23  065321            3550#   6872
EMSG24  065366            3557#   6879    7208    7308    7393
EMSG25  065442            3565#   6205    6554    6624
EMSG3   003467 G          1647    1682#
EMSG30  065531            3575#   7867
EMSG31  065576            3582#   7791
EMSG32  065637            3588#   7370
EMSG33  065714            3596#
EMSG34  065732            3599#   7317
EMSG35  066002            3606#   3793
EMSG36  066037            3611#   3800
EMSG37  066064            3615#   3814
EMSG38  066130            3621#   3821
EMSG4   003470 G          1648    1684#
EMSG40  066174            3627#   7414
EMSG41  066263            3637#   7445
EMSG42  066353            3647#   7228    7477
EMSG5   003570 G          1649    1697#
ENP   = 000400 G          1439#   4529
ENTRND  112322            6510    8005    8153#
EOFF    112106            7975    8048*   8073*   8082#
EOFFND  112223            7978    8098#
ERRBLK  052434 G          2516#
ERRFLG  050530 G          2254#   4087    4118*   4422*
ERRMSG  052432 G          2515#
ERRNBR  052430 G          2514#
ERROR   070150 G          4097    4118#
ERRS  = 040000 G          1445#   4543    4635
ERRTYP  052426 G          2513#
ERR1    067460 G          3770#   4130    4136    4143    4150    4194    4220    4328    4337    4346    4571    4577    4583
                          4590    4654    4938    4944    5021    5027    5076    5082    5135    6122    6873    6880
ERR2    067516 G          3786#   7318
ERR3    067604 G          3811#   7229    7415    7446    7478
EVL   = 000004 G          1197#
EXIT  = 000011            1252#   2384
E$END = 002100             628#
E$LOAD= 000035             628#   1033
FAADR1= 000022 G          1526#   4997*   4998*   4999*   5056*   5057*   5058*
FAADR2= 000032 G          1528#   5001*   5002*   5003*   5060*   5061*   5062*
FAADR3= 000042 G          1530#   5005*   5006*   5007*   5064*   5065*   5066*   5067
FAADR4= 000052 G          1532#   5009*   5010*   5011*   5012
FAFCT1= 000020 G          1525#   4996*   5055*
```

```
FAFCT2= 000030 G        1527#   5000*   5059*
FAFCT3= 000040 G        1529#   5004*   5063*
FAFCT4= 000050 G        1531#   5008*
FASIST  051026 G        2325#
FASKIP= 000016 G        1524#   4995*   5054*
FATFLG  050512 G        2247#   4122    4144*   4393*
FATI  = 000400 G        1303#   4390
FILENM  111544          8078#
FILLIN  111560          7985    7992    8046    8079#
FINDSL  112400          6582    8154    8184#
FRAM  = 020000 G        1480#
FRDADR= 000004 G        1499#
FREMEM  047662 G        1884#   5941*   5942*
FRESIZ  047660 G        1883#   5940*   5941
FULAST  052207          3233#
FULSLT  112470          7194    7199    7244    7264    7278    7345    7358    7805    7814    8220#
FUNCT   071372 G        4322    4340    4484#   6028    6040    6063    6199    6549    6619    6760    6885    7629    7862
FUNTAB  047676 G        1898#   4486
F$AU  = 000015           628#   6282    6308
F$AUTO= 000020           628#   6170    6178
F$BGN = 000040           628#   3770    3786    3811    3986    5818    5834    5858    5885    6142    6170    6188    6219
                        6245    6257    6282    6294    6334    6379    8304    8320    8339    8405
F$CLEA= 000007           628#   6188    6235
F$DU  = 000016           628#   6245    6271
F$END = 000041           628#   3784    3809    3829    4013    5832    5850    6142    6160    6180    6219    6237    6255
                        6273    6292    6310    6334    6379    8304    8320    8322    8370    8420
F$HARD= 000004           628#   8339    8368
F$HW  = 000013           628#   1102    1112
F$INIT= 000006           628#   5885    6158
F$JMP = 000050           628#   5832    6142    6219    6255    6292    6379    8304
F$MOD = 000000           628#
F$MSG = 000011           628#   3770    3782    3786    3807    3811    3827
F$PROT= 000021           628#   5858    5865
F$PWR = 000017           628#
F$RPT = 000012           628#   5818    5848
F$SEG = 000003           628#
F$SOFT= 000005           628#   8405    8418
F$SRV = 000010           628#   3986    4011
F$SUB = 000002           628#
F$SW  = 000014           628#   1124    1130
F$TEST= 000001           628#   6335    8320
GETCL   100274          6341#   6366    6375    6377
GETCOM  074304          5166    5171#
GETFNT= 000002 G        1312#   4496
GETPCB= 000001 G        1311#   4214
GETRNX  074262 G        4251    4632    4633    4649    5163#   6213    6572    6788    7510
GETXNX  074274 G        4244    4546    4558    4559    4933    5016    5071    5130    5168#
G$CNTO= 000200           628#
G$DELM= 000372           628#
G$DISP= 000003           628#
G$EXCP= 000400           628#
G$HILI= 000002           628#
G$LOLI= 000001           628#
G$NO  = 000000           628#   6347
G$OFFS= 000400           628#   6347    8351    8356    8361
G$OFSI= 000376           628#   6347    8351    8356    8361
```

CZUACAO DEUNA NI EXERCISER DIAGNOSTIC MACY11 30A(1052) 20-JUL-83 13:27 PAGE 175
CZUACA.P11 19-JUL-83 17:13 CROSS REFERENCE TABLE -- USER SYMBOLS

```
G$PRMA= 000001            628#  8351  8356
G$PRMD= 000002            628#  6347  8361
G$PRML= 000000            628#
G$RADA= 000140            628#  6347
G$RADB= 000000            628#
G$RADD= 000040            628#
G$RADL= 000120            628#
G$RADO= 000020            628#  8351  8356  8361
G$XFER= 000004            628#
G$YES = 000010            628#  8351  8356  8361
HDMSG1  055342           2791#  6034
HDMSG2  055413           2798#  6046
HDMSG3  055466           2806#  6052
HDMSG4  055522           2811#  6071
HDMSG5  055577           2819#  6080
HDMSG6  055650           2826#  6087
HDMSG7  055710           2832#  6094
HDMSG8  055751           2838#  6102
HDMSG9  056014           2844#  6109
HEADER= 000016 G         1397#  5126*  5127*  5128*
HELP  = 000001            627#   628   963  1051  1060  1073  1078  1080  1092  1110  1128  1134  1141
                         1244#  2357  2359  2500  3830  3856  3863  3869  3876  3882  3889  3895  3903
                         3911   3916  5820  5835  5866  5887  6145  6172  6190  6222  6247  6258  6284
                         6295   6314  6324  6329  6336  8307  8323  8342  8371  8408  8422  8430  8442
HELP1   056060           1602  2851#
HELP10  057101           1611  2943#
HELP11  057172           1612  2953#
HELP12  057270           1613  2964#
HELP13  057375           1614  2976#
HELP14  057474           1615  2987#
HELP15  057573           1616  2998#
HELP16  057676           1617  3010#
HELP17  057765           1618  3020#
HELP18  060070           1619  3032#
HELP19  060140           1620  3039#
HELP2   056161           1603  2862#
HELP20  060243           1621  3051#
HELP21  060321           1622  3059#
HELP22  060404           1623  3068#
HELP23  060505           1624  3079#
HELP24  060605           1625  3090#
HELP25  060735           1626  3105#
HELP26  061021           1627  3114#
HELP27  061125           1628  3126#
HELP28  061227           1629  3138#
HELP29  061346           1630  3152#
HELP3   056254           1604  2872#
HELP30  061416           1631  3159#
HELP4   056325           1605  2879#
HELP5   056376           1606  2886#
HELP6   056476           1607  2897#
HELP7   056611           1608  2910#
HELP8   056722           1609  2923#
HELP9   057012           1610  2933#
HEXBIN  072552           4701  4789#
HEXC    072650           4841#  4860  4861
```

```
HLPEND  003260            1632#   6459
HLPTAB  003164            1602#   6452
HN      072526            4783#   4790*   4793*   4818
HOE   = 100000 G          1210#
HXFORM  072414            4694    4726#
HXN     072412            4725#   4727*   4731
IBE   = 010000 G          1207#
ICAB  = 040000 G          1330#
IDENT = 000010            1251#   2439
IDU   = 000040 G          1200#
IER   = 020000 G          1208#
ILADMS  053051            2574#   6488    6746
ILADM1  053135            2583#   6494
ILLADR  002650            1583#   6485    6743
INICLN  100130            5912    5971    5978    6123    6139#
INIEXI  100132            5936    6138    6141#
INIT    076600            5909#
INIT1   076620            5910    5913#
INTE  = 000100 G          1305#   4204    4454
INTR  = 000200 G          1304#
ISR   = 000100 G          1201#
IXE   = 004000 G          1206#
I$AU  = 000041             628#   6282#   6310#
I$AUTO= 000041             628#   6170#   6180#
I$CLN = 000041             628#   6188#   6219    6237#
I$DU  = 000041             628#   6245#   6273#
I$HRD = 000041            8339#   8370#
I$INIT= 000041             628#   5885#   6142    6160#
I$MOD = 000041             628#
I$MSG = 000041             628#   3770#   3784#   3786#   3809#   3811#   3829#
I$PROT= 000040             628#   5858#
I$PTAB= 000041             628#
I$PWR = 000041             628#
I$RPT = 000041             628#   5818#   5850#
I$SEG = 000041             628#   6334
I$SETU= 000041             628#
I$SFT = 000041            8405#   8420#
I$SRV = 000041             628#   3986#   4013#
I$SUB = 000041             628#   6334
I$TST = 000041             628#   6334#   6379    8304    8320#   8322#
J$JMP = 000167             628#   5832    6255    6292
KEYWD1  002310            1565#   6512*   6899*   7154    7158    7162    7177*   7252*   7289    7339*   7554*
KEYWD2  002312            1566#   6504*   6507*   6515    7516*   7535*   7539*   7544*   8292*   8294*   8296
LCAR  = 004000 G          1451#
LCLKEN= 000100 G          1223#   5951
LCOL  = 010000 G          1452#
LDADR1= 000022 G          1517#   4922*   4923*   4924*
LDADR2= 000032 G          1519#   4926*   4927*   4928*   4929
LDFCT1= 000020 G          1516#   4921*
LDFCT2= 000030 G          1518#   4925*
LDMEM = 000021 G          1387#
LDRESP  052574            2543#
LDSKIP= 000016 G          1515#   4920*
LENGTH  062342            3251#   7516
LGERMS  067117            3717#   7521
LOCDST  072206            4686#   4694    4701
```

```
LOE    = 040000 G          1209#
LOGDEL   054426            2707#   7849
LOGNAM   053440            2619#   8282
LOPDIR   050720 G          2288#   4919
LOT    = 000010 G          1198#
LST      072670            4844#   4847*   4849*   4863
LUPAIR   062162            3228#
L$ACP    002110 G          1040#
L$APT    002036 G           998#
L$AU     100266 G          6282#
L$AUT    002070 G          1024#
L$AUTO   100140 G          1041    6170#
L$CCP    002106 G          1038#
L$CLEA   100142 G          1039    6188#
L$CO     002032 G           994#
L$DEPO   002011 G           976#
L$DESC   002130 G          1031    1064#
L$DESP   002076 G          1030#
L$DEVP   002060 G          1016#
L$DISP   002164 G          1001    1089#
L$DLY    002116 G          1046#
L$DTP    002040 G          1000#
L$DTYP   002034 G           996#
L$DU     100260 G          6245#
L$DUT    002072 G          1026#
L$DVTY   002122 G          1017    1056#
L$EF     002052 G          1011#
L$ENVI   002044 G          1004#
L$ERRT   052426 G          2512#
L$ETP    002102 G          1034#
L$EXP1   002046 G          1006#
L$EXP4   002064 G          1020#
L$EXP5   002066 G          1022#
L$HARD   112752 G           983    8339    8340#
L$HIME   002120 G          1048#
L$HPCP   002016 G           982#
L$HPTP   002022 G           986#
L$HW     002170 G           987    1102    1103#
L$ICP    002104 G          1036#
L$INIT   076600 G          1037    5885#
L$LADP   002026 G           990#
L$LAST   113156 G           991    8439#
L$LOAD   002100 G          1032#
L$LUN    002074 G          1028#
L$MREV   002050 G          1008#
L$NAME   002000 G           965#
L$PRIO   002042 G          1002#
L$PROT   076572 G          1043    5858#
L$PRT    002112 G          1042#
L$REPP   002062 G          1018#
L$REV    002010 G           974#
L$RPT    076560 G          1019    5818#
L$SOFT   113132 G          8405    8406#
L$SPC    002056 G          1014#
L$SPCP   002020 G           984#
L$SPTP   002024 G           988#
```

```
L$STA    002030 G          992#
L$SW     002200 G         1124    1125#
L$TEST   002114 G         1044#
L$TIML   002014 G          980#
L$UNIT   002012 G          978#
L10000   002176           1102    1112#
L10001   002200           1124    1130#
L10002   067514           3782#
L10003   067602           3807#
L10004   067654           3827#
L10005   070022           4011#
L10006   076570           5833    5848#
L10010   100136           6143    6158#
L10011   100140           6178#
L10012   100256           6220    6235#
L10013   100264           6256    6271#
L10014   100272           6293    6308#
L10015   112746           6380    8305    8320#
L10016   113004           8339    8369#
L10017   113132           8405    8419#
MATCH    111764           8080#
MESPAT   062054           3215#
MESPA1   062125           3222#   7296
MORE   = 010000 G         1443#   4550
MSGAD    003316 G         1652#   5213    5219
MSGCNT   003300 G         1643#   5217
MSGPRM   054641           2733#   6904    7579
MSGTAB   0032o2           1633#   3790    6902    7293    7577    7614
MSGTY0   062372           1633    3258#
MSGTY1   062400           1634    3259#
MSGTY2   062405           1635    3260#
MSGTY3   062413           1636    3262#
MSGTY4   062420           1637    3263#
MSGTY5   062425           1638    3264#
MSGTY6   062433           1639    3266#
MSG0C    003300           1644#
MSG00    003334 G         1644    1653    1661#
MSG01    003464 G         1645    1654    1677#
MSG02    003465 G         1646    1655    1679#
MSG03    003466 G         1647    1656    1681#
MSG04    003467 G         1648    1657    1683#
MSG05    003470 G         1649    1658    1685#
MSG1     054711           2740#   6530
MSG1C    003302           1645#
MSG11    055024           2753#   6536
MSG12    055137           2766#   6542
MSG2     055177           2772#   6516
MSG2C    003304           1646#
MSG3     055235           2778#
MSG3C    003306           1647#
MSG4     055276           2784#   6913    7588    7615
MSG4C    003310           1648#
MSG5C    003312           1649#
MSG6C    003314           1650#   8133*
NCHN   = 020000 G         1487#
NEW      100112           5935    6134#
```

```
NIHLT = 000006 G          1345#
NIRCNT  050516 G          2249#   4320*   4403*   4618    4633*   4647*   6207
NIUNI = 000007 G          1346#
NOCLK   062622            3295#   5966
NOCMPR  054160            2679#   7106
NOD     053701            2648#   6595    7137    7267    7348    7821    8191
NODADR  053430            2617#   8270
NODE  = 000002            1245#   2395
NODTBL  002404            1579#   5997    6560    6574    7098    7132    7192    7262    7275    7343    7355    7803    7840
                          7876    7896    8184    8276
NODTY   002400            1577#   6505*   6508*   8001*   8165
NODTYP  053453            2621#   8297
NOD0    051070            2356#
NOD1    051074            2357#
NOD10   051154            2364#
NOD100  051720            2430#
NOD101  051722            2431#
NOD102  051724            2435#
NOD103  051730            2436#
NOD104  051734            2437#
NOD105  051740            2438#
NOD106  051744            2439#
NOD107  051746            2443#
NOD11   051156            2365#
NOD110  051752            2444#
NOD111  051756            2445#
NOD112  051772            2446#
NOD113  051776            2447#
NOD114  052012            2448#
NOD115  052016            2449#
NOD116  052032            2450#
NOD117  052036            2451#
NOD12   051170            2366#
NOD120  052052            2452#
NOD121  052056            2453#
NOD122  052072            2454#
NOD123  052076            2455#
NOD124  052112            2456#
NOD125  052116            2457#
NOD126  052132            2458#
NOD127  052136            2459#
NOD13   051174            2367#
NOD130  052142            2460#
NOD131  052146            2461#
NOD132  052152            2462#
NOD133  052166            2463#
NOD134  052172            2464#
NOD135  052176            2465#
NOD136  052202            2466#
NOD137  052220            2467#
NOD14   051200            2368#
NOD140  052224            2468#
NOD141  052230            2469#
NOD142  052234            2470#
NOD143  052240            2474#
NOD144  052244            2475#
```

```
NOD145  052264            2476#
NOD146  052270            2477#
NOD147  052302            2478#
NOD15   051212            2369#
NOD150  052306            2479#
NOD151  052324            2480#
NOD152  052330            2481#
NOD153  052346            2482#
NOD154  052352            2483#
NOD155  052366            2484#
NOD156  052372            2485#
NOD157  052376            2486#
NOD16   051216            2370#
NOD160  052400            2490#
NOD161  052404            2491#
NOD162  052410            2492#
NOD163  052412            2496#
NOD164  052416            2497#
NOD165  052422            2498#
NOD166  052424            2499#
NOD17   051234            2371#
NOD2    051100            2358#
NOD20   051236            2372#
NOD21   051250            2373#
NOD22   051252            2374#
NOD23   051254            2375#
NOD24   051256            2376#
NOD25   051272            2377#
NOD26   051276            2378#
NOD27   051316            2379#
NOD3    051102            2359#
NOD30   051322            2380#
NOD31   051340            2381#
NOD32   051344            2382#
NOD33   051362            2383#
NOD34   051366            2384#
NOD35   051402            2385#
NOD36   051404            2386#
NOD37   051424            2387#
NOD4    051116            2360#
NOD40   051430            2388#
NOD41   051432            2389#
NOD42   051434            2393#
NOD43   051440            2394#
NOD44   051444            2395#
NOD45   051450            2397#
NOD46   051454            2398#
NOD47   051456            2402#
NOD5    051120            2361#
NOD50   051462            2403#
NOD51   051476            2404#
NOD52   051502            2405#
NOD53   051520            2406#
NOD54   051524            2407#
NOD55   051544            2408#
NOD56   051550            2409#
```

```
NOD57   051554            2410#
NOD6    051134            2362#
NOD60   051556            2414#
NOD61   051562            2415#
NOD62   051576            2416#
NOD63   051602            2417#
NOD64   051606            2418#
NOD65   051612            2419#
NOD66   051624            2420#
NOD67   051630            2421#
NOD7    051140            2363#
NOD70   051634            2422#
NOD71   051640            2423#
NOD72   051644            2424#
NOD73   051650            2425#
NOD74   051654            2426#
NOD75   051672            2427#
NOD76   051676            2428#
NOD77   051714            2429#
NORESP  062302            3245#   7544
NOTNUF= 000012            1253#   2361    2365    2367    2376    2378    2380    2386
NO.NTR= 000006 G          1495#   2018    2023    4240    4247
NTBHDR  053462            2623#   7808
NULL  = 000000            1243#
NULSTR  053362            2610#   5798
N10$    051074            2356#
N100$   051456            2369    2401#
N101$   051462            2402#
N102$   051502            2403    2404#
N104$   051524            2405    2406#
N106$   051550            2407    2408#
N110$   051554            2404    2406    2408    2409#
N12$    051102            2357    2358#
N120$   051556            2377    2413#
N121$   051562            2414#
N122$   051602            2416#
N123$   051630            2418    2420#
N124$   051644            2419    2421    2423#
N126$   051650            2424#
N130$   051654            2415    2425#
N132$   051676            2426    2427#
N134$   051720            2428    2429#
N135$   051722            2420    2423    2425    2427    2429    2430#
N14$    051120            2359    2360#
N140$   051724            2379    2434#
N141$   051730            2435#
N142$   051734            2436#
N143$   051744            2438#
N16$    051140            2361    2362#
N160$   051746            2381    2442#   2461    2465    2469
N161$   051752            2443#
N162$   052016            2447    2448#
N163$   052036            2449    2450#
N164$   052056            2451    2452#
N165$   052076            2453    2454#
N166$   052116            2455    2456#
```

```
N167$   052136          2457    2458#
N168$   052146          2448    2450    2452    2454    2456    2458    2460#
N170$   052152          2445    2461#
N175$   052202          2462    2465#
N178$   052234          2444    2469#
N18$    051156          2363    2364#
N180$   052240          2366    2473#
N181$   052244          2474#
N182$   052270          2475    2476#
N183$   052306          2477    2478#
N184$   052330          2479    2480#
N185$   052346          2476    2478    2480    2481#
N186$   052352          2482#
N190$   052376          2485#
N20$    051174          2365    2366#
N200$   052400          2387    2489#
N201$   052404          2490#
N210$   052412          2383    2495#
N212$   052416          2496#
N215$   052424          2497    2498#
N22$    051216          2368    2369#
N23$    051236          2370    2371#
N24$    051252          2372    2373#
N25$    051256          2367    2375#
N26$    051276          2376    2377#
N28$    051322          2378    2379#
N29$    051344          2380    2381#
N30$    051366          2382    2383#
N31$    051404          2384    2385#
N32$    051430          2386    2387#   2397    2409    2417    2422    2437    2446    2459    2463    2464    2466    2467
                        2468    2470    2481    2483    2484    2485    2491
N80$    051434          2362    2392#
N81$    051440          2393#
N82$    051444          2394#
N90$    051450          2395    2396#
N95$    051454          2397#
OFLO  = 010000 G        1479#
OK      061661          3193#   7322
OKFU    062013          3209#   7232    7481
OKRE    061702          3196#   7418
OKTR    061746          3202#   7449
ONE   = 004000 G        1442#   4550
ONEALT= 000003 G        1285#   6986
ONES  = 000001 G        1283#   6969
OPNERR  112110          7965    8084#
OPRSEL= 000006 G        1288#   5794    7016    7018
OPSLBF  003570          1659    1698#   8125
OWN   = 100000 G        1446#   4525    4529    4541    4623    4902    4978    5037    5113
O$APTS= 000000           628#    992
O$AU  = 000000           628#   1024
O$BGNR= 000001           628#    962#   1018
O$BGNS= 000000           628#    984
O$DU  = 000000           628#   1026
O$ERRT= 000000           628#   1034
O$GNSW= 000000           628#    988
O$POIN= 000001           628#    962#   1050
```

```
O$SETU= 000000            628#   978   8437
PART    075214 G         5405*  5410*  5416*  5417   5447#
PASABT  061526           3176#  7186   7376
PATCH   050572 G         2271#
PATTRN  062262           3242#
PCBB0   047666 G         1891#  4210   4489
PCBB2   047670 G         1892#  4348   6032   6042
PCBB4   047672 G         1893#
PCBB6   047674 G         1894#
PCCALL  050564 G         2268#  3773   4040*
PCEFLG  050514 G         2248#  4120   4137*  4399*
PCEI  = 040000 G         1297#  4396   4419
PCLKCT= 001600 G         1225#  5961
PCLKEN= 000111 G         1224#  5963
PCMSG   066714 G         3692#  3774
PCSR0   047632 G         1868#  4181   4374   4382   4455*  5989*  5990
PCSR0C  047642 G         1873#  4225*  4378*
PCSR1   047634 G         1869#  5990*  5991*  5992
PCSR1C  047644 G         1874#  4226*  4379*
PCSR2   047636 G         1870#  5992*  5993*  5994
PCSR2C  047646 G         1875#  4227*  4380*
PCSR3   047640 G         1871#  5994*  5995*
PCSR3C  047650 C         1876#  4228*  4381*
PCTO  = 000200 G         1333#
PDMD  = 000010 G         1318#  4532
PFNOP = 000000 G         1364#
PHYADR  047762 G         1941#  4349   49c    4923   4924   4926   4927   4928   5005   5006   5007   5009   5010
                         5011   5060   5061   5062   5064   5065   5066   6754   7635
PLINE   112172           7986   8093#
PNOP  = 000003 G         1313#
PNT   = 001000 G         1204#
PREG14  070024           4036#  4097   4214   4244   4251   4322   4331   4340   4457   4496   4532   4546   4558
                         4559   4632   4633   4649   4694   4701   4932   4933   5015   5016   5070   5071   5130
                         5334   6027   6028   6032   6040   6063   6199   6210   6213   6475   6485   6501   6510
                         6549   6568   6572   6576   6582   6619   6654   6688   6698   6743   6754   6760   6764
                         6765   6784   6788   6812   6883   6885   6936   7020   7100   7157   7161   7165   7167
                         7181   7194   7199   7202   7203   7211   7212   7223   7240   7244   7256   7264   7278
                         7281   7302   7303   7312   7330   7345   7358   7387   7388   7397   7398   7409   7424
                         7425   7428   7429   7440   7455   7456   7460   7461   7472   7490   7504   7510   7532
                         7561   7629   7635   7650   7658   7680   7688   7710   7718   7726   7734   7742   7750
                         7758   7805   7814   7817   7818   7862   7973   7992   8005   8154
PRI   = 002000 G         1205#
PRIMLD= 000001 G         1340#
PRI00 = 000000 G         1193#  6024   6136
PRI01 = 000040 G         1192#
PRI02 = 000100 G         1191#
PRI03 = 000140 G         1190#
PRI04 = 000200 G         1189#
PRI05 = 000240 G         1108   1188#
PRI06 = 000300 G         1187#
PRI07 = 000340 G         1003   1186#  6216
PROTOT= 000014 G         1401#  4626   4918*  4994*  5053*  5124*
PROT00  050544 G         2260#  4627   4918   4994   5053
PROT02  050546 G         2261#  4629   5124
PRTNOD  112572           7818   8268#
P$ACT   003150           1593#  5516   6355*
```

```
P$AERR  003162          1599#  6738   6937*  7094
P$BUFA  003144          1591#  5469   6353*
P$CNT   003152          1594#  5619*  5626   5691*  5698*  5703
P$CPYS  002374          1575#  6116*  6910   7077*  7195   7276   7356   7585   7604*  7612
P$EXIT  075342          5474   5509#  5534   5540
P$GDBD  003161          1598#  5532*  5538*  5548   5601*  5645*  5769*  5785*  6341*  6358   6472   7012   7014*
                        7032*
P$MERR  003163          1600#  5803*  7010*  7030
P$NNUF  003160          1597#  6342*  6367   6438*  6470*  6636*  6741*  6918*  6945*  7093*  7143*  7152*  7593*
                        7620*  7794*  7802*  7846*  7860*  7891*  7910*  7917*  8008*
P$NUM   003154          1595#  5614*  5617*  5619   5620*  5622*  5626*  5628*  5634*  7051   7053   7055   7073
                        7075   7077   7153   7836   7848   7862   7946*
P$PASS  002376          1576#  7153*  7167   7169*
P$RADX  003156          1596#  5573*  5576*  5584*  5591   5624   5632
P$SIZE  002372          1574#  4905   4981   5040   5211   6115*  6911   7055*  7530   7532   7586   7603*  7613
P$TREE  003146          1592#  5470   6354*
P$TRV   075216          5468#  6357
P$TR5   075226          5471#  5482   5502   5507
P$TYPE  002370          1573#  3788   5209   6114*  6900   6961*  6969*  6978*  6986*  6994*  7002*  7018*  7291
                        7559*  7565*  7566   7569*  7575   7602*
QUIT    111014          7895   7909#
RASIST  050774 G        2312#
RBFCNT  050524 G        2252#  4415*
RCBI  = 002000 G        1301#  4413   4419
RCVBUF  050540 G        2258#  4645*
RCVERR  050536 G        2257#  4637*
RDCNTS= 000012 G        1376#  7629
RDDEFA= 000002 G        1365#  6028
RDERR   112144          8055   8089#
RDLIN   111432          7973   8040#
RDMODE= 000014 G        1379#
RDMULA= 000006 G        1370#
RDPHYA= 000004 G        1367#  4340
RDRNGS= 000010 G        1373#
RDSTA = 000016 G        1382#  6040
RDSYS = 000022 G        1389#
READY = 000002 G        1341#
RECAST  062243          3239#
RECERR  052631          2548#  4639
RECEVE  071776 G        4616#  6210   6568   6784   7504
REQID   050712 G        2277#  5125
RESET = 000000 G        1339#
RESTOR  054627          2731#  7899   8010
RESTRT  100046          5923   5929   6117   6124#
RETRY   062316          3247#  7539
RETRYS  050534 G        2256#  4523*  4554*  4555   6768   7494
RMTC  = 000010 G        1335#
RPKLEN= 002756 G        1494#  1764   1770   1776   1782   1788   1794   1807   1813   1819   1825   1831   1837
                        1852   1853   1854   1855   1856   1857
RRGCUR  003760 G        1751#  4237*  4248   4251   4620   4632   4649
RRGLST  003770 G        1755#  5165
RRGNXT  003764 G        1753#  4238*  4633   6213   6570   6572   6786   6788   7508   7510
RRGSRT  003754 G        1749#  5164
RRG001  026006 G        1726   1808   1852#
RRG002  030764 G        1727   1814   1853#
RRG003  033742 G        1728   1820   1854#
```

```
RRG004  036720 G        1729    1826    1855#
RRG005  041676 G        1730    1832    1856#
RRG006  044654 G        1731    1838    1857#
RRING   004066 G        1749    1751    1753    1755    1802#   1996    2020    4237    4238
RRNGTB  003722 G        1724#
RSET  = 000040 G        1306#   4182
RSTT  = 000015 G        1321#
RTRY  = 002000 G        1450#   4552
RTRYER  052675          2555#   6771
RUN   = 000003 G        1342#
RUNALL  104366          7157    7180#
RUNCOM  106516          7223    7312    7409    7440    7472    7494#
RUNDIR  105002          7165    7255#
RUNLUP  105430          7161    7342#
RUNPAT  107032          7167    7558#
RXI   = 020000 G        1298#   4401    4417
SAVCOM  111002          7885    7905#
SAVED   054620          2729#   7879
SAVTBL  002530          1582#   7877    7897
SELMSG  112264          7020    8124#
SERI  = 100000 G        1296#   4390    4419
SFPTBL  002200 G        1126#
SIADDR= 000042 G        1509#   6810
SIDEV = 000053 G        1510#   6859
SIECO = 000030 G        1506#   6835
SIFNCT= 000035 G        1508#   6851
SIMSG1  066440          3658#   6814
SIMSG2  066477          3664#   6822
SIMSG3  066532          3669#   6830
SIMSG4  066572          3675#   6838
SIMSG5  066613          3678#   6846
SIMSG6  066641          3682#   6854
SIMSG7  066667          3686#   6862
SIRCPT= 000022 G        1504#   6789    6819
SIUECO= 000031 G        1507#   6843
SIVERS= 000027 G        1505#   6827
SIZLMT  054061          2668#   7058
SLOT    002402 G        1578#   6560*   6584    7192*   7194    7196*   7197    7199    7202    7240    7242*   7244    7262*
                        7264    7275*   7278    7281    7302    7330*   7343*   7345    7355*   7358    7361    7490*   7803*
                        7805    7814    7817    7818*   8159    8199*   8220    8225*   8227*   8275    8287
SOURCC= 000006 G        1400#   4915*   4916*   4917*   4991*   4992*   4993*   5050*   5051*   5052*   5121*   5122*   5123*
                        6573
SOURCE  072320          4687#   4690*   4694
START   076700          5917    5937#
STATBL  002656          1586#   5328    6002    6638    7920
STATUS  050312 G        2127#
STBLEN= 000132 G        1498#   1586    6001    7919
STOP  = 000017 G        1322#
STP   = 001000 G        1440#   4529
STRBUF  002322          1570#   3792    3813    6032    6033    6501    6514    6654    6655    6812    6813    7211    7215
                        7281    7282    7397    7401    7428    7432    7460    7464    7635    7636    7817    8269
STRBU1  002344          1571#   3820    7212    7213    7398    7399    7429    7430    7461    7462
STRT  = 000004 G        1314#   4331
SUMM    053706          2649#   5359    6642    7925
SUMMRY= 000007          1250#   2370
SUMMS1  067215          3728#   6656
```

CZUACA0 DEUNA NI EXERCISER DIAGNOSTIC MACY11 30A(1052) 20-JUL-83 13:27 PAGE 186
CZUACA.P11 19-JUL-83 17:13 CROSS REFERENCE TABLE -- USER SYMBOLS

SUMMS2 067236 3731# 6662
SUMMS3 067322 3740# 6674
SUMMS4 067351 3744# 6680
SUMMS5 067434 3753# 6691
SUMMS6 067452 3756# 6700

SVCGBL= 000000 628# 965 974 976 978 980 982 984 986 988 990 992 994
996 998 1000 1002 1004 1006 1008 1011 1014 1016 1018 1020 1022
1024 1026 1028 1030 1032 1034 1036 1038 1040 1042 1044 1046 1048
1056 1064 1089 1103 1104 1125 1126 2512 3770 3786 3811 3986 5818
5858 5885 6170 6188 6245 6282 8340 8406 8439# 8440

SVCINS= 000001 628# 966 967 968 969 970 971 972 973 975 977 979 981
983 985 987 989 991 993 995 997 999 1001 1003 1005 1007
1009 1010 1012 1013 1015 1017 1019 1021 1023 1025 1027 1029 1031
1033 1035 1037 1039 1041 1043 1045 1047 1049 1057 1058 1065 1070
1088 1090 1102 1124 3773 3774 3775 3776 3777 3778 3780 3783 3792
3793 3794 3795 3796 3797 3799 3800 3801 3802 3803 3804 3808 3813
3814 3815 3816 3817 3818 3820 3821 3822 3823 3824 3825 3828 4012
4127 4128 4129 4130 4133 4134 4135 4136 4140 4141 4142 4143 4147
4148 4149 4150 4191 4192 4193 4194 4217 4218 4219 4220 4325 4326
4327 4328 4334 4335 4336 4337 4343 4344 4345 4346 4568 4569 4570
4571 4574 4575 4576 4577 4580 4581 4582 4583 4587 4588 4589 4590
4639 4640 4641 4642 4643 4651 4652 4653 4654 4935 4936 4937 4938
4941 4942 4943 4944 5018 5019 5020 5021 5024 5025 5026 5027 5073
5074 5075 5076 5079 5080 5081 5082 5132 5133 5134 5135 5267 5268
5269 5270 5271 5272 5273 5274 5282 5283 5284 5285 5286 5287 5359
5360 5361 5362 5363 5364 5596 5597 5598 5599 5600 5640 5641 5642
5643 5644 5798 5799 5800 5801 5802 5832 5833 5849 5914 5915 5917
5919 5920 5922 5925 5926 5928 5931 5932 5934 5939 5940 5945 5946
5947 5949 5954 5955 5956 5958 5966 5967 5968 5969 5970 5973 5974
5975 5977 5983 5984 5985 5986 5987 5988 6016 6017 6018 6019 6020
6021 6024 6025 6033 6034 6035 6036 6037 6038 6045 6046 6047 6048
6049 6050 6052 6053 6054 6055 6056 6071 6072 6073 6074 6075 6080
6081 6082 6083 6084 6087 6088 6089 6090 6091 6094 6095 6096 6097
6098 6102 6103 6104 6105 6106 6109 6110 6111 6112 6113 6119 6120
6121 6122 6128 6129 6130 6131 6132 6133 6136 6137 6140 6142 6143
6159 6179 6203 6204 6205 6206 6216 6217 6219 6220 6236 6255 6256
6272 6292 6293 6309 6344 6345 6346 6347 6348 6349 6350 6351 6361
6362 6363 6364 6365 6370 6371 6372 6373 6374 6379 6380 6454 6455
6456 6457 6458 6478 6479 6480 6481 6482 6488 6489 6490 6491 6492
6494 6495 6496 6497 6498 6514 6515 6516 6517 6518 6519 6520 6530
6531 6532 6533 6534 6536 6537 6538 6539 6540 6542 6543 6544 6545
6546 6552 6553 6554 6555 6564 6595 6596 6597 6598 6599 6600 6608
6609 6610 6611 6612 6613 6614 6622 6623 6624 6625 6642 6643 6644
6645 6646 6647 6655 6656 6657 6658 6659 6660 6662 6663 6664 6665
6666 6671 6672 6673 6674 6675 6676 6677 6678 6680 6681 6682 6683
6684 6689 6690 6691 6692 6693 6694 6695 6699 6700 6701 6702 6703
6704 6746 6747 6748 6749 6750 6771 6772 6773 6774 6775 6780 6795
6796 6797 6798 6799 6813 6814 6815 6816 6817 6818 6821 6822 6823
6824 6825 6826 6829 6830 6831 6832 6833 6834 6837 6838 6839 6840
6841 6842 6845 6846 6847 6848 6849 6850 6853 6854 6855 6856 6857
6858 6861 6862 6863 6864 6865 6866 6870 6871 6872 6873 6877 6878
6879 6880 6904 6905 6906 6907 6908 6910 6911 6912 6913 6914 6915
6916 6917 6940 6941 6942 6943 6944 7024 7025 7026 7027 7028 7058
7059 7060 7061 7062 7080 7081 7082 7083 7084 7106 7107 7108 7109
7110 7117 7118 7119 7120 7121 7137 7138 7139 7140 7141 7142 7186
7187 7188 7189 7190 7206 7207 7208 7209 7213 7214 7215 7216 7217

			7218	7219	7220	7221	7226	7227	7228	7229	7232	7233	7234	7235	7236
			7267	7268	7269	7270	7271	7272	7282	7283	7284	7285	7286	7287	7288
			7295	7296	7297	7298	7299	7300	7306	7307	7308	7309	7315	7316	7317
			7318	7322	7323	7324	7325	7326	7348	7349	7350	7351	7352	7353	7370
			7371	7372	7373	7374	7376	7377	7378	7379	7380	7391	7392	7393	7394
			7399	7400	7401	7402	7403	7404	7405	7406	7407	7412	7413	7414	7415
			7418	7419	7420	7421	7422	7430	7431	7432	7433	7434	7435	7436	7437
			7438	7443	7444	7445	7446	7449	7450	7451	7452	7453	7462	7463	7464
			7465	7466	7467	7468	7469	7470	7475	7476	7477	7478	7481	7482	7483
			7484	7485	7500	7519	7520	7521	7522	7523	7524	7525	7579	7580	7581
			7582	7583	7585	7586	7587	7588	7589	7590	7591	7592	7606	7607	7608
			7609	7610	7612	7613	7614	7615	7616	7617	7618	7619	7636	7637	7638
			7639	7640	7641	7643	7644	7645	7646	7647	7648	7651	7652	7653	7654
			7655	7656	7659	7660	7661	7662	7663	7664	7666	7667	7668	7669	7670
			7671	7673	7674	7675	7676	7677	7678	7681	7682	7683	7684	7685	7686
			7689	7690	7691	7692	7693	7694	7696	7697	7698	7699	7700	7701	7703
			7704	7705	7706	7707	7708	7711	7712	7713	7714	7715	7716	7719	7720
			7721	7722	7723	7724	7727	7728	7729	7730	7731	7732	7735	7736	7737
			7738	7739	7740	7743	7744	7745	7746	7747	7748	7751	7752	7753	7754
			7755	7756	7759	7760	7761	7762	7763	7764	7766	7767	7768	7769	7770
			7771	7773	7774	7775	7776	7777	7778	7780	7781	7782	7783	7784	7785
			7789	7790	7791	7792	7808	7809	7810	7811	7812	7821	7822	7823	7824
			7825	7826	7848	7849	7850	7851	7852	7853	7865	7866	7867	7868	7879
			7880	7881	7882	7883	7884	7899	7900	7901	7902	7903	7904	7925	7926
			7927	7928	7929	7930	7959	7960	7962	7964	7965	7966	7967	7968	7969
			7971	7978	7979	7980	7981	7982	7985	7986	7987	7988	7989	7990	7995
			7996	7997	7998	7999	8010	8011	8012	8013	8014	8015	8050	8051	8053
			8055	8056	8057	8058	8059	8191	8192	8193	8194	8195	8196	8269	8270
			8271	8272	8273	8274	8281	8282	8283	8284	8285	8286	8296	8297	8298
			8299	8300	8301	8304	8305	8321	8339	8351	8352	8353	8354	8356	8357
			8358	8359	8361	8362	8363	8364	8365	8368	8405	8418	8436	8437	8438
SVCSUB=	000001		628#												
SVCTAG=	000001		628#	1112	1130	3782	3807	3827	4011	5848	6158	6178	6235	6271	6308
			6352	8320	8369	8419									
SVCTST=	000001		628#	6334											
S$LSYM=	010000		628#	1113#	1131#	3783#	3808#	3828#	4012#	5849#	6159#	6179#	6236#	6272#	6309#
			6345	6352	6353#	8321#	8370#	8420#							
S.BYTE	050506	G	2240#	5346	5369*	6010*	7530*								
S.COMP	050504	G	2239#	5345	5368*	6009*	7534*								
S.LEN	050502	G	2238#	5344	5367*	6008*	7515*								
S.NREC	050500	G	2237#	5342	5365*	6006*	6800*	7543*							
S.REC	050476	G	2236#	5343	5366*	6007*	6807*	7507*							
S.XFER	050510	G	2241#	5350	5370*	6011*	6808*	7506*							
TABCLR	054514		2716#	7138	7926										
TABEMT	053632		2641#	6643	7268	7349	7822								
TABFUL	053560		2634#	5360	6596	8192									
TASIST	050742	G	2299#												
TBLLEN=	000050		1496#	1579	1582	5996	7131	7905							
TEMP	050550	G	2262#	5258*	5263*	5264	5279	5282	5289	5323*	5354*	6059	6065	6556*	6591*
			6609	6615*											
TEMP1	050552	G	2263#	6557*	6588	6589*	6590*	7342*							
TEMP2	050554	G	2264#	6558*	6604*	6605	6608	6756*	6791*						
TEMP3	050556	G	2265#	6559*	6592*	6602*									
TENPWR	075130		5402	5403	5425#										
TIMERS	003720		1716#	4004	4008*	6561	6777	7496							
TIMER1	003714		1714#	3998	4000*	4084									

```
TIMER2  003716            1715#   4001    4003*   4534
TIMMIN  003706            1710#   3995*   6012*   6124*
TIMOUT  050532 G          2255#   4585*
TIMSEC  003710            1711#   3992*   3993    3996*   6013*   6125*
TIMTCK  003712            1712#   3989*   3991*   4006    6014*   6126*
TMRF  = 000012 G          1320#
TMRO  = 000011 G          1319#
TRAST   062223            3236#
TRVACT  075344            5504    5514#   5531    5537    5543    5547    5568    5636    5660    5681    5707
TRVADR  076330            5755#   6471    7011
TRVALN  076144            5492    5664#
TRVALP  076100            5491    5650#
TRVBIF  075450            5488    5547#
TRVBR   075440            5487    5543#
TRVBRC  075364            5501    5521#   5544    5550    5570    5647    5662    5683    5712
TRVDEC  075544            5494    5573#
TRVERR  075402            5485    5531#
TRVEXI  075422            5486    5537#
TRVNMA  075564            5574    5577#
TRVNOB  075374            5526#   5551    5569    5637    5661    5682
TRVNUM  075556            5490    5576#
TRVOCT  075556            5493    5575#
TRVSPA  075472            5489    5554#
TRVSTR  076232            5495    5687#
TSTMS1  061551            3180#
TSTMS2  061571            3183#   7284
TSTMS3  061611            3186#
TSTMS4  061624            3188#   7217    7403    7434    7466
TXI   = 010000 G          1299#   4405    4417
T$ARGC= 000002             966#    967#    968#    969#    970#    971#   3773#   3778   3792#   3797   3799#   3804   3813#
                          3818    3820#   3825    4639#   4643    5267#   5274   5282#   5287   5359#   5364   5596#   5600
                          5640#   5644    5798#   5802    5966#   5970    6033#   6038   6045#   6050   6052#   6056   6071#
                          6075    6080#   6084    6087#   6091    6094#   6098   6102#   6106   6109#   6113   6361#   6365
                          6370#   6374    6454#   6458    6478#   6482    6488#   6492   6494#   6498   6514#   6520   6530#
                          6534    6536#   6540    6542#   6546    6595#   6600   6608#   6614   6642#   6647   6655#   6660
                          6662#   6666    6671#   6678    6680#   6684    6689#   6695   6699#   6704   6746#   6750   6771#
                          6775    6795#   6799    6813#   6818    6821#   6826   6829#   6834   6837#   6842   6845#   6850
                          6853#   6858    6861#   6866    6904#   6908    6910#   6917   6940#   6944   7024#   7028   7058#
                          7062    7080#   7084    7106#   7110    7117#   7121   7137#   7142   7186#   7190   7213#   7221
                          7232#   7236    7267#   7272    7282#   7288    7295#   7300   7322#   7326   7348#   7353   7370#
                          7374    7376#   7380    7399#   7407    7418#   7422   7430#   7438   7449#   7453   7462#   7470
                          7481#   7485    7519#   7525    7579#   7583    7585#   7592   7606#   7610   7612#   7619   7636#
                          7641    7643#   7648    7651#   7656    7659#   7664   7666#   7671   7673#   7678   7681#   7686
                          7689#   7694    7696#   7701    7703#   7708    7711#   7716   7719#   7724   7727#   7732   7735#
                          7740    7743#   7748    7751#   7756    7759#   7764   7766#   7771   7773#   7778   7780#   7785
                          7808#   7812    7821#   7826    7848#   7853    7879#   7884   7899#   7904   7925#   7930   7964#
                          7969    7978#   7982    7985#   7990    7995#   7999   8010#   8015   8055#   8059   8191#   8196
                          8269#   8274    8281#   8286    8296#   8301
T$CODE= 002032            6347#   8351#   8356#   8361#
T$ERRN= 000036             628#   4128#   4134#   4141#   4148#   4192#   4218#   4326#   4335#   4344#   4569#   4575#   4581#
                          4588#   4652#   4936#   4942#   5019#   5025#   5074#   5080#   5133#   6120#   6204#   6553#   6623#
                          6871#   6878#   7207#   7227#   7307#   7316#   7392#   7413#   7444#   7476#   7790#   7866#
T$EXCP= 000000            6347#   6352    8351#   8355    8356#   8360    8361#   8366
T$FLAG= 000040            5832#   5834    6142#   6219#   6255#   6257    6292#   6294   6379#   8304#
T$GMAN= 000000             628#   6344#   6347    6353#
T$HILI= 000007            6347#   6351    8351#   8354    8356#   8359    8361#   8365
```

```
T$LAST= 000001          628#  8437#
T$LOLI= 000000         6347#  6350  8351#  8353  8356#  8358  8361#  8364
T$LSYM= 010000          628#  1113  1131  3783  3808  3828  4012  5849  6159  6179  6236  6272  6309
                       8321  8370  8420
T$LTNO= 000001         8440#
T$NEST= 177777          628#  1102#  1112#  1124#  1130#  3770#  3782#  3786#  3807#  3811#  3827#  3986#  4011#
                       5818#  5848#  5858#  5865#  5885#  6158#  6170#  6178#  6188#  6235#  6245#  6271#  6282#
                       6308#  6335#  8320#  8339#  8368#  8405#  8418#
T$NSO = 000005         1102#  1112  1124#  1130  3770#  3782  3786#  3807  3811#  3827  3986#  4011  5818#
                       5848  5858#  5865  5885#  6158  6170#  6178  6188#  6235  6245#  6271  6282#  6308
                       6335#  8320  8339#  8368  8405#  8418
T$PTNU= 000000          628#
T$SAVL= 177777          628#
T$SEGL= 177777          628#
T$SUBN= 000000          628#  6334#
T$TAGL= 177777          628#
T$TAGN= 010020          628#  1102#  1124#  3770#  3786#  3811#  3986#  5818#  5858#  5885#  6170#  6188#  6245#
                       6282#  6335#  8339#  8405#
T$TEMP= 000005         1090#  1091#  1112#  1130#  3782#  3807#  3827#  4011#  5832#  5833  5848#  5865#  6142#
                       6143  6158#  6178#  6219#  6220  6235#  6255#  6256  6271#  6292#  6293  6308#  6347#
                       6379#  6380  8304#  8305  8320#  8351#  8356#  8361#  8368#  8418#
T$TEST= 000001          628#  6334#  8440
T$TSTM= 177777          628#  3777  3780  3783  3796  3803  3808  3817  3824  3828  4127  4133  4140
                       4147  4191  4217  4325  4334  4343  4568  4574  4580  4587  4642  4651  4935
                       4941  5018  5024  5073  5079  5132  5273  5286  5363  5599  5643  5801  5849
                       5915  5920  5926  5932  5939  5946  5955  5969  5974  5987  6020  6025  6037
                       6049  6055  6074  6083  6090  6097  6105  6112  6119  6132  6137  6140  6142
                       6159  6179  6203  6217  6219  6236  6272  6309  6344  6364  6373  6379  6457
                       6481  6491  6497  6519  6533  6539  6545  6552  6564  6599  6613  6622  6646
                       6659  6665  6677  6683  6694  6703  6749  6774  6780  6798  6817  6825  6833
                       6841  6849  6857  6865  6870  6877  6907  6916  6943  7027  7061  7083  7109
                       7120  7141  7189  7206  7220  7226  7235  7271  7287  7299  7306  7315  7325
                       7352  7373  7379  7391  7406  7412  7421  7437  7443  7452  7469  7475  7484
                       7500  7524  7582  7591  7609  7618  7640  7647  7655  7663  7670  7677  7685
                       7693  7700  7707  7715  7723  7731  7739  7747  7755  7763  7770  7777  7784
                       7789  7811  7825  7852  7865  7883  7903  7929  7960  7968  7971  7981  7989
                       7998  8014  8050  8058  8195  8273  8285  8300  8304  8321
T$TSTS= 000001          628#  6335#
T$$AU = 010014         6282#  6292  6308
T$$AUT= 010011         6170#  6178
T$$CLE= 010012         6188#  6219  6235
T$$DU = 010013         6245#  6255  6271
T$$HAR= 010016         8339#  8369
T$$HW = 010000         1102#  1112
T$$INI= 010010         5885#  6142  6158
T$$MSG= 010004         3770#  3782  3786#  3807  3811#  3827
T$$PRO= 010007         5858#
T$$RPT= 010006         5818#  5832  5848
T$$SOF= 010017         8405#  8419
T$$SRV= 010005         3986#  4011
T$$SW = 010001         1124#  1130
T$$TES= 010015         6335#  6379  8304  8320
T1      100274 G       1090  6334#
UAM   = 000200 G       1202#
UBTO  = 040000 G       1454#
UCB10   050110 G       1987  1993#  4279
```

```
UCB11   050134 G        2008    2014#
UCB12   050160 G        2033    2045#   4291    7643    7650    7658    7666    7673    7680    7688    7696    7703    7710
                        7718    7726    7734    7742    7750    7758    7766    7773    7780
UCB13   050160 G        2044#   2085
UCB20   050340 G        2159    2163#   6057
UCB21   050340 G        2164#   2175
UCB22   050400          2185    2198#   6803    6804
UCB23   050400          2194    2199#
UCB6    050010 G        1960    1964#
UCB7    050050 G        1972    1976#
UDBB    050466 G        2227#
UNACSR  047652 G        1879#   5979*   5989
UNAINI  070262 G        4181#   6027
UNAISR  071144 G        4368#   5984
UNAPRI  047656 G        1881#   5981*   5983
UNAVEC  047654 G        1880#   5980*   5985
UNBOND  054252          2689#   7024
UNIHLT= 000005 G        1344#
UNIT    047664 G        1886#
UNSMSG  054561          2723#   7880    7900    8011
WAIT    070066 G        4083#   4457
WDMODE= 000015 G        1380#   6760    6885
WDMULA= 000007 G        1371#   6199    6549    6619
WDPHYA= 000005 G        1368#
WDRNGS= 000011 G        1374#   4322
WDSYS = 000023 G        1390#
WRITES  074544          5322#   6883    7240    7330    7490
XFER    050560 G        2266#   4907*   4908*   4983*   4984*   5042*   5043*   7506
XFLAG   050520 G        2250#   4407*   4530*   4536
XMIT    071434 G        4523#   6765    7203    7303    7388    7425    7456
XPKLEN= 002756 G        1493#   1845    1846    1847    1848    1849    1850    4909    4985    5044
XPWR  = 100000 G        1329#
XRGCUR  003756 G        1750#   4235*   4241    4244    4524    4546    4547    4558    4560
XRGLST  003766 G        1754#   5170
XRGNXT  003762 G        1752#   4236*   4547    4559    4901    4933    4977    5016    5036    5071    5112    5130
XRGSRT  003752 G        1748#   5169
XRG001  004162 G        1739    1765    1845#
XRG002  007140 G        1740    1771    1846#
XRG003  012116 G        1741    1777    1847#
XRG004  015074 G        1742    1783    1848#
XRG005  020052 G        1743    1789    1849#
XRG006  023030 G        1744    1795    1850#
XRING   003772 G        1748    1750    1752    1754    1759#   1993    2015    4235    4236
XRNGTB  003736 G        1737#
X$    = 000167           908#   2356#   2357#   2358#   2359#   2360#   2361#   2362#   2363#   2364#   2365#   2366#   2367#
                        2368#   2369#   2370#   2371#   2372#   2373#   2374#   2375#   2376#   2377#   2378#   2379#   2380#
                        2381#   2382#   2383#   2384#   2385#   2386#   2387#   2388#   2389#   2393#   2394#   2395#   2397#
                        2398#   2402#   2403#   2404#   2405#   2406#   2407#   2408#   2409#   2410#   2414#   2415#   2416#
                        2417#   2418#   2419#   2420#   2421#   2422#   2423#   2424#   2425#   2426#   2427#   2428#   2429#
                        2430#   2431#   2435#   2436#   2437#   2438#   2439#   2443#   2444#   2445#   2446#   2447#   2448#
                        2449#   2450#   2451#   2452#   2453#   2454#   2455#   2456#   2457#   2458#   2459#   2460#   2461#
                        2462#   2463#   2464#   2465#   2466#   2467#   2468#   2469#   2470#   2474#   2475#   2476#   2477#
                        2478#   2479#   2480#   2481#   2482#   2483#   2484#   2485#   2486#   2490#   2491#   2492#   2496#
                        2497#   2498#   2499#
X$ALWA= 000000           628#
X$FALS= 000040           628#
```

```
X$OFFS= 000400            628#
X$TRUE= 000020            628#
ZEROS = 000002 G         1284#   6978
ZROALT= 000004 G         1286#   6994
$CLRC   050260 G         1909    2084#   4297
$CLRS   050320 G         1913    2142#   4315
$DMEM   050330 G         1914    2158#
$LMEM   050350 G         1915    2174#
$PATCH  113132 G         8427#
$PNOP   047746 G         1898    1923#
$RDCN   050150 G         1908    2032#   4288
$RDDE   047750 G         1900    1930#   4260
$RDMC   050000 G         1904    1959#
$RDMO   050270 G         1910    2101#   4300
$RDPH   047760 G         1902    1940#   4265
$RDRN   050100 G         1906    1986#   4275
$RDST   050310 G         1912    2126#   4310
$RDSY   050360 G         1916    2184#
$WDMC   050040 G         1905    1971#   6197*   6199*   6617*   6627*
$WDMO   050300 G         1911    2113#   4305    6757
$WDPH   047770 G         1903    1949#   4270
$WDRN   050124 G         1907    2007#   4284
$WTSY   050370 G         1917    2193#
.       = 113156          630#   1070#   1564#   1570#   1571#   1579#   1582#   1586#   1698#   1845#   1846#   1847#   1848#
                         1849#   1850#   1852#   1853#   1854#   1855#   1856#   1857#   1964#   1979#   2271#   2359#   2361#
                         2370#   2378#   2382#   2384#   2386#   2407#   2415#   2419#   2445#   2449#   2453#   2455#   2462#
                         2466#   2475#   2479#   3171#   3256#   3657#   3691#   4686#   4787#   5446#   5833    6143    6220
                         6256    6293    6380    8078#   8079#   8080#   8305    8391#   8428#
```

```
BCOMPL      1#    628#  5916  5976  7961  8052
BERROR      1#    628#
BGNAU       1#    628#  6281
BGNAUT      1#    628#  6169
BGNCLN      1#    628#  6187
BGNDU       1#    628#  6244
BGNHRD      1#    628#  8338
BGNHW       1#    628#  1101
BGNINI      1#    628#  5884
BGNMOD      1#    628#
BGNMSG      1#    628#  3769  3785  3810
BGNPRO      1#    628#  5857
BGNPTA      1#    628#
BGNRPT      1#    628#  5817
BGNSEG      1#    628#
BGNSET      1#    628#
BGNSFT      1#    628#  8404
BGNSRV      1#    628#  3985
BGNSUB      1#    628#
BGNSW       1#    628#  1123
BGNTST      1#    628#  6333
BNCOMP      1#    628#  5921  5927  5933  5948  5957
BNERRO      1#    628#
BREAK       1#    628#  6563  6779  7499
BRESET      1#    628#
CALL      744#  4096  4213  4243  4250  4321  4330  4339  4456  4495  4531  4545  4557  4558  4631
         4632  4648  4693  4700  4931  4932  5014  5015  5069  5070  5129  5333  6026  6027  6031
         6039  6062  6198  6209  6212  6474  6484  6500  6509  6548  6567  6571  6575  6581  6618
         6653  6687  6697  6742  6753  6759  6763  6764  6783  6787  6811  6882  6884  6935  7019
         7099  7156  7160  7164  7166  7180  7193  7198  7201  7202  7210  7211  7222  7239  7243
         7255  7263  7277  7280  7301  7302  7311  7329  7344  7357  7386  7387  7396  7397  7408
         7423  7424  7427  7428  7439  7454  7455  7459  7460  7471  7489  7503  7509  7531  7560
         7628  7634  7649  7657  7679  7687  7709  7717  7725  7733  7741  7749  7757  7804  7813
         7816  7817  7861  7972  7991  8004  8153
CKLOOP      1#    628#
CLI       909#  2355  2356  2357  2358  2359  2360  2361  2362  2363  2364  2365  2366  2367  2368
         2369  2370  2371  2372  2373  2374  2375  2376  2377  2378  2379  2380  2381  2382  2383
         2384  2385  2386  2387  2388  2392  2393  2394  2396  2397  2401  2402  2403  2404  2405
         2406  2407  2408  2409  2413  2414  2415  2416  2417  2418  2419  2420  2421  2422  2423
         2424  2425  2426  2427  2428  2429  2430  2434  2435  2436  2437  2438  2442  2443  2444
         2445  2446  2447  2448  2449  2450  2451  2452  2453  2454  2455  2456  2457  2458  2459
         2460  2461  2462  2463  2464  2465  2466  2467  2468  2469  2473  2474  2475  2476  2477
         2478  2479  2480  2481  2482  2483  2484  2485  2489  2490  2491  2495  2496  2497  2498
CLOCK       1#    628#  5944  5953
CLOSE       1#    628#  7970
CLRVEC      1#    628#
COMMEN      1#    628#
DELAY       1#    628#
DESCRI      1#    628#  1063
DEVTYP      1#    628#  1055
DISPAT      1#    628#  1087
DISPLA      1#    628#
DOCLN       1#    628#  3779  6139
DODU        1#    628#
DORPT       1#    628#
ENDAU       1#    628#  6307
```

```
ENDAUT      1#    628#  6177
ENDCLN      1#    628#  6234
ENDCOM      1#    628#
ENDDU       1#    628#  6270
ENDHRD      1#    628#  8367
ENDHW       1#    628#  1111
ENDINI      1#    628#  6157
ENDMOD      1#    628#
ENDMSG      1#    628#  3781  3806  3826
ENDPRO      1#    628#  5964
ENDPTA      1#    628#
ENDRPT      1#    628#  5847
ENDSEG      1#    628#
ENDSET      1#    628#
ENDSFT      1#    628#  8417
ENDSRV      1#    628#  4010
ENDSUB      1#    628#
ENDSW       1#    628#  1129
ENDTST      1#    628#  8319
EQUALS      1#    628#  1142
ERRDF       1#    628#  4126  4132  4139  4146  4190  4216  4324  4333  4342  4567  4573  4579  4650
         4934   4940  5017  5023  5072  5078  5131  6118  6869  6876
ERRHRD      1#    628#  4586  6202  6551  6621  7205  7225  7305  7314  7390  7411  7442  7474  7788
         7864
ERROR       1#    628#
ERRSF       1#    628#
ERRSOF      1#    628#
ERRTBL      1#    628#  2511
ESCAPE      1#    628#
EXIT        1#    628#  5831  6141  6218  6254  6291  6378  8303
FEQUAL      1#    628#
GETBYT      1#    628#  8049
GETPRI      1#    628#
GETWOR      1#    628#
GMANIA      1#    628#
GMANID      1#    628#  6343
GMANIL      1#    628#
GPHARD      1#    628#  5972
GPRMA       1#    628#  8350  8355
GPRMD       1#    628#  6344#  6347  8360
GPRML       1#    628#
HEADER      1#    628#   964
INLOOP      1#    628#
IOSETU      1#    628#
IOSTAR      1#    628#
I$STAC    648#  5937
KT11        1#    628#
LASTAD      1#    628#  8435
MANUAL      1#    628#
MEMORY      1#    628#  5938
M$BYTE      1#    628#   965#   971   972   973
M$CHEC      1#    628#  5832#  6142#  6219#  6255#  6292#  6379#  8304#
M$CNTO      1#    628#  6347#  8351#  8356#  8361#
M$COUN      1#    628#  3773#  3792#  3799#  3813#  3820#  4639#  5267#  5282#  5359#  5596#  5640#  5798#  5966#
         6033#  6045#  6052#  6071#  6080#  6087#  6094#  6102#  6109#  6361#  6370#  6454#  6478#  6488#  6494#
         6514#  6530#  6536#  6542#  6595#  6608#  6642#  6655#  6662#  6671#  6680#  6689#  6699#  6746#  6771#
```

	6795#	6813#	6821#	6829#	6837#	6845#	6853#	6861#	6904#	6910#	6940#	7024#	7058#	7080#	7106#
	7117#	7137#	7186#	7213#	7232#	7267#	7282#	7295#	7322#	7348#	7370#	7376#	7399#	7418#	7430#
	7449#	7462#	7481#	7519#	7579#	7585#	7606#	7612#	7636#	7643#	7651#	7659#	7666#	7673#	7681#
	7689#	7696#	7703#	7711#	7719#	7727#	7735#	7743#	7751#	7759#	7766#	7773#	7780#	7808#	7821#
	7848#	7879#	7899#	7925#	7964#	7978#	7985#	7995#	8010#	8055#	8191#	8269#	8281#	8296#	
M$DATA	1#	628#	965#	974	976	978	980	982	984	986	988	990	992	994	996
	993	1000	1002	1004#	1006	1008	1011	1014	1016	1018	1020	1022	1024	1026	1028
	1030	1032	1034	1036	1038	1040	1042	1044	1046	1048	1056#	1064#			
M$DECR	1#	628#	1112#	1130#	3782#	3807#	3827#	4011#	5848#	5865#	6158#	6178#	6235#	6271#	6308#
	8320#	8368#	8418#												
M$DEFA	1#	628#	6347#	8351#	8356#	8361#									
M$ENDE	1#	628#	1112#	1130#	3782#	3807#	3827#	4011#	5848#	6158#	6178#	6235#	6271#	6308#	8320#
	8368#	8418#													
M$ERRI	1#	628#	4127#	4133#	4140#	4147#	4191#	4217#	4325#	4334#	4343#	4568#	4574#	4580#	4587#
	4651#	4935#	4941#	5018#	5024#	5073#	5079#	5132#	6119#	6203#	6552#	6622#	6870#	6877#	7206#
	7226#	7306#	7315#	7391#	7412#	7443#	7475#	7789#	7865#						
M$ESCA	1#	628#													
M$ESCS	1#	628#													
M$EXCP	1#	628#	6347#	8351#	8356#	8361#									
M$EXIT	1#	628#	5832#	6142#	6143	6219#	6220	6255#	6292#	6379#	6380	8304#	8305		
M$EXSE	1#	628#	5832#	6142#	6219#	6255#	6292#	6379#	8304#						
M$EXTJ	1#	628#	5832#	5833	6142#	6219#	6255#	6256	6292#	6293	6379#	8304#			
M$GEN	1#	628#	965#	974#	976#	978#	980#	982#	984#	986#	988#	990#	992#	994#	996#
	998#	1000#	1002#	1004#	1006#	1008#	1011#	1014#	1016#	1018#	1020#	1022#	1024#	1026#	1028#
	1030#	1032#	1034#	1036#	1038#	1040#	1042#	1044#	1046#	1048#	1056#	1064#	1089#	1103#	1104#
	1112#	1125#	1126#	1130#	2512#	3770#	3782#	3786#	3807#	3811#	3827#	3986#	4011#	5818#	5848#
	5858#	5885#	6158#	6170#	6178#	6188#	6235#	6245#	6271#	6282#	6308#	6334#	6352#	8320#	8340#
	8369#	8406#	8419#	8439#											
M$GENB	1#	628#	6344#	6345											
M$GETS	1#	628#	1112#	1130#	3782#	3807#	3827#	4011#	5848#	5865#	6158#	6178#	6235#	6271#	6308#
	8320#	8368#	8418#												
M$GETT	1#	628#	5832#	6142#	6219#	6255#	6292#	6379#	8304#						
M$GNGB	1#	628#	965#	974#	976#	978#	980#	982#	984#	986#	988#	990#	992#	994#	996#
	998#	1000#	1002#	1004#	1006#	1008#	1011#	1014#	1016#	1018#	1020#	1022#	1024#	1026#	1028#
	1030#	1032#	1034#	1036#	1038#	1040#	1042#	1044#	1046#	1048#	1056#	1064#	1088#	1089	1102#
	1103	1104	1124#	1125	1126	2512#	3770#	3786#	3811#	3986#	5818#	5858#	5885#	6170#	6188#
	6245#	6282#	8339#	8340	8405#	8406	8436#	8439							
M$GNIN	1#	628#	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#	1001	1002#	1003	1004#	1005	1006#	1007
	1008#	1009	1010	1011#	1012	1013#	1014#	1015	1016#	1017	1018#	1019	1020#	1021	1022#
	1023	1024#	1025	1026#	1027	1028#	1029	1030#	1031	1032#	1033	1034#	1035	1036#	1037
	1038#	1039	1040#	1041	1042#	1043	1044#	1045	1046#	1047	1048#	1049	1056#	1057	1058
	1064#	1065	1070	1088#	1090#	1102#	1124#	3773#	3774#	3775#	3776	3777#	3778	3780#	3783#
	3792#	3793#	3794#	3795	3796#	3797	3799#	3800#	3801#	3802	3803#	3804	3808#	3813#	3814#
	3815#	3816	3817#	3818	3820#	3821#	3822#	3823	3824#	3825	3828#	4011#	4012	4127#	4128#
	4129#	4130#	4133#	4134#	4135#	4136#	4140#	4141#	4142#	4143#	4147#	4148#	4149#	4150#	4191#
	4192#	4193#	4194#	4217#	4218#	4219#	4220#	4325#	4326#	4327#	4328#	4334#	4335#	4336#	4337#
	4343#	4344#	4345#	4346#	4568#	4569#	4570#	4571#	4574#	4575#	4576#	4577#	4580#	4581#	4582#
	4583#	4587#	4588#	4589#	4590#	4639#	4640#	46·1	4642#	4643	4651#	4652#	4653#	4654#	4935#
	4936#	4937#	4938#	4941#	4942#	4943#	4944#	5018#	5019#	5020#	5021#	5024#	5025#	5026#	5027#
	5073#	5074#	5075#	5076#	5079#	5080#	5081#	5082#	5132#	5133#	5134#	5135#	5267#	5268#	5269#
	5270#	5271#	5272	5273#	5274	5282#	5283#	5284#	5285	5286#	5287	5359#	5360#	5361#	5362
	5363#	5364	5596#	5597#	5598	5599#	5600	5640#	5641#	5642	5643#	5644	5798#	5799#	5800
	5801#	5802	5832#	5833#	5849#	5914#	5915#	5917#	5919#	5920#	5922#	5925#	5926#	5928#	5931#
	5932#	5934#	5939#	5940#	5945#	5946#	5947#	5949#	5954#	5955#	5956#	5958#	5966#	5967#	5968

	5969#	5970	5973#	5974#	5975#	5977#	5983#	5984#	5985#	5986#	5987#	5988	6016#	6017#	6018#
	6019#	6020#	6021	6024#	6025#	6033#	6034#	6035#	6036	6037#	6038	6045#	6046#	6047#	6048
	6049#	6050	6052#	6053#	6054	6055#	6056	6071#	6072#	6073	6074#	6075	6080#	6081#	6082
	6083#	6084	6087#	6088#	6089	6090#	6091	6094#	6095#	6096	6097#	6098	6102#	6103#	6104
	6105#	6106	6109#	6110#	6111	6112#	6113	6119#	6120#	6121#	6122#	6128#	6129#	6130#	6131#
	6132#	6133	6136#	6137#	6140#	6142#	6143#	6159#	6179#	6203#	6204#	6205#	6206#	6216#	6217#
	6219#	6220#	6236#	6255#	6256#	6272#	6292#	6293#	6309#	6344#	6345#	6346#	6347#	6348	6349
	6350	6351	6361#	6362#	6363	6364#	6365	6370#	6371#	6372	6373#	6374	6379#	6380#	6454#
	6455#	6456	6457#	6458	6478#	6479#	6480	6481#	6482	6488#	6489#	6490	6491#	6492	6494#
	6495#	6496	6497#	6498	6514#	6515#	6516#	6517#	6518	6519#	6520	6530#	6531#	6532	6533#
	6534	6536#	6537#	6538	6539#	6540	6542#	6543#	6544	6545#	6546	6552#	6553#	6554#	6555#
	6564#	6595#	6596#	6597#	6598	6599#	6600	6608#	6609#	6610#	6611#	6612	6613#	6614	6622#
	6623#	6624#	6625#	6642#	6643#	6644#	6645	6646#	6647	6655#	6656#	6657#	6658	6659#	6660
	6662#	6663#	6664	6665#	6666	6671#	6672#	6673#	6674#	6675#	6676	6677#	6678	6680#	6681#
	6682	6683#	6684	6689#	6690#	6691#	6692#	6693	6694#	6695	6699#	6700#	6701#	6702	6703#
	6704	6746#	6747#	6748	6749#	6750	6771#	6772#	6773	6774#	6775	6780#	6795#	6796#	6797
	6798#	6799	6813#	6814#	6815#	6816	6817#	6818	6821#	6822#	6823#	6824	6825#	6826	6829#
	6830#	6831#	6832	6833#	6834	6837#	6838#	6839#	6840	6841#	6842	6845#	6846#	6847#	6848
	6849#	6850	6853#	6854#	6855#	6856	6857#	6858	6861#	6862#	6863#	6864	6865#	6866	6870#
	6871#	6872#	6873#	6877#	6878#	6879#	6880#	6904#	6905#	6906	6907#	6908	6910#	6911#	6912#
	6913#	6914#	6915	6916#	6917	6940#	6941#	6942	6943#	6944	7024#	7025#	7026	7027#	7028
	7058#	7059#	7060	7061#	7062	7080#	7081#	7082	7083#	7084	7106#	7107#	7108	7109#	7110
	7117#	7118#	7119	7120#	7121	7137#	7138#	7139#	7140	7141#	7142	7186#	7187#	7188	7189#
	7190	7206#	7207#	7208#	7209#	7213#	7214#	7215#	7216#	7217#	7218#	7219	7220#	7221	7226#
	7227#	7228#	7229#	7232#	7233#	7234	7235#	7236	7267#	7268#	7269#	7270	7271#	7272	7282#
	7283#	7284#	7285#	7286	7287#	7288	7295#	7296#	7297#	7298	7299#	7300	7306#	7307#	7308#
	7309#	7315#	7316#	7317#	7318#	7322#	7323#	7324	7325#	7326	7348#	7349#	7350#	7351	7352#
	7353	7370#	7371#	7372	7373#	7374	7376#	7377#	7378	7379#	7380	7391#	7392#	7393#	7394#
	7399#	7400#	7401#	7402#	7403#	7404#	7405	7406#	7407	7412#	7413#	7414#	7415#	7418#	7419#
	7420	7421#	7422	7430#	7431#	7432#	7433#	7434#	7435#	7436	7437#	7438	7443#	7444#	7445#
	7446#	7449#	7450#	7451	7452#	7453	7462#	7463#	7464#	7465#	7466#	7467#	7468	7469#	7470
	7475#	7476#	7477#	7478#	7481#	7482#	7483	7484#	7485	7500#	7519#	7520#	7521#	7522#	7523
	7524#	7525	7579#	7580#	7581	7582#	7583	7585#	7586#	7587#	7588#	7589#	7590	7591#	7592
	7606#	7607#	7608	7609#	7610	7612#	7613#	7614#	7615#	7616#	7617	7618#	7619	7636#	7637#
	7638#	7639	7640#	7641	7643#	7644#	7645#	7646	7647#	7648	7651#	7652#	7653#	7654	7655#
	7656	7659#	7660#	7661#	7662	7663#	7664	7666#	7667#	7668#	7669	7670#	7671	7673#	7674#
	7675#	7676	7677#	7678	7681#	7682#	7683#	7684	7685#	7686	7689#	7690#	7691#	7692	7693#
	7694	7696#	7697#	7698#	7699	7700#	7701	7703#	7704#	7705#	7706	7707#	7708	7711#	7712#
	7713#	7714	7715#	7716	7719#	7720#	7721#	7722	7723#	7724	7727#	7728#	7729#	7730	7731#
	7732	7735#	7736#	7737#	7738	7739#	7740	7743#	7744#	7745#	7746	7747#	7748	7751#	7752#
	7753#	7754	7755#	7756	7759#	7760#	7761#	7762	7763#	7764	7766#	7767#	7768#	7769	7770#
	7771	7773#	7774#	7775#	7776	7777#	7778	7780#	7781#	7782#	7783	7784#	7785	7789#	7790#
	7791#	7792#	7808#	7809#	7810	7811#	7812	7821#	7822#	7823#	7824	7825#	7826	7848#	7849#
	7850#	7851	7852#	7853	7865#	7866#	7867#	7868#	7879#	7880#	7881#	7882	7883#	7884	7899#
	7900#	7901#	7902	7903#	7904	7925#	7926#	7927#	7928	7929#	7930	7959#	7960#	7962#	7964#
	7965#	7966#	7967	7968#	7969	7971#	7978#	7979#	7980	7981#	7982	7985#	7986#	7987#	7988
	7989#	7990	7995#	7996#	7997	7998#	7999	8010#	8011#	8012#	8013	8014#	8015	8050#	8051#
	8053#	8055#	8056#	8057	8058#	8059	8191#	8192#	8193#	8194	8195#	8196	8269#	8270#	8271#
	8272	8273#	8274	8281#	8282#	8283#	8284	8285#	8286	8296#	8297#	8298#	8299	8300#	8301
	8304#	8305#	8321#	8339#	8351#	8352	8353	8354	8356#	8357	8358	8359	8361#	8362	8363
	8364	8365	8368#	8405#	8418#	8436#	8437#	8438#							
M$GNLS	1#	628#	6344#	6352											
M$GNSU	1#	628#													
M$GNTA	1#	628#	1112#	1130#	3782#	3807#	3827#	4011#	5848#	6158#	6178#	6235#	6271#	6308#	8320#
	8368#	8369	8418#	8419											
M$GNTE	1#	628#	6334#												

M$HAPT 1# 628# 965#

M$HNAP 1# 628# 965# 1004

M$INCR 1# 628# 1102# 1124# 3770# 3777# 3780# 3783# 3786# 3796# 3803# 3808# 3811# 3817# 3824#
3828# 3986# 4127# 4133# 4140# 4147# 4191# 4217# 4325# 4334# 4343# 4568# 4574# 4580# 4587#
4642# 4651# 4935# 4941# 5018# 5024# 5073# 5079# 5132# 5273# 5286# [illegible]3# 5599# 5643# 5801#
5818# 5849# 5858# 5885# 5915# 59[illegible]0# 5926# 5932# 5939# 5946# 5955# [illegible]969# 5974# 5987# 6020#
6025# 6037# 6049# 6055# 6074# 6083# 6090# 6097# 6105# 6112# 6119# 6132# 6137# 6140# 6142#
6159# 6170# 6179# 61[illegible]5# 6203# 6217# 6219# 6236# 6245# 6272# 6282# 6309# 6334# 6335# 6344#
6353 6364# 6373# 63[illegible]9# 6457# 6481# 6491# 6497# 6519# 6533# 6539# 6545# 6552# 6564# 659[illegible]#
6613# 6622# 6646# 6659# 6665# 6677# 6683# 6694# 6703# 6749# 6774# 6780# 6798# 6817# 682[illegible]#
6833# 6841# 6849# 6857# 6865# 6870# 6877# 6907# 6916# 6943# 7027# 7061# 7083# 7109# 7120#
7141# 7189# 72[illegible]6# 7220# 7226# 7235# 7271# 7287# 7299# 7306# 7315# 7325# 7352# 7373# 7379#
7391# 7406# 74[illegible]2# 7421# 7437# 7443# 7452# 7469# 7475# 7484# 7500# 7524# 7582# 7591# 7609#
7618# 7640# 7647# 7655# 7663# 7670# 7677# 7685# 7693# 7700# 7707# 7715# 7723# 7731# 7739#
7747# 7755# 7763# 7770# 7777# 7784# 7789# 781[illegible]# 7825# 7852# 7865# 7883# 7903# 7929# 7960#
7968# 7971# 7981# 7989# 7998# 8014# 8050# 8058# 8195# 8273# 8285# 8300# 8304# 8321# 8339#
8405#

M$IOSE 1# 628#

M$LDRO 1# 628# 5914# 5919# 5925# 5931# 5945# 5954# 5973# 6024# 6136# 6216# 7959#

M$MASK 1# 628#

M$MCHI 1# 628#

M$MCLO 1# 628#

M$MSK1 1# 628#

M$POP 1# 628# 1112# 1130# 3782# 3807# 3827# 4011# 5848# 5865# 6158# 6178# 6235# 6271# 6308#
8320# 8368# 8418#

M$PRIN 1# 628# 3773# 3792# 3799# 3813# 3820# 4639# 5267# 5282# 5359# 5596# 5640# 5798# 5966#
6033# 6045# 6052# 6071# 6080# 6087# 6094# 6102# 6109# 6361# 6370# 6454# 6478# 6488# 6494#
6514# 6530# 6536# 6542# 6595# 6608# 6642# 6655# 6662# 6671# 6680# 6689# 6699# 6746# 6771#
6795# 6813# 6821# 6829# 6837# 6845# 6853# 6861# 6904# 6910# 6940# 7024# 7058# 7080# 7106#
7117# 7137# 7186# 7213# 7232# 7267# 7282# 7295# 7322# 7348# 7370# 7376# 7399# 7418# 7430#
7449# 7462# 748[illegible]# 7519# 7579# 7585# 7606# 7612# 7636# 7643# 7651# 7659# 7666# 7673# 7681#
7689# 7696# 7703# 7711# 7719# 7727# 7735# 7743# 7751# 7759# 7766# 7773# 7780# 7808# 7821#
7848# 7879# 78[illegible]9# 7925# 7964# 7978# 7985# 7995# 8010# 8055# 8191# 8269# 8281# 8296#

M$PUSH 1# 628# 1102# 1124# 3770# 3786# 3811# 3986# 5818# 5858# 5885# 6170# 6188# 6245# 6282#
6334# 6335 8339# 8405#

M$PUT 1# 628# 3773# 3792# 3799# 3813# 3820# 4639# 5267# 5282# 5359# 5596# 5640# 5798# 5966#
5983# 6016# 6033# 6045# 6052# 6071# 6080# 6087# 6094# 6102# 6109# 6128# 6361# 6370# 6454#
6478# 6488# 6494# 6514# 6530# 6536# 6542# 6595# 6608# 6642# 6655# 6662# 6671# 6680# 6689#
6699# 6746# 6771# 67[illegible]5# 6813# 6821# 6829# 6837# 6845# 6853# 6861# 6904# 6910# 6940# 7024#
7058# 7080# 7106# 7117# 7137# 7186# 7213# 7232# 7267# 7282# 7295# 7322# 7348# 7370# 7376#
7399# 7418# 7430# 7449# 7462# 7481# 7519# 7579# 7585# 7606# 7612# 7636# 7643# 7651# 7659#
7666# 7673# 7681# 7689# 7696# 7703# 7711# 7719# 7727# 7735# 7743# 7751# 7759# 7766# 7773#
7780# 7808# 7821# 7848# 7879# 7899# 7925# 7964# 7978# 7985# 7995# 8010# 8055# 8191# 8269#
8281# 8296#

M$PUT1 1# 628# 3773# 3774 3775 3792# 3793 3794 3799# 3800 3801 3813# 3814 3815 3820#
3821 3822 4639# 4640 5267# 5268 5269 5270 5271 5282# 5283 5284 5359# 5360 5361
5596# 5597 5640# 5641 5798# 5799 5966# 5967 5983# 5984 5985 5986 6016# 6017 6018
6019 6033# 6034 6035 6045# 6046 6047 6052# 6053 6071# 6072 6080# 6081 6087# 6088
6094# 6095 6102# 6103 6109# 6110 6128# 6129 6130 6131 6361# 6362 6370# 6371 6454#
6455 6478# 6479 6488# 6489 6494# 6495 6514# 6515 6516 6517 6530# 6531 6536# 6537
6542# 6543 6595# 6596 6597 6608# 6609 6610 6611 6642# 6643 6644 6655# 6656 6657
6662# 6663 6671# 6672 6673 6[illegible]74 6675 6680# 6681 6689# 6690 6691 6692 6699# 6700
6701 6746# 6747 6771# 67[illegible]2 6795# 6796 6813# 6814 6815 6821# 6822 6823 6829# 6830
6831 6837# 6838 6839 6845# 6846 6847 6853# 6854 6855 6861# 6862 6863 6904# 6905
6910# 6911 6912 6913 6914 6940# 6941 7024# 7025 7058# 7059 7080# 7081 7106# 7107
7117# 7118 7137# 7138 7139 7186# 7187 7213# 7214 7215 7216 7217 7218 7232# 7233

	7267#	7268	7269	7282#	7283	7284	7285	7295#	7296	7297	7322#	7323	7348#	7349	7350
	7370#	7371	7376#	7377	7399#	7400	7401	7402	7403	7404	7418#	7419	7430#	7431	7432
	7433	7434	7435	7449#	7450	7462#	7463	7464	7465	7466	7467	7481#	7482	7519#	7520
	7521	7522	7579#	7580	7585#	7586	7587	7588	7589	7606#	7607	7612#	7613	7614	7615
	7616	7636#	7637	7638	7643#	7644	7645	7651#	7652	7653	7659#	7660	7661	7666#	7667
	7668	7673#	7674	7675	7681#	7682	7683	7689#	7690	7691	7696#	7697	7698	7703#	7704
	7705	7711#	7712	7713	7719#	7720	7721	7727#	7728	7729	7735#	7736	7737	7743#	7744
	7745	7751#	7752	7753	7759#	7760	7761	7766#	7767	7768	7773#	7774	7775	7780#	7781
	7782	7808#	7809	7821#	7822	7823	7848#	7849	7850	7879#	7880	7881	7899#	7900	7901
	7925#	7926	7927	7964#	7965	7966	7978#	7979	7985#	7986	7987	7995#	7996	8010#	8011
	8012	8055#	8056	8191#	8192	8193	8269#	8270	8271	8281#	8282	8283	8296#	8297	8298
M$RADI	1#	628#	6347#	8351#	8356#	8361#									
M$RBRO	1#	628#	8050#	8051											
M$RNRO	1#	628#	5939#	5940	5945#	5947	5954#	5956	5973#	5975					
M$SETS	1#	628#	1102#	1124#	3770#	3786#	3811#	3986#	5818#	5858#	5885#	6170#	6188#	6245#	6282#
	6335#	8339#	8405#												
M$STAR	1#	628#													
M$SVC	1#	628#	3773#	3777	3780#	3782#	3783	3792#	3796	3799#	3803	3807#	3808	3813#	3817
	3820#	3824	3827#	3828	4127	4133	4140	4147	4191	4217	4325	4334	4343	4568	4574
	4580	4587	4639#	4642	4651	4935	4941	5018	5024	5073	5079	5132	5267#	5273	5282#
	5286	5359#	5363	5596#	5599	5640#	5643	5798#	5801	5832#	5848#	5849	5914#	5915	5919#
	5920	5925#	5926	5931#	5932	5939#	5945#	5946	5954#	5955	5966#	5969	5973#	5974	5983#
	5987	6016#	6020	6024#	6025	6033#	6037	6045#	6049	6052#	6055	6071#	6074	6080#	6083
	6087#	6090	6094#	6097	6102#	6105	6109#	6112	6119	6128#	6132	6136#	6137	6140#	6142#
	6158#	6159	6178#	6179	6203	6216#	6217	6219#	6235#	6236	6255#	6271#	6272	6292#	6308#
	6309	6344#	6361#	6364	6370#	6373	6379#	6454#	6457	6478#	6481	6488#	6491	6494#	6497
	6514#	6519	6530#	6533	6536#	6539	6542#	6545	6552	6564#	6595#	6599	6608#	6613	6622
	6642#	6646	6655#	6659	6662#	6665	6671#	6677	6680#	6683	6689#	6693	6699#	6703	6746#
	6749	6771#	6774	6780#	6795#	6798	6813#	6817	6821#	6825	6829#	6833	6837#	6841	6845#
	6849	6853#	6857	6861#	6865	6870	6877	6904#	6907	6910#	6916	6940#	6943	7024#	7027
	7058#	7061	7080#	7083	7106#	7109	7117#	7120	7137#	7141	7186#	7189	7206	7213#	7220
	7226	7232#	7235	7267#	7271	7282#	7287	7295#	7299	7306	7315	7322#	7325	7348#	7352
	7370#	7373	7376#	7379	7391	7399#	7406	7412	7418#	7421	7430#	7437	7443	7449#	7452
	7462#	7469	7475	7481#	7484	7500#	7519#	7524	7579#	7582	7585#	7591	7606#	7609	7612#
	7618	7636#	7640	7643#	7647	7651#	7655	7659#	7663	7666#	7670	7673#	7677	7681#	7685
	7689#	7693	7696#	7700	7703#	7707	7711#	7715	7719#	7723	7727#	7731	7735#	7739	7743#
	7747	7751#	7755	7759#	7763	7766#	7770	7773#	7777	7780#	7784	7789	7808#	7811	7821#
	7825	7848#	7852	7865	7879#	7883	7899#	7903	7925#	7929	7959#	7960	7964#	7968	7971#
	7978#	7981	7985#	7989	7995#	7998	8010#	8014	8050#	8055#	8058	8191#	8195	8269#	8273
	8281#	8285	8296#	8300	8304#	8320#	8321								
M$TLAB	1#	628#	3777#	3780#	3783#	3796#	3803#	3808#	3817#	3824#	3828#	4127#	4133#	4140#	4147#
	4191#	4217#	4325#	4334#	4343#	4568#	4574#	4580#	4587#	4642#	4651#	4935#	4941#	5018#	5024#
	5073#	5079#	5132#	5273#	5286#	5363#	5599#	5643#	5801#	5849#	5915#	5920#	5926#	5932#	5939#
	5946#	5955#	5969#	5974#	5987#	6020#	6025#	6037#	6049#	6055#	6074#	6083#	6090#	6097#	6105#
	6112#	6119#	6132#	6137#	6140#	6142#	6159#	6179#	6203#	6217#	6219#	6236#	6272#	6309#	6344#
	6364#	6373#	6379#	6457#	6481#	6491#	6497#	6519#	6533#	6539#	6545#	6552#	6564#	6599#	6613#
	6622#	6646#	6659#	6665#	6677#	6683#	6694#	6703#	6749#	6774#	6780#	6798#	6817#	6825#	6833#
	6841#	6849#	6857#	6865#	6870#	6877#	6907#	6916#	6943#	7027#	7061#	7083#	7109#	7120#	7141#
	7189#	7206#	7220#	7226#	7235#	7271#	7287#	7299#	7306#	7315#	7325#	7352#	7373#	7379#	7391#
	7406#	7412#	7421#	7437#	7443#	7452#	7469#	7475#	7484#	7500#	7524#	7582#	7591#	7609#	7618#
	7640#	7647#	7655#	7663#	7670#	7677#	7685#	7693#	7700#	7707#	7715#	7723#	7731#	7739#	7747#
	7755#	7763#	7770#	7777#	7784#	7789#	7811#	7825#	7852#	7865#	7883#	7903#	7929#	7960#	7968#
	7971#	7981#	7989#	7998#	8014#	8050#	8058#	8195#	8273#	8285#	8300#	8304#	8321#		
M$TS`L	1#	628#	3777#	3780#	3783#	3796#	3803#	3808#	3817#	3824#	3828#	4127#	4133#	4140#	4147#
	4191#	4217#	4325#	4334#	4343#	4568#	4574#	4580#	4587#	4642#	4651#	4935#	4941#	5018#	5024#
	5073#	5079#	5132#	5273#	5286#	5363#	5599#	5643#	5801#	5849#	5915#	5920#	5926#	5932#	5939#

```
            5946#  5955#  5969#  5974#  5987#  6020#  6025#  6037#  6049#  6055#  6074#  6083#  6090#  6097#  6105#
            6112#  6119#  6132#  6137#  6140#  6142#  6159#  6179#  6203#  6217#  6219#  6236#  6272#  6309#  6344#
            6364#  6373#  6579#  6457#  6481#  6491#  6497#  6519#  6533#  6539#  6545#  6552#  6564#  6599#  6613#
            6622#  6646#  6659#  6665#  6677#  6683#  6694#  6703#  6749#  6774#  6780#  6798#  6817#  6825#  6833#
            6841#  6849#  6857#  6865#  6870#  6877#  6907#  6916#  6943#  7027#  7061#  7083#  7109#  7120#  7141#
            7189#  7206#  7220#  7226#  7235#  7271#  7287#  7299#  7306#  7315#  7325#  7352#  7373#  7379#  7391#
            7406#  7412#  7421#  7437#  7443#  7452#  7469#  7475#  7484#  7500#  7524#  7582#  7591#  7609#  7618#
            7640#  7647#  7655#  7663#  7670#  7677#  7685#  7693#  7700#  7707#  7715#  7723#  7731#  7739#  7747#
            7755#  7763#  7770#  7777#  7784#  7789#  7811#  7825#  7852#  7865#  7883#  7903#  7929#  7960#  7968#
            7971#  7981#  7989#  7998#  8014#  8050#  8058#  8195#  8273#  8285#  8300#  8304#  8321#
MSWORD        1#   628#  1004#  1013   1088#  1090   4127#  4128   4129   4130   4133#  4134   4135   4136   4140#
            4141   4142   4143   4147#  4148   4149   4150   4191#  4192   4193   4194   4217#  4218   4219   4220
            4325#  4326   4327   4328   4334#  4335   4336   4337   4343#  4344   4345   4346   4568#  4569   4570
            4571   4574#  4575   4576   4577   4580#  4581   4582   4583   4587#  4588   4589   4590   4651#  4652
            4653   4654   4935#  4936   4937   4938   4941#  4942   4943   4944   5018#  5019   5020   5021   5024#
            5025   5026   5027   5073#  5074   5075   5076   5079#  5080   5081   5082   5132#  5133   5134   5135
            5832#  6119#  6120   6121   6122   6142#  6203#  6204   6205   6206   6219#  6255#  6292#  6344#  6346
            6347#  6379#  6552#  6553   6554   6555   6622#  6623   6624   6625   6870#  6871   6872   6873   6877#
            6878   6879   6880   7206#  7207   7208   7209   7226#  7227   7228   7229   7306#  7307   7308   7309
            7315#  7316   7317   7318   7391#  7392   7393   7394   7412#  7413   7414   7415   7443#  7444   7445
            7446   7475#  7476   7477   7478   7789#  7790   7791   7792   7865#  7866   7867   7868   8304#  8351#
            8356#  8361#  8437   8438
MSXFER        1#   628#
NEXT        950#  1765   1771   1777   1783   1789   1795   1808   1814   1820   1826   1832   1838
NODCL       914#  2356   2357   2358   2359   2360   2361   2362   2363   2364   2365   2366   2367   2368   2369
            2370   2371   2372   2373   2374   2375   2376   2377   2378   2379   2380   2381   2382   2383   2384
            2385   2386   2387   2388   2389   2393   2394   2395   2397   2398   2402   2403   2404   2405   2406
            2407   2408   2409   2410   2414   2415   2416   2417   2418   2419   2420   2421   2422   2423   2424
            2425   2426   2427   2428   2429   2430   2431   2435   2436   2437   2438   2439   2443   2444   2445
            2446   2447   2448   2449   2450   2451   2452   2453   2454   2455   2456   2457   2458   2459   2460
            2461   2462   2463   2464   2465   2466   2467   2468   2469   2470   2474   2475   2476   2477   2478
            2479   2480   2481   2482   2483   2484   2485   2486   2490   2491   2492   2496   2497   2498   2499
OPEN          1#   628#  7958
POINTE        1#   628#   961
POP         702#  5635   5646   5706   5711
PRINTB        1#   628#  6594   7212   7231   7281   7294   7321   7398   7417   7429   7448   7461   7480
PRINTF        1#   628#  4638   5358   5595   5639   5797   5965   6360   6369   6453   6477   6487   6493   6641
            6654   6661   6670   6679   6698   6698   6745   6770   6794   6812   6820   6828   6836   6844   6852
            6860   6903   6909   6939   7023   7057   7079   7105   7116   7136   7266   7347   7369   7375   7578
            7584   7605   7611   7635   7642   7650   7658   7665   7672   7680   7688   7695   7702   7710   7718
            7726   7734   7742   7750   7758   7765   7772   7779   7807   7820   7847   7878   7898   7924   7963
            7977   7984   7994   8009   8054   8190   8268   8280   8295
PRINTS        1#   628#  6032   6044   6051   6070   6079   6086   6093   6101   6108   6513   6529   6535   6541
            6607   7185
PRINTX        1#   628#  3772   3791   3798   3812   3819   5266   5281   7518
PUSH        662#  5577   5687
P$POP       875#  4214   4322   4331   4340   4453   4484   4532   4689   4694   4726   4789   4846   4900   4975
            5034   5171   5208   5254   5322   5325   5327   5334   5397   6028   6040   6063   6200   6210   6461
            6475   6485   6510   6549   6568   6576   6582   6619   6628   6707   6743   6754   6760   6765   6784
            6885   6887   6936   7100   7122   7144   7181   7203   7223   7256   7303   7312   7388   7409   7425
            7440   7456   7472   7504   7532   7561   7568   7629   7845   7862   7909   7931   7992   8005   8007
            8124   8154   8247
P$PUSH      830#  6451   6547   6637   6752   7096   7130   7558   7835   7875   7892   7918   7956   8156   8166
            8197   8200
READBU        1#   628#
READEF        1#   628#  5913   5918   5924   5930
```

```
RETURN    788#   4098   4152   4354   4457   4496   4592   4655   4702   4761   4820   4866   4945   5028   5083
         5136   5177   5223   5288   5371   5423   7246   7259   7332   7492   7547   7569   8016   8076   8134
         8167   8201   8231   8257   8302
RFLAGS      1#    628#
RNGFRM    934#   1764   1770   1776   1782   1788   1794   1807   1813   1819   1825   1831   1837
SETPRI      1#    628#   6023   6135   6215
SETVEC      1#    628#   5982   6015   6127
SLASH       1#    628#
STARS       1#    628#
SVC         1#    628#
XFER        1#    628#   5832#  6142#  6219#  6255#  6292#  6379#  8304#
XFERF       1#    628#
XFERT       1#    628#


. ABS.  113156     000


 ERRORS DETECTED:  0

 CZUACA.BIN,CZUACA.SEQ/CRF/SOL/NL:TOC=SVC34R.MLB,CZUACA.P11
 RUN-TIME: 35 42 4 SECONDS
 RUN-TIME RATIO: 132/82=1.5
 CORE USED:  21K  (41 PAGES)
```