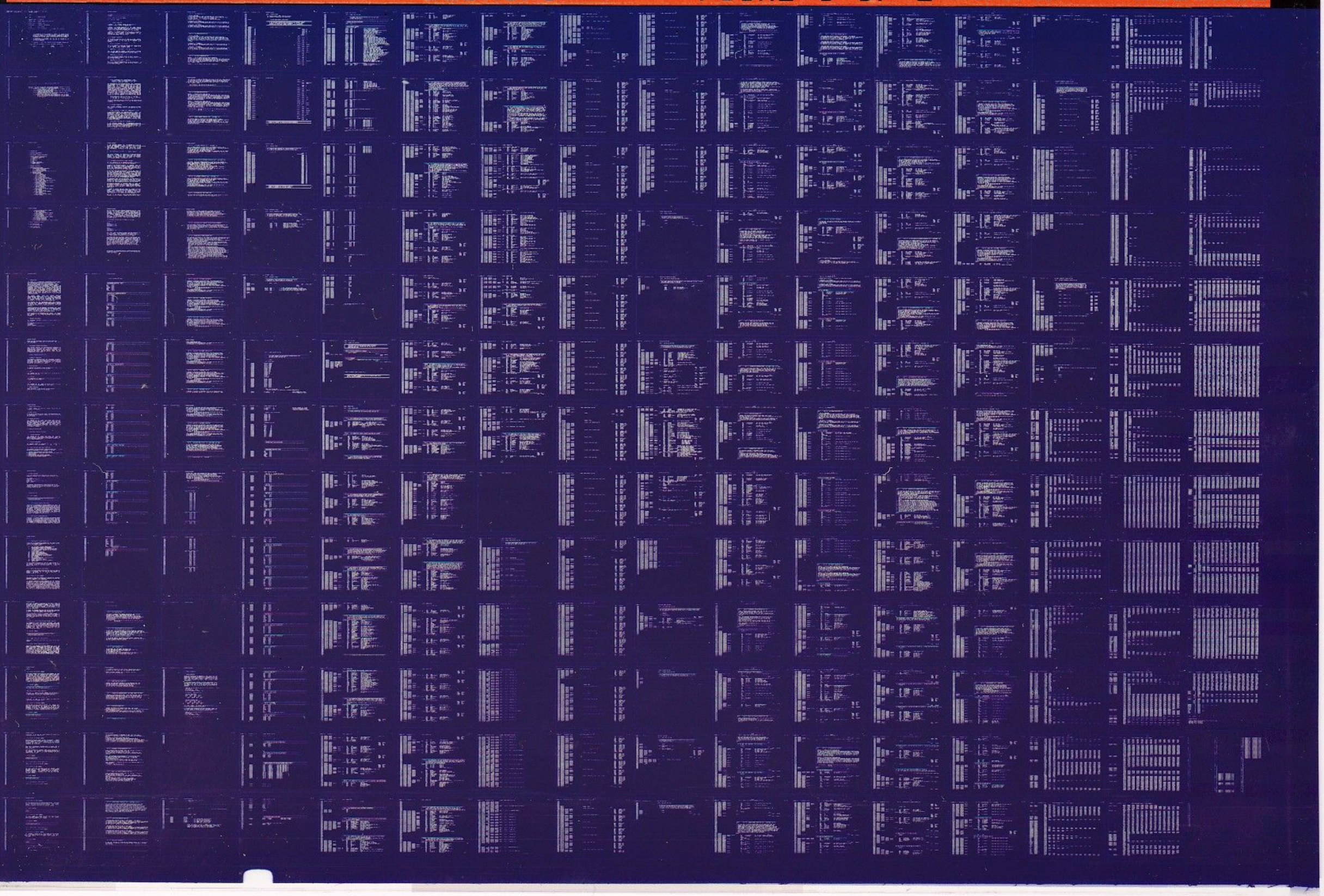
DMP-11,DMR-11, M8203 STATIC DIAG#2 AH-E235F-MC APR 1982
M8203 CZDMSF0 FICHE 1 OF 2



DMP-11,DMR-11, M8203 STATIC DIAG#2 AH-E235F-MC
M8203 CZDMSFO FICHE 2 OF 2

```
.REM @

                    IDENTIFICATION
                    --------------

PRODUCT CODE:   AC-E234F-MC

PRODUCT NAME:   CZDMSF0 M8203 STATIC DIAG #2

PRODUCT DATE:   FEBRUARY 1982

MAINTAINER:     DIAGNOSTIC ENGINEERING

AUTHOR:         DAVID HOFFMAN

MODIFIED BY:    BERT KLEINSCHMIDT       SEPTEMBER 1981  VERSION F
```

38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

***** MODIFICATION HISTORY *****

FOR THE 11-NOV-1981 RELEASE BERT KLEINSCHMIDT ENHANCED VERSION E TO CREATE
VERSION F. VERSION F CORRECTS THE FOLLOWING PROBLEMS:
VERSION E FAILS TESTS 2, 3, AND 6 WHEN USED TO TEST M8203 LINE UNITS
WITH NEW VENDER'S SILO CHIPS.
VERSION E DOES NOT TEST MODEM CONTROL SIGNALS (TEST 23) WHEN EXTERNAL
LOOPBACK CONNECTORS ARE USED.
VERSION E MALFUNCTIONS WHEN USING THE AUTOMATIC MODEM LOOP BACK
FEATURES IN TESTS 19, 20, AND 24-30.
THE HARDWARE P-TABLE QUESTIONS, THE DOCUMENTATION, AND THE M8203
SWITCH PACK PRINTOUT MESSAGES INCORRECTLY LIST THE ORDER
IN WHICH THE SWITCH BITS ARE INTERPRETTED IN M8203 REGISTERS
11, 15, AND 16.

CONTENTS

1.0 INTRODUCTION

THE M8203 IS A SINGLE-LINE SYNCHRONOUS LINE UNIT MODULE WHICH SUPPORTS BOTH CHARACTER-ORIENTED (DDCMP, BSC, ETC.) AND BIT-ORIENTED (SDLC, HDLC, ETC.) PROTOCOLS. THE PURPOSE OF THIS PROGRAM IS TO PERFORM DIAGNOSTIC TESTING OF ALL M8203 LOGIC IN A RELATIVELY STATIC MANNER. THE FOLLOWING FUNCTIONS WILL BE PERFORMED: LINE UNIT REGISTER ADDRESSING, USYRT ADDRESSING, STATIC BIT INTERACTION AND READ/WRITE LOGIC TESTS, BASIC TRANSMITTER AND RECEIVER SEQUENCING AND DATA BUFFERING AND STATIC OPERATIONS IN CHARACTER AND BIT-STUFFING MODES. IN ADDITION DATA MESSAGES WILL BE SENT AT SPEEDS OF 2400 BAUD TO 1 MEGABAUD, WITH LOOPBACK IN THE USYRT, ON THE LINE UNIT AT TTL LEVEL, OR THROUGH AN EXTERNAL TEST CONNECTOR WITH A SPECIFIC MODEM INTERFACE SELECTED.

THE STATIC LOGIC TESTS WILL PROVIDE EXTENSIVE TROUBLESHOOTING CAPABILITIES, SUCH AS TIGHT SCOPE LOOPS, SWITCH OPTIONS, AND ABILITY TO "LOCK" ONTO INTERMITTENT ERRORS. IN ADDITION TESTS WILL BE DESIGNED AND STRUCTURED TO ACHIEVE MAXIMUM FAULT RESOLUTION AND FACILITATE REPLACEMENT OF THE SMALLEST FIELD REPLACEABLE UNIT.

THIS PROGRAM WILL BE IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR AND A STRUCTURED PROGRAMMING APPROACH. BECAUSE THE DESIGN WILL CONFORM TO THE SUPERVISOR (STANDALONE VERSION) THE PROGRAM WILL BE COMPATIBLE WITH ACT, APT, XXDP+, AND SLIDE.

THROUGH DIALOGUE WITH THE OPERATOR, THE PROGRAM WILL ALLOW MODIFICATION OF DEVICE PARAMETERS, SUCH AS UNIBUS ADDRESS, VECTOR ADDRESSES AND DEVICE PRIORITY. IN ADDITION, THE OPERATOR CAN SPECIFY PARTICULAR TESTS TO BE RUN AND A VARIETY OF LOOPING, RUNNING, AND REPORTING MODES.

DEVICE ERRORS WILL BE REPORTED AS THEY OCCUR. THE REPORT WILL INCLUDE A TEST NUMBER AND DESCRIPTION OF THE ERROR, GOOD AND BAD TEST DATA, AND APPLICABLE DEVICE REGISTER CONTENTS.

2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE M8203 STATIC LOGIC TESTS:

PDP-11/04,05,10,20,30,34,35,40,45,50,60, OR 70
16K MEMORY
CONSOLE TERMINAL
DMC-11 OR KMC-11 MICROPROCESSOR

M8203 LINE UNIT AND BC08S-1 CABLE AND BERG CONNECTORS
H3254 AND H3255 TEST CONNECTORS (IF NOT PRESENT, SOME TESTS
WILL BE SKIPPED)

3.0 PRELIMINARY PROGRAM REQUIREMENTS

THIS PROGRAM OPERATES THE MICROPROCESSOR EXTENSIVELY IN ORDER TO TEST THE LINE INIT. FOR THIS REASON, THE MICROPROCESSOR DIAGNOSTIC AND SUBSYSTEM FUNCTIONAL TESTS SHOULD BE RUN FIRST, AND ANY FAULTS FOUND IN THE MICROPROCESSOR MODULE SHOULD BE REPAIRED, PRIOR TO RUNNING THE M8203 STATIC LOGIC TESTS.

4.0 GENERAL PROGRAM CONSIDERATIONS

4.1 DIAGNOSTIC SUPERVISOR

THIS PROGRAM IS COMPATIBLE WITH THE STANDALONE DIAGNOSTIC SUPERVISOR, AND MUST BE LOADED TO BE CO-RESIDENT WITH THE SUPERVISOR, OR BE PREVIOUSLY COMBINED WITH THE SUPERVISOR AND LOADED AS A SINGLE FILE. IN EITHER CASE, THE COMBINED PROGRAM WILL NOT EXCEED 16K OF MEMORY.

4.2 EXECUTION TIME

THE MAXIMUM TIME REQUIRED TO RUN THE M8203 STATIC LOGIC TESTS IS ABOUT 45 SECONDS PER PASS FOR EACH UNIT.

4.3 XXDP+

THIS PROGRAM MAY BE LOADED UNDER XXDP+, AND MAY BE RUN IN DUMP MODE OR CHAIN MODE.

4.4 ACT/SLIDE

THIS PROGRAM MAY BE LOADED UNDER ACT OR SLIDE AND MAY BE RUN IN DUMP MODE OR CHAIN MODE.

4.5 APT

THIS PROGRAM MAY BE LOADED BY THE APT SYSTEM (INCLUDING APT-RD) AND RUN IN PROGRAM MODE OR SCRIPT MODE.

4.6 MEMORY MANAGEMENT

MEMORY MANAGEMENT IS NOT UTILIZED IN THIS PROGRAM. IF IT IS INSTALLED, IT IS DISABLED BY THE PROGRAM.

4.7 MEMORY PARITY OPTION

IF PARITY MEMORY IS INSTALLED, MEMORY PARITY TRAPS ARE DISABLED BY THE PROGRAM.

4.8 ERROR LOGGING

AT THE END OF EACH PASS ON ALL UNITS, THE PROGRAM PRINTS OUT THE CUMULATIVE TOTAL NUMBER OF ERRORS SINCE THE LAST START OR RESTART COMMAND.

5.0 PROGRAM LOAD MEDIA

THIS PROGRAM CAN BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER OR FROM ACT, SLIDE, OR APT SYSTEMS, OR FROM ANY MEDIA SUPPORTED BY XXDP+. WHEN USING THE PAPER TAPE ABSOLUTE LOADER, THE PROGRAM SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC SUPERVISOR. WHEN USING XXDP+, THE DIAGNOSTIC SUPERVISOR SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC PROGRAM.

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

THIS PROGRAM MAY BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER. IT MAY ALSO BE LOADED FROM ANY XXDP+ LOAD MEDIA. WHEN LOADED UNDER XXDP+, THE DIAGNOSTIC SUPERVISOR WILL BE LOADED AUTOMATICALLY.

6.1.2 STARTING PROCEDURES

THE PROGRAM STARTS AT LOCATION 200. USE STANDARD DEC PROCEDURES TO START THE PROGRAM.

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

THE DIAGNOSTIC CAN BE EXECUTED STANDALONE UNDER XXDP+, WITHOUT READING THE REMAINDER OF THIS DOCUMENT, AS FOLLOWS:

A) LOAD AND START DIAGNOSTIC USING RUN COMMAND
B) RECEIVE DIAGNOSTIC SUPERVISOR IDENTIFICATION AND PROMPT (DRS-C>)
C) ENTER STA<CR>
D) ANSWER HARDWARE AND SOFTWARE QUESTIONS
E) GET END OF PASS MESSAGES OR ERROR MESSAGES
F) TO END EXECUTION, ENTER CONTROL/C

6.2 INITIAL DIALOGUE

AFTER THE PROGRAM AND THE SUPERVISOR ARE LOADED AND THE PROGRAM IS STARTED, THE FOLLOWING IDENTIFICATION IS TYPED :

```
DRS LOADED
DIAG. RUN-TIME SERVICES
CZDMS-F-0
M8203 STATIC LOGIC TESTS - PART 2 OF 2
UNIT IS M8203
DR>
```

THE OPERATOR THEN RESPONDS BY TYPING ONE OR MORE OF THE COMMANDS DESCRIBED IN THE FOLLOWING SECTION 6.3. (FOR MORE DETAILED INFORMATION, REFER TO THE DIAGNOSTIC SUPERVISOR FUNCTIONAL SPECIFICATION).

6.3 PROGRAM OPTIONS

6.3.1 START COMMAND

```
**********************************************************************
STA(RT)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:
   <FLAG-LIST>/EOP:<INCR>
**********************************************************************
```

6.3.1.1 TESTS SWITCH (/TESTS:<TEST-LIST>)

<TEST-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS. ON THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION USED IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE OPERATOR. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.2 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION. IN THIS CASE EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURANCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT

END OF 6.3.1.5.

6.3.1.3 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR
IER	INHIBIT ERROR REPORTING
IBE	INHIBIT BASIC ERROR REPORTS
IXE	INHIBIT EXTENDED ERROR REPORTS
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER
PNT	PRINT NUMBER OF TEST BEING EXECUTED
BOE	BELL ON ERROR
UAM	RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
ISR	INHIBIT STATISTICAL REPORTS
IDU	INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
LOT	LOOP ON TEST

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.4 END OF PASS SWITCH (/EOP:<INCR>)

<INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.5 EFFECT OF START COMMAND

THE EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE PARAMETER DIALOGUE, THE SOFTWARE PARAMETER DIALOGUE, AND THEN THE DIAGNOSTIC TESTS THEMSELVES.

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION "# UNITS?" TO WHICH THE OPERATOR REPLIES WITH A DECIMAL NUMBER N FROM 1 TO 16. THE TERM "UNIT" REFERS TO THE DEVICE TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED. FOLLOWING THIS ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL BE BUILT. EACH P-TABLE IS A CORE-RESIDENT TABLE CONTAINING ALL THE HARDWARE INFORMATION FOR ONE UNIT. THE OPERATOR MUST SUPPLY N (NUMBER OF UNITS) VALUES FOR EACH QUESTION. HE MAY DO THIS BY GIVING ONE ANSWER TO EACH QUESTION (IN

WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR BY GIVING N VALUES, SEPARATED BY COMMAS, TO EACH QUESTION (SERIES WILL BE POSED ONCE). EACH QUESTION IS FOLLOWED BY THE RESPONSE RADIX (D FOR DECIMAL, B FOR BINARY, O FOR OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT VALUE AFTER THE PARENTHESES.

FOLLOWING THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS TO BUILD THE SOFTWARE TABLES, WHICH DEFINE THE MODE (QUICK VERIFY ETC.) THAT THE DIAGNOSTIC WILL EXECUTE IN.

WHEN THE QUESTION "# UNITS?" IS ANSWERED, MEMORY STORAGE IS ALLOCATED FOR THE P-TABLES, AND IF THERE IS NOT ENOUGH TO ACCOMMODATE THEM THE MESSAGE "TOO MANY UNITS" IS ISSUED. IN THIS CASE THE DIAGNOSTIC MUST BE EXECUTED MORE THAN ONCE TO TEST ALL UNITS.

EXAMPLE:

STA/TESTS:1:2-4:6:8-10/PASS:3/FLAGS:IER:HOE=1:UAM:LOE

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, EACH PASS CONSISTING OF TESTS 1,2,3,4,6,8,9, AND 10 EXECUTED AGAINST ALL UNITS. THERE IS NO DIFFERENCE BETWEEN SAYING <FLAG> AND SAYING <FLAG=1>. THE NOTATION <FLAG=0> IS MEANINGFUL ONLY ON A COMMAND OTHER THAN START TO CLEAR A FLAG THAT WAS PREVIOUSLY SET. NOTE THAT ON ALL COMMANDS ONLY THE FIRST THREE LETTERS ARE SCANNED.

6.3.2 RESTART COMMAND

****
RES(TART)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:
<FLAG-LIST>/UNITS:<UNIT-LIST>
****

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

<TEST-LIST>, <PASS-CNT>, AND <FLAG-LIST> ARE AS IN THE START COMMAND.

6.3.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (0,1 ETC.) OR RANGES OF DECIMAL NUMBERS (0-5, 8-10 ETC.) THAT SPECIFY THE UNITS TO BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS MAY RANGE FROM 0 THRU N-1 (N IS THE NUMBER OF UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE DIAGLOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP

COMMAND.

6.3.2.3 EFFECT OF RESTART COMMAND

THE RESTART COMMAND DIFFERS FROM THE START COMMAND IN THAT THE P-TABLES FROM THE PREVIOUS START COMMAND (THERE MUST HAVE BEEN ONE) ARE USED, INSTEAD OF NEW ONES BEING BUILT. THE UNITS SWITCH GIVES THE ABILITY TO SELECT A SUBSET OF THESE. THE SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED (OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE B) AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

6.3.3 CONTINUE COMMAND

****
CON(TINUE)/PASS:<PASS-CNT/FLAGS:<FLAG-LIST>
****

6.3.3.1 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS SAME AS IN START COMMAND, BUT THE DEFAULT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART. IF NONE REMAINS, THE DEFAULT IS NON-ENDING EXECUTION.

6.3.3.2 FLAG SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS SAME AS IN START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

6.3.3.3 EFFECT OF CONTINUE COMMAND

CONTINUE MUST FOLLOW A START OR RESTART, AND COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

6.3.4 PROCEED COMMAND

****
PRO(CEED)/FLAGS:<FLAG-LIST>
****

6.3.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS AS IN THE START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

6.3.4.2 EFFECT OF PROCEED COMMAND

PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE. COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS MAY BE ALTERED.

NOTE THAT IF THE MESSAGE "TOO MANY UNITS" IS ISSUED, TWO OR MORE CORE IMAGES MUST BE CREATED (WITH DIFFERENT NAMES) TO TEST ALL UNITS.

NOTE THAT ALTHOUGH THE CHAINABLE IMAGE CAN BE EXECUTED ON A 16K MACHINE, THE ORIGINAL CCI CREATION MUST BE DONE ON A LARGE MACHINE, THE EXACT SIZE BEING DEPENDENT ON WHICH UPDATE UTILITY IS USED.

6.3.5 ADD COMMAND

****
ADD/UNITS:<UNIT-LIST>
****

6.3.5.1 UNITS SWITCH (/UNITS:<UNIT-LIST>

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.5.2 EFFECT OF ADD COMMAND

THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH UNIT MUST HAVE A P-TABLE IN MEMORY DUE TO AN EARLIER HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED. THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE PREVIOUSLY DROPPED.

6.3.6 DROP COMMAND

****
DRO(P)/UNITS:<UNIT-LIST>
****

6.3.6.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

585
586 6.3.6.2 EFFECT OF DROP COMMAND
587
588 THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS
589 WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START
590 COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND
591 MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.
592
593
594 6.3.7 PRINT COMMAND
595
596 **********
597 PRI(NT)
598 **********
599
600
601 6.3.7.1 EFFECT OF PRINT COMMAND
602
603 THE TOTAL NUMBER OF ERRORS FOR EACH UNIT SINCE THE LAST
604 START OR RESTART COMMAND ARE PRINTED. THE ISR (INHIBIT
605 STATISTICAL REPORTING) FLAG IS CLEARED.
606
607
608 6.3.8 DISPLAY COMMAND
609
610 **********
611 DIS(PLAY)/UNITS:<UNIT-LIST>
612 **********
613
614
615 6.3.8.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)
616
617 <UNIT-LIST> IS AS IN THE RESTART COMMAND.
618
619
620 6.3.8.2 EFFECT OF DISPLAY COMMAND
621
622 THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED
623 OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS
624 THAT WERE DROPPED BY THE OPERATOR 'DROP'' COMMAND ARE SO
625 DESIGNATED.
626
627
628 6.3.9 FLAGS COMMAND
629
630 **********
631 FLA(GS)
632 **********
633
634
635 6.3.9.1 EFFECT OF FLAGS COMMAND
636
637 THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.
638
639
640 6.3.10 ZFLAGS COMMAND

```
641
642                     **********************************************************************
643                     ZFL(AGS)
644                     **********************************************************************
645
646                     6.3.10.1 EFFECT OF ZFLAGS COMMAND
647
648                     ALL FLAGS ARE CLEARED.
649
650
651                     6.3.11  CONTROL CHARACTERS
652
653                     A CONTROL C  (C)  ENTERED  DURING THE EXECUTION OF A
654                     DIAGNOSTIC CAUSES A RETURN TO COMMAND MODE.
655
656                     A CONTROL Z (Z) ENTERED DURING ONE  OF  THE  THREE  OPERATOR
657                     DIALOGUES-  HARD  CORE QUESTIONS (SEE 6.2),HARDWARE DIALOGUE
658                     (SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE
659                     DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.
660
661                     A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC
662                     CAUSES  ALL  TELETYPE  OUTPUT  TO  BE  SURPRESSED  FOR  THE
663                     REMAINDER  OF  THE  DIAGNOSTIC  OR UNTIL ANOTHER O IS TYPED,
664                     WHICH RESTORES NORMAL TELETYPE OUTPUT.
665
666
667                     6.3.12 HARDWARE PARAMETERS
668
669                     THE FOLLOWING 6 QUESTIONS WILL BE ASKED ON A START  COMMAND.
670                     THE  VALUE  LOCATED  TO THE LEFT OF THE QUESTION MARK IS THE
671                     DEFAULT VALUE THAT  WILL  BE  TAKEN  ON  A  CARRIAGE  RETURN
672                     RESPONSE.
673
674                     1.   DEVICE CSR ADDRESS : (O) 160170?
675
676                     THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SELO) RESIDE
677                     ON  THE  UNIBUS.   THE  ALLOWABLE  RANGE  IS   160000-177776
678                     (OCTAL), AND THE DEFAULT VALUE IS 160170.
679
680                     2.   M8203 REG 11 (E134 SW10,9 , E121 SW9,10) : (O) 0 ?
681
682                     THIS  IS  THE  EXPECTED  CONTENT  (OCTAL) OF REG 11 SWITCHES.
683                     THE ALLOWABLE RANGE IS 000-056, AND THE DEFAULT VALUE IS 000.
684                     BITS 1,2 ARE E134 SW10,9, AND BITS 3,5 ARE E121 SW9,10.
685
686                     3.   M8203 REG 15 (E134 SW8-1) : (O) 0 ?
687
688                     THIS IS THE EXPECTED CONTENT (OCTAL) OF REG 15 SWITCHES.
689                     THE ALLOWABLE RANGE IS 000-377, AND THE DEFAULT VALUE IS 000.
690                     BITS 0-7 ARE E134 SW8-1.
691
692                     4.   M8203 REG 16 (E121 SW8-1) : (O) 0 ?
693
694                     THIS IS THE EXPECTED CONTENT (OCTAL) OF REG 16 SWITCHES.
695                     THE ALLOWABLE RANGE IS 000-377, AND THE DEFAULT VALUE IS 000.
696                     BITS 0-7 ARE E121 SW8-1.
```

```
697
698                     5.   SELECT TURNAROUND TYPE; 0=H3254&H3255, 1=H325,
699                        2=H3250, 3=H3251, 4=INTEGRAL MODEM HDX SWITCH,
700                        5=MOD LOC, 6=MOD REM, 7=NONE) :  (0)   0?
701
702                     THIS INDICATOR TELLS THE PROGRAM WHETHER  TEST  CONNECTOR(S)
703                     WILL BE MOUNTED SO THAT CERTAIN TESTS CAN BE RUN IN EXTERNAL
704                     LOOPBACK MODE.  THE  ALLOWABLE  ANSWERS  ARE  0-7,  AND  THE
705                     DEFAULT  VALUE  IS 0.  0 MEANS THAT THE H3254 AND H3255 TEST
706                     CONNECTORS WILL BE USED.   1  MEANS THE H325 TEST  CONNECTOR
707                     WILL BE USED. 2 MEANS THE H3250 TEST CONNECTOR WILL BE USED.
708                     3 MEANS THE H3251 TEST CONNECTOR WILL BE USED.   4 MEANS THE
709                     INTEGRAL MODEM HDX SWITCH WILL BE USED.   5 MEANS THAT MODEM
710                     LOCAL LOOPBACK WILL BE USED.  6 MEANS THAT MODEM REMOTE LOOP
711                     BACK WILL BE USED.  7 MEANS THAT NO EXTERNAL TURNAROUND WILL
712                     BE PROVIDED.  WHEN 0 IS SELECTED, ALL TESTS WILL BE RUN, AND
713                     IF  1-7  IS SELECTED,  CERTAIN TESTS CANNOT BE RUN,  AND THE
714                     PROGRAM WILL TYPE THE NUMBER(S) OF THE TEST(S) TO BE SKIPPED.
715
716                     6.   SELECT BAUD RATE; TYPE '0' FOR 2.4K; '1' FOR 4.8K;
717                          '2' FOR 9.6K; '3' FOR 19.2K; '4' FOR 56K; '5' FOR 250K;
718                          '6' FOR 500K; OR '7' FOR 1 MEG BAUD :  (0)  4?
719
720                     THIS IS THE BAUD RATE WHICH IS SELECTED IN THE SWITCH PACK ON
721                     THE M8203. THE ALLOWABLE RANGE IS 0-7, AND THE DEFAULT VALUE IS
722                     4 (FOR 56K).
723
724
725                     6.3.13 SOFTWARE PARAMETERS
726
727                     FOUR  SOFTWARE  PARAMETER  QUESTIONS  ARE ASKED BY THE M8203
728                     STATIC LOGIC TESTS PROGRAM, PART 2. THESE QUESTIONS ARE  THE
729                     FOLLOWING:
730
731                     1.   DO MAN. INTERVEN. TO MOUNT TEST CONNECTOR(S) (L) N?
732
733                     IF THE OPERATOR ANSWERS  THE  QUESTION  WITH  Y  (YES),  THE
734                     PROGRAM  WILL  LATER  PAUSE BEFORE TESTING EACH LOGICAL UNIT
735                     AND INFORM THE OPERATOR  TO  INSTALL  THE  APPROPRIATE  TEST
736                     CONNECTOR(S)  ON  THAT  UNIT,  AND THEN PROCEED TO TEST THAT
737                     UNIT.  IF THE OPERATOR ANSWERS N (NO) TO THE ABOVE QUESTION,
738                     THE  PROGRAM  WILL  PERFORM  TESTING  ON  ALL  UNITS WITHOUT
739                     ALLOWING MANUAL INTERVENTION BETWEEN UNITS.  IN  THIS  CASE,
740                     ALL TEST CONNECTOR(S) ON ALL UNITS SHOULD BE INSTALLED PRIOR
741                     TO RUNNING THE PROGRAM.
742
743                     2.   ALLOW SWITCH PACK AND AX3-15 PRINTOUT (L) N?
744
745                     IF  THE  OPERATOR  ANSWERS  YES,  THE PROGRAM WILL ALLOW THE
746                     PRINTOUT OF SWITCHES IN REGS 11,15,16 AND MODEM INTERFACE REG AX3-15
747                     ON  ANY  PASS IN WHICH THE CORRESPONDING TESTS ARE RUN.  THE
748                     DEFAULT IS NO, WHICH ONLY ALLOWS THE PRINTOUT ON  THE  FIRST
749                     PASS AFTER LOADING, IF THE TESTS ARE RUN.
750
751                     3.   ALLOW SWITCH PACK TESTS (L) N ?
752
```

IF THE OPERATOR ANSWERS YES, THE PROGRAM WILL ALLOW THE READING AND COMPARISON OF SWITCHES IN REGS 11,15,16 TO VALUES ENTERED INTO THE HARDWARE P-TABLE FOR THIS UNIT. IF THE CORRESPONDING TEST IS RUN. IF ALLOWED, SWITCH PACK ERRORS WILL BE REPORTED. THE DEFAULT IS NO, AND THE TESTS ARE NOT RUN.

4. MSG TIMER VALUE (0-177777), 0 = LONGEST TIME-OUT : (0)
0 ?

THIS VALUE CONTROLS THE DURATION OF THE RECEIVER MESSAGE TIME-OUT IN A NUMBER OF TESTS WHICH SEND AND RECEIVE MESSAGES ON THE VARIOUS MODEM INTERFACES WITH EXTERNAL LOOPBACK. THE SMALLER THE VALUE, THE LONGER THE TIME-OUT (UP TO SEVERAL SECONDS). THE DEFAULT IS 0.

6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION "# UNITS?" IS ANSWERED (WITH THE NUMBER N, SAY) SPACE IN CORE IS ALLOCATED FOR N P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE-TO ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P-TABLE BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED. THE LAST VALUE IN THE STRING BECOMES THE NEW DEFAULT AND IS USED TO FILL THAT SLOT IN THE REMAINING P-TABLES.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR. IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6-10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 16 UNITS.

AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 16 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (0,1,2,...,15) EXCEPT FOR UNIT 12, WHICH SHOULD RECEIVE THE VALUE 11. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 7 UNITS AND THE NUMBER 77 FOR THE LAST 9 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

```
# UNITS (D) ? 16

UNIT 0
<QUESTION 1> ? 75
<QUESTION 2> ? 0-6
<QUESTION 3> ? 76

UNIT 7
<QUESTION 1> ?
<QUESTION 2> ? 7-11,,13-15
<QUESTION 3> ? 77
```

THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75 IN ALL 16 TABLES. SLOT TWO RECEIVES THE VALUES 0,1,2,...,6 IN TABLES 0 THRU 6 AND A CONSTANT 6 IN TABLES 7 THRU 15. SLOT THREE RECEIVES A CONSTANT 76 IN ALL 16 TABLES.

THE SECOND TIME THRU THE SERIES, TABLES 7 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE THE OPERATOR IN THE FORM "UNIT XX" AT THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS AT CONSTANT 75 IN TABLES 7 THRU 15, SINCE NO NEW EXPLICIT VALUES ARE TYPED IN. SLOT TWO GETS THE VALUES 7,8,9,10,11 IN TABLES 7 THRU 11, AND GETS AN 11 IN SLOT 12, AND GETS THE VALUES 13,14,15 IN TABLES 13 THRU 15. SLOT THREE GETS THE VALUE 77 IN TABLES 7 THRU 15.

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 16 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION (NAMELY QUESTION 2).

```
859
860
861
862                             7.0 DEVICE INFORMATION TABLES
863
864
865
866                             ;*****************************************************************************
867                             * MAINTENANCE REGISTER - BSEL1
868                             ;*****************************************************************************
869                             RUN     = BIT7
870                             MCLR    = BIT6
871                             STEPLU  = BIT4
872                             LULOOP  = BIT3
873                             ROMO    = BIT2
874                             ROMI    = BIT1
875                             STEPMP  = BIT0
876
877                             ;*****************************************************************************
878                             * OBUS REG 10 - TRANSMITTER BUFFER
879                             ;*****************************************************************************
880                             TX7     = BIT7
881                             TX6     = BIT6
882                             TX5     = BIT5
883                             TX4     = BIT4
884                             TX3     = BIT3
885                             TX2     = BIT2
886                             TX1     = BIT1
887                             TX0     = BIT0
888
889                             ;*****************************************************************************
890                             * OBUS REG 11
891                             ;*****************************************************************************
892                             OC      = BIT7
893                             GOAH    = BIT3
894                             ABORT   = BIT2
895                             EOM     = BIT1
896                             SOM     = BIT0
897
898                             ;*****************************************************************************
899                             * OBUS REG 12
900                             ;*****************************************************************************
901                             IC      = BIT7
902                             BPOLL   = BIT6
903                             LULP    = BIT5
904
905                             ;*****************************************************************************
906                             * OBUS REG 13
907                             ;*****************************************************************************
908                             POLL    = BIT7
909                             DTR     = BIT6
910                             SELFR   = BIT5
911                             HDX     = BIT4
912                             MAINT1  = BIT3
913                             MAINT2  = BIT2
914                             SELSBY  = BIT1
```

```
915
916                     ;*****************************************************************************
917                     * OBUS REG 14
918                     ;*****************************************************************************
919                     TXEN    = BIT6
920                     DISSI   = BIT5
921                     RDAX    = BIT4
922                     WAX     = BIT3
923                     ENAX    = BIT2
924                     AX2     = BIT1
925                     AX1     = BIT0
926
927                     ;*****************************************************************************
928                     * OBUS REG 17
929                     ;*****************************************************************************
930                     CRC2    = BIT7
931                     CRC1    = BIT6
932                     IDLE    = BIT5
933                     SECA    = BIT4
934                     STRIP   = BIT3
935                     RDALL   = BIT2
936                     IERR    = BIT1
937                     DDCMP   = BIT0
938
939                     ;*****************************************************************************
940                     * IBUS REG 10 - RECEIVER BUFFER
941                     ;*****************************************************************************
942                     RX7     = BIT7
943                     RX6     = BIT6
944                     RX5     = BIT5
945                     RX4     = BIT4
946                     RX3     = BIT3
947                     RX2     = BIT2
948                     RX1     = BIT1
949                     RX0     = BIT0
950
951                     ;*****************************************************************************
952                     * IBUS REG 11
953                     ;*****************************************************************************
954                     OC      = BIT7
955                     OACT    = BIT6
956                     SW3     = BIT5
957                     ORDY    = BIT4
958                     SW2     = BIT3
959                     SW1     = BIT2
960                     SW0     = BIT1
961                     UNRR    = BIT0
962
963                     ;*****************************************************************************
964                     * IBUS REG 12
965                     ;*****************************************************************************
966                     IC      = BIT7
967                     IACT    = BIT6
968                     LULP    = BIT5
969                     IRDY    = BIT4
970                     OVRR    = BIT3
```

```
                        RAB     = BIT2
                        EBLK    = BIT1
                        BCC     = BIT0

                        ;*****************************************************************************
                        * IBUS REG 13
                        ;*****************************************************************************
                        RING    = BIT7
                        DTR     = BIT6
                        RTS     = BIT5
                        HDX     = BIT4
                        MODR    = BIT3
                        CS      = BIT2
                        STBY    = BIT1
                        CARR    = BIT0

                        ;*****************************************************************************
                        * IBUS REG 14
                        ;*****************************************************************************
                        READY   = BIT7
                        TXEN    = BIT6
                        DISSI   = BIT5
                        RDAX    = BIT4
                        WAX     = BIT3
                        ENAX    = BIT2
                        AX2     = BIT1
                        AX1     = BIT0

                        ;*****************************************************************************
                        * IBUS REG 17
                        ;*****************************************************************************
                        SIGR    = BIT7
                        SIGQ    = BIT6
                        TXDATA  = BIT5
                        OCOR    = BIT4
                        ICIR    = BIT3
                        TESTMD  = BIT2
                        MCLK  = BIT1
                        DDCMP   = BIT0

                        ;*****************************************************************************
                        * AX0-15 - USYRT REG 0 (READ ONLY)
                        ;*****************************************************************************
                        RX7     = BIT7
                        RX6     = BIT6
                        RX5     = BIT5
                        RX4     = BIT4
                        RX3     = BIT3
                        RX2     = BIT2
                        RX1     = BIT1
                        RX0     = BIT0

                        ;*****************************************************************************
                        * AX0-16 - USYRT REG 1 (READ ONLY)
                        ;*****************************************************************************
                        RERR    = BIT7
```

```
1027                    ASBC2   = BIT6
1028                    ASBC1   = BIT5
1029                    ASBC0   = BIT4
1030                    ROR     = BIT3
1031                    RABT    = BIT2
1032                    REOM    = BIT1
1033                    RSOM    = BIT0
1034
1035                    ;*****************************************************************************
1036                    * AX1-15 - USYRT REG 2
1037                    ;*****************************************************************************
1038                    TX7     = BIT7
1039                    TX6     = BIT6
1040                    TX5     = BIT5
1041                    TX4     = BIT4
1042                    TX3     = BIT3
1043                    TX2     = BIT2
1044                    TX1     = BIT1
1045                    TX0     = BIT0
1046
1047                    ;*****************************************************************************
1048                    * AX1-16 - USYRT REG 3
1049                    ;*****************************************************************************
1050                    TERR    = BIT7
1051                    TXGA    = BIT3
1052                    TXAB    = BIT2
1053                    TEOM    = BIT1
1054                    TSOM    = BIT0
1055
1056                    ;*****************************************************************************
1057                    * AX2-15 - USYRT REG 4
1058                    ;*****************************************************************************
1059                    SYN7    = BIT7
1060                    SYN6    = BIT6
1061                    SYN5    = BIT5
1062                    SYN4    = BIT4
1063                    SYN3    = BIT3
1064                    SYN2    = BIT2
1065                    SYN1    = BIT1
1066                    SYN0    = BIT0
1067                    SYNCH   = 226
1068
1069                    ;*****************************************************************************
1070                    * AX2-16 - USYRT REG 5
1071                    ;*****************************************************************************
1072                    APA     = BIT7
1073                    DDC     = BIT6
1074                    STR     = BIT5
1075                    SEC     = BIT4
1076                    IDL     = BIT3
1077                    CRCTY2  = BIT2
1078                    CRCTY1  = BIT1
1079                    CRCTY0  = BIT0
1080
1081                    ;*****************************************************************************
1082                    * AX3-15 - USYRT REG 6
```

CZDMSF.P11 30-SEP-81 15:40 PROGRAM DOCUMENT

```
1083                    ;*****************************************************************************
1084                    I422    = BIT7
1085                    XYZ     = BIT6
1086                    C32BCC  = BIT5
1087                    V35     = BIT4
1088                    INTGRL  = BIT3
1089                    C32ENB  = BIT2
1090                    OP      = BIT1
1091                    TEST    = BIT0
1092                    AX315U  = I422!XYZ!C32BCC!V35!INTGRL!OP
1093
1094                    ;*****************************************************************************
1095                    * AX3-16 - USYRT REG 7
1096                    ;*****************************************************************************
1097                    TXLEN2  = BIT7
1098                    TXLEN1  = BIT6
1099                    TXLEN0  = BIT5
1100                    RXLEN2  = BIT2
1101                    RXLEN1  = BIT1
1102                    RXLEN0  = BIT0
1103
1104
1105
1106
1107
1108
```

```
1109
1110
1111                    8.0 TEST DESCRIPTIONS
1112
1113
1114
1115
1116
1117                    ;*************************************************************************
1118                        TEST 1 - BIT STUFFING TEST
1119                    *
1120                    * THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
1121                    * INITIATED IN BIT MODE . TWO LEADING FLAGS ARE SENT,
1122                    * FOLLOWED BY ALL SIXTEEN CHARS IN DATA PATTERN S. THIS PATTERN
1123                    * CONSISTS OF CHARACTERS WHICH REQUIRE NO BIT STUFFING AND CHARACTERS
1124                    * WHICH REQUIRE BIT STUFFING INDIVIDUALLY AND IN COMBINATION WITH
1125                    * ADJACENT CHARACTERS. ALL 16 CHARACTERS ARE READ AND COMPARED
1126                    * BY THE RECEIVER.
1127                    *   PATTERN S = 000,017,036,074,170,360,037,076,174,370,077,176,374,
1128                    *               177,376,377
1129                    ;*************************************************************************
1130
1131
1132
1133
1134
1135                    ;*************************************************************************
1136                        TEST 2 - RCV OVERRUN ERROR SET AND CLEAR TEST
1137                    *
1138                    * IN THIS TEST, A RCV OVERRUN ERROR IS FORCED IN EACH OF 2 SUBTESTS.
1139                    * IN THE FIRST, A MESSAGE IS INITIATED, 64 001 CHARS ARE SENT, AND THE
1140                    * RECEIVER IS NOT SERVICED IN RESPONSE TO THE USYRT RCV FLAG, WHICH CAUSES RCV
1141                    * OVERRUN TO SET. THEN, A CHECK IS MADE TO INSURE THAT OVRR IS NOT
1142                    * CLEARED BY THE LINE UNIT READING THE USYRT STATUS.
1143                    * THEN, IC IS SET TO CLEAR THE ERROR, A NEW 001 CHAR IS CLOCKED INTO THE
1144                    * RECVR, AND THE CLEARING OF OVRR IS VERIFIED.
1145                    *
1146                    * IN THE SECOND SUBTEST, RCV OVRUN IS FORCED AGAIN, A MASTER CLEAR
1147                    * IS ISSUED TO CLEAR THE ERROR, A NEW 001 CHAR IS CLOCKED INTO THE RECVR,
1148                    * AND THE CLEARING OF OVRR IS VERIFIED.
1149                    ;*************************************************************************
1150
1151
1152
1153
1154
1155                    ;*************************************************************************
1156                        TEST 3 - ABORT SEQUENCE TEST
1157                    *
1158                    * SET BIT MODE, CRC, AND ENABLE THE DEVICE FOR
1159                    * TRANSMIT AND RECEIVE. SEND 2 FLAGS AND 4 DATA CHARS (001).
1160                    * AS THE FIRST DATA CHAR IS BEING TRANSMITTED,
1161                    * SET THE ABORT BIT (REG 11).
1162                    * ON THE RECEIVER SIDE, CHECK FOR RECEPTION OF THE FIRST DATA CHAR
1163                    * AND THEN THE SETTING OF RAB AND REOM A CHAR TIME LATER.
1164                    * ALSO, CHECK FOR IACT = 0. THEN, CHECK THAT RAB
```

```
1165                    * IS CLEARED BY READING THE USYRT STATUS, TRANSMITTING A NEW MSG,
1166                    * RECEIVING THE FIRST CHAR (003) AND CHECKING FOR RAB CLEARED.
1167                    *
1168                    * REPEAT THE ABOVE SEQUENCE, SET IC, TRANSMIT A NEW MSG
1169                    * AND CHECK THAT THIS CLEARS RAB.
1170                    *
1171                    ;*************************************************************************
1172
1173
1174
1175
1176
1177                    ;*************************************************************************
1178                        TEST 4 - ABORT AND IDLE FLAGS TEST
1179                    *
1180                    * TRANSMIT THE SAME ABORT SEQUENCE AS IN THE PREVIOUS TEST, BUT
1181                    * WITH THE IDLE BIT SET. CHECK THAT FLAGS ARE SENT AND RECEIVED
1182                    * (NOT ABORT CHARACTERS) BY VERIFYING THAT RAB DOES
1183                    * NOT SET, AND THAT THE MESSAGE TERMINATES WITH EBLK = 1.
1184                    ;*************************************************************************
1185
1186
1187
1188
1189
1190                    ;*************************************************************************
1191                        TEST 5 - TRANSMITTER UNDERRUN ERROR, IDLE ABORT CHARS, BIT MODE
1192                    *
1193                    * A MESSAGE IS INITIATED IN BIT MODE, 4 001 CHARS ARE SENT, AND THE TRANSMITTER
1194                    * IS NOT SERVICED IN RESPONSE TO THE LAST TX FLAG, WHICH CAUSES TX
1195                    * UNDERRUN ERROR TO SET. ON THE RECEIVER SIDE, CHECK THAT THE DATA
1196                    * CHAR IS RECEIVED, AND THAT 8 CYCLES LATER THE RAB BIT SETS, AND
1197                    * THE DEVICE IDLES ABORT CHARACTERS.
1198                    ;*************************************************************************
1199
1200
1201
1202
1203
1204                    ;*************************************************************************
1205                        TEST 6 - RECEIVER DISABLE TEST
1206                    *
1207                    * TRANSMIT AND RECEIVE ARE ENABLED IN BIT MODE, AND 2 FLAGS
1208                    * ARE SENT, FOLLOWED BY 5 252 DATA CHARS. AFTER THE SECOND DATA CHAR HAS BEGUN
1209                    * TO BE RECEIVED, IC IS SET.
1210                    * THEN, THE PROGRAM CHECKS THAT A USYRT RCV FLAG IS NOT GENERATED, AND
1211                    * THE RECEIVER DATA PATH STOPS OPERATING IN THE MIDDLE OF THE CHAR.
1212                    ;*************************************************************************
1213
1214
1215
1216
1217
1218                    ;*************************************************************************
1219                        TEST 7 - ASSEMBLED BIT COUNT TEST
1220                    *
```

```
1221                    * THE FOLLOWING SEQUENCE IS PERFORMED 8 TIMES, EACH TIME USING A
1222                    * DIFFERENT TX CHAR LENGTH (FROM 2 TO 8 BITS) AND A RCV CHAR LENGTH = 8
1223                    * BITS :
1224                    * A MESSAGE IS INITIATED IN BIT MODE, NO CRC.
1225                    * 2 FLAGS ARE SENT, FOLLOWED BY 3 000 DATA CHARACTERS AND A
1226                    * TERMINATING FLAG. AFTER THE RECEIVER HAS RECEIVED THE MESSAGE, AX0-16
1227                    * IS READ TO RETRIEVE THE ASSEMBLED BIT COUNT. THIS COUNT IS CHECKED TO INSURE
1228                    * THAT IT IS CORRECT FOR THE TX CHAR LENGTH USED IN THAT TRANSMISSION.
1229                    ;*************************************************************************
1230
1231
1232
1233
1234
1235                    ;*************************************************************************
1236                        TEST 8 - SECONDARY STATION ADDRESS BIT TEST
1237                    *
1238                    * FIRST, A MASTER CLEAR IS ISSUED. THEN, THE LINE UNIT IS PLACED IN
1239                    * BIT MODE, AND THE SECA BIT (REG 17) IS SET.
1240                    * 2 FLAGS ARE SENT, FOLLOWED BY 252, 000, AND A TERMINATING FLAG.
1241                    * THEN, THE RECEIVER IS CHECKED TO MAKE SURE THAT NO DATA CHARS ARE
1242                    * RECEIVED.
1243                    *
1244                    * NEXT, THE SECONDARY STATION ADDRESS BITS IN AX2-15 ARE LOADED
1245                    * WITH THE FIRST WORD OF DATA PATTERN T . 2 FLAGS ARE SENT,
1246                    * FOLLOWED BY THE FIRST WORD OF DATA PATTERN T, A 000 CHAR,
1247                    * AND A TERMINATING FLAG.
1248                    * THEN, THE RCV'D DATA IS CHECKED TO MAKE SURE THAT THE SEC STATION
1249                    * ADDRESS IS RCV'D AS THE FIRST DATA CHAR, FOLLOWED BY 000.
1250                    *
1251                    * THEN, THE SUBTEST IS REPEATED FOR EACH OF THE REMAINING WORDS OF
1252                    * DATA PATTERN T.
1253                    *   PATTERN T = 000,125,252,176,177
1254                    ;*************************************************************************
1255
1256
1257
1258
1259
1260                    ;*************************************************************************
1261                        TEST 9 - RDALL (ALL PARTIES ADDRESS) BIT TEST
1262                    *
1263                    * FIRST, A MASTER CLEAR IS ISSUED. THEN, THE LINE UNIT IS PLACED IN
1264                    * BIT MODE, AND THE SECA BIT IS SET.
1265                    * 2 FLAGS ARE SENT, FOLLOWED BY 377, 125, AND A TERMINATING FLAG.
1266                    * THEN, THE RECEIVER IS CHECKED TO MAKE SURE THAT NO DATA CHARS ARE
1267                    * RECEIVED.
1268                    * NEXT, THE RDALL BIT IN REG 17 IS SET TO 1. 2 FLAGS
1269                    * ARE SENT, FOLLOWED BY 377, 125, AND A TERMINATING FLAG.
1270                    * THEN, THE REC'D DATA IS CHECKED TO MAKE SURE THAT 377
1271                    * IS REC'D AS THE FIRST DATA CHAR, FOLLOWED BY 125.
1272                    ;*************************************************************************
1273
1274
1275
1276
```

```
1277
1278                    ;*************************************************************************
1279                        TEST 10 - INSERT ERROR (IERR) BIT TEST - CHAR MODE, NO CRC
1280                    *
1281                    * THE LINE UNIT IS PLACED IN DDCMP MODE WITH NO ERROR DETECTION, AND 2
1282                    * SYNCHS, A 000 CHAR, A 377 CHAR, AND 2 SYNCHS ARE LOADED INTO THE
1283                    * TRANSMITTER SILO. THEN, THE LU IS CLOCKED UNTIL THE 2ND BIT OF THE 000
1284                    * CHAR IS ABOUT TO BE SENT AND THE IERR BIT IS SET FOR A CLOCK TIME AND
1285                    * THEN CLEARED. IN THE SAME WAY, IERR IS SET PRIOR TO THE SENDING OF THE 4TH
1286                    * AND 5TH BITS OF THE 000 CHAR. IT IS ALSO SET FOR THE SENDING OF THE FIRST
1287                    * 4 BITS OF THE 377 CHAR. THE PROGRAM READS THE FIRST RCV'D CHAR FROM AX0
1288                    * AND CHECKS IT TO BE 032, AND READS THE 2ND CHAR AND CHECKS IT TO BE 377.
1289                    * THEN, A MASTER CLEAR IS DONE TO IDLE THE DEVICE.
1290                    ;*************************************************************************
1291
1292
1293
1294
1295
1296                    ;*************************************************************************
1297                        TEST 11 - SWITCH PACK PRINTOUT AND TEST
1298                    *
1299                    * - READ AND PRINT SWITCHES IN REG 11 (E134 SW10,9 , E121 SW9,10) :
1300                    * THE PROGRAM READS REG 11 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
1301                    * (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
1302                    * THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). BITS 1,2
1303                    * ARE E121 SW10,9 , AND BITS 3,5 ARE E134 SW9,10.
1304                    *
1305                    * - READ AND PRINT SWITCHES IN REG 15 (E134 SW8-1) :
1306                    * THE PROGRAM READS REG 15 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
1307                    * (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
1308                    * THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). BITS 0-7 ARE E134 SW1-8.
1309                    *
1310                    * - READ AND PRINT SWITCHES IN REG 16 (E121 SW8-1) :
1311                    * THE PROGRAM READS REG 16 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
1312                    * (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
1313                    * THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). BITS 0-7 ARE E121 SW1-8.
1314                    ;*************************************************************************
1315
1316
1317
1318
1319
1320                    ;*************************************************************************
1321                        TEST 12 - REG AX3-15 PRINTOUT
1322                    *
1323                    * IN THIS TEST, REG AX3-15 IS READ AND THE CONTENTS PRINTED OUT IF DESIRED BY
1324                    * THE OPERATOR, AS INDICATED IN THE SOFTWARE P-TABLE. THE DEFAULT IS TO NOT
1325                    * PRINT THE REG.
1326                    ;*************************************************************************
1327
1328
1329
1330
1331
1332                    ;*************************************************************************
```

```
1333                                    TEST 13 - CRC GENERATION TEST
1334                                *
1335                                * - CRC-16, CHAR MODE:
1336                                * THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE WITH CRC-16 SELECTED -
1337                                * 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOOP AND STEPLU
1338                                * TO CLOCK THE DATA.    AT THE END OF THE MESSAGE THE
1339                                * PROGRAM CHECKS FOR BCC = 1 (IN REG 12) INDICATING NO ERROR.
1340                                *
1341                                * - CRC-CCITT - 1'S PRESET:
1342                                * THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED. AT
1343                                * THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.
1344                                *
1345                                * - CRC-CCITT - 0'S PRESET:
1346                                * THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED. AT
1347                                * THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.
1348                                ;*****************************************************************************
1349
1350
1351
1352
1353
1354                                ;*****************************************************************************
1355                                    TEST 14 - CRC ERROR DETECTION TEST
1356                                *
1357                                * - CRC-16, CHAR MODE :
1358                                * THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE, WITH CRC-16 SELECTED -
1359                                * 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOOP AND STEPLU
1360                                * TO CLOCK THE DATA. JUST BEFORE THE FIRST BIT OF THE LAST 000 CHAR IS SENT,
1361                                * THE IERR BIT IS SET IN REG 17 TO CAUSE A 1 TO BE SENT, INTRODUCING A DATA
1362                                * ERROR. AT THE END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 0, INDICATING
1363                                * AN ERROR.
1364                                *
1365                                * - CRC-CCITT - 1'S PRESET :
1366                                * THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED. AT THE
1367                                * END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
1368                                *
1369                                * - CRC-CCITT - 0'S PRESET :
1370                                * THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED. AT THE
1371                                * END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
1372                                ;*****************************************************************************
1373
1374
1375
1376
1377
1378                                ;*****************************************************************************
1379                                    TEST 15 - VRC PARITY GENERATION TEST
1380                                *
1381                                * SUBTEST 1 - TEST OF CORRECT ODD VRC PARITY GENERATION :
1382                                * THE LINE UNIT IS PLACED IN CHAR MODE, WITH ODD VRC AND 7-BIT CHARS SELECTED.
1383                                * THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)
1384                                * OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.
1385                                * FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 1 AND FOR THE
1386                                * LAST 4 CHARS IT SHOULD = 0.
1387                                *
1388                                * SUBTEST 2 - TEST OF CORRECT EVEN VRC PARITY GENERATION :
```

```
1389                    * THE LINE UNIT IS PLACED IN CHAR MODE, WITH EVEN VRC AND 7-BIT CHARS SELECTED.
1390                    * THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)
1391                    * OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.
1392                    * FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 0 AND FOR THE
1393                    * LAST 4 CHARS IT SHOULD = 1.
1394                    *
1395                    *    DATA PATTERN Q = 000,120,125,137,040,052,057,177
1396                    ;*****************************************************************************
1397
1398
1399
1400
1401
1402                    ;*****************************************************************************
1403                        TEST 16 - VRC ERROR DETECTION TEST
1404                    *
1405                    * SUBTEST 1 - FORCING OF BCC USING ODD VRC
1406                    * THE LINE UNIT IS PLACED IN CHAR MODE WITH ODD VRC AND 7-BIT CHARS SELECTED.
1407                    * THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER
1408                    * 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM
1409                    * CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING
1410                    * NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE
1411                    * RECEIVED (INDICATING AN ERROR).
1412                    *
1413                    * SUBTEST 2 - FORCING OF BCC USING EVEN VRC
1414                    * THE LINE UNIT IS PLACED IN CHAR MODE WITH EVEN VRC AND 7-BIT CHARS SELECTED.
1415                    * THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER
1416                    * 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM
1417                    * CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING
1418                    * NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE
1419                    * RECEIVED (INDICATING AN ERROR).
1420                    *
1421                    *   DATA PATTERN R = 000,100,120,124,164,172,176,177,000,100,120,124,164,
1422                    *                    172,176.
1423                    ;*****************************************************************************
1424
1425
1426
1427
1428
1429                    ;*****************************************************************************
1430                        TEST 17 - INTEGRAL MODEM INTERFACE TEST - CHAR MODE, CRC
1431                    *
1432                    * THE INTEGRAL MODEM IS SELECTED BY THE PROGRAM IN AX3-15, AND A
1433                    * MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
1434                    * ON THE LINE UNIT OR AT THE CABLE. THE MESSAGE CONSISTS OF
1435                    * 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
1436                    * UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
1437                    * SKIPPED FOR THAT UNIT.
1438                    ;*****************************************************************************
1439
1440
1441
1442
1443
1444                    ;*****************************************************************************
```

CZDMSF.P11 30-SEP-81 15:40 PROGRAM DOCUMENT

```
1445                        TEST 18 - V.35 MODEM INTERFACE TEST - CHAR MODE, CRC
1446                    *
1447                    * THE V.35 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A
1448                    * MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
1449                    * ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
1450                    * OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
1451                    * 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
1452                    * UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
1453                    * SKIPPED FOR THAT UNIT.
1454                    ;*****************************************************************************
1455
1456
1457
1458
1459
1460                    ;*****************************************************************************
1461                        TEST 19 - RS 232C AND RS 423 MODEM INTERFACE TEST - CHAR MODE, CRC
1462                    *
1463                    * THE RS232C  RS423 (XYZ) MODEM INTERFACE IS SELECTED BY THE PROGRAM IN
1464                    * AX3-15, AND A MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURN-
1465                    * AROUND CONNECTOR ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
1466                    * OR A MODEM TEST MODE. THE MESSAGE CONSISTS
1467                    * OF 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE
1468                    * P-TABLE FOR THE CURRENT UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS
1469                    * PROVIDED, THE TEST WILL BE SKIPPED FOR THAT UNIT.
1470                    * TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
1471                    * RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
1472                    ;*****************************************************************************
1473
1474
1475
1476
1477
1478                    ;*****************************************************************************
1479                        TEST 20 - RS 422 MODEM INTERFACE TEST - CHAR MODE, CRC
1480                    *
1481                    * THE RS 422 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A
1482                    * MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
1483                    * ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
1484                    * OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
1485                    * 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
1486                    * UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
1487                    * SKIPPED FOR THAT UNIT.
1488                    * TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
1489                    * RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
1490                    ;*****************************************************************************
1491
1492
1493
1494
1495
1496                    ;*****************************************************************************
1497                        TEST 21 - HALF-DUPLEX BIT (HALF DUPX) TEST
1498                    *
1499                    * THIS TEST VERIFIES THAT SETTING HALF-DUPLEX BIT IN REG 13 DOES NOT INHIBIT
1500                    * LOADING OF THE USYRT TRANSMITTER FROM THE TRANSMITTER SILO.
```

```
1501                    * A MASTER CLEAR IS ISSUED, DDCMP MODE IS ENTERED, AND THE HALF DUPX
1502                    * BIT IN REG 13 IS SET. A MESSAGE IS LOADED INTO THE TX SILO
1503                    * CONSISTING OF 2 SYNCHS, 000,125,252,377,000, AND 2 MORE SYNCHS.
1504                    * THE LINE UNIT IS THEN CLOCKED EXTENSIVELY, AND THE TX SILO IS CHECKED TO
1505                    * BE UNLOADED (ALL CHARS SHOULD HAVE BEEN REMOVED) AND THE RECEIVER
1506                    * IS MONITORED TO INSURE THAT NO RCV FLAGS ARE GENERATED.
1507                    ;*************************************************************************
1508
1509
1510
1511
1512
1513                    ;*************************************************************************
1514                        TEST 22 - HALF-DUPLEX RCV DISABLED TEST WITH SILOS DISABLED
1515                    *
1516                    * THIS TEST SENDS A MESSAGE IN HDX, CHAR MODE, WITH NO ERROR DETECTION, AND
1517                    * THE SILOS DISABLED. THE MSG CONSISTS OF 2 SYNCHS AND 2 000 CHARS.
1518                    * THE DATA IS SENT WITH LULOOP SET FOR TTL DATA LOOPBACK. THE PROGRAM CHECKS
1519                    * THAT THE RECEIVER NEVER BECOMES ACTIVE, BECAUSE THE RCV CLOCK IS INHIBITED
1520                    * WHEN THE HDX BIT IS SET.
1521                    ;*************************************************************************
1522
1523
1524
1525
1526
1527                    ;*************************************************************************
1528                        TEST 23 - INTERACTION OF MODEM CONTROL BITS
1529                    *
1530                    * THIS TEST WILL BE RUN ONLY IF THE P-TABLE FOR THIS UNIT INDICATES THAT EITHER
1531                    * THE H3254 AND H3255, THE H325, THE H3250, OR THE H3251 TEST CONNECTORS ARE
1532                    * INSTALLED.  OTHERWISE, THE TEST WILL BE SKIPPED FOR THE UNIT.
1533                    * SUBTESTS 2 THRU 6 ARE SKIPPED IF AN H325 OR H3250 TEST CONNECTOR IS INSTALLED.
1534                    * THE FOLLOWING SUBTESTS ARE PERFORMED:
1535                    1 - A MASTER CLEAR IS DONE AND REG 13 IS READ AND CHECKED FOR INITIALIZED
1536                    *   STATE, WITH LULOOP SET TO 1. THEN, LULOOP IS CLEARED AND REG 13 IS READ
1537                    *   AND CHECKED FOR THE PROPER STATE, WITH LULOOP CLEARED.
1538                    *   REG 13 IS THEN LOADED WITH 0'S, AND READ AND CHECKED FOR THE INITIALIZED
1539                    *   STATE.
1540                    *   REG 17 IS THEN READ AND CHECKED FOR INITIALIZED STATE.
1541                    2 - RUN IS SET IN BSEL1, AND REG 13 IS READ AND CHECKED FOR RING SET.
1542                    3 - POLL IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR SIGQ SET.
1543                    4 - SELFR IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR SIGR SET.
1544                    5 - MAINT1 IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR TEST MODE SET.
1545                    6 - SELSBY IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR STBY SET.
1546                    7 - DTR IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR DTR AND MODR SET.
1547                    *   IF USING H325 TEST CONNECTOR, REG 13 IS ALSO CHECKED FOR RING SET.
1548                    8 - BPOLL IS SET IN REG 12, ONLY TO LIGHT THE LED FOR THIS SIGNAL.
1549                    9 - HDX IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR HDX SET.
1550                    10 - A MASTER CLEAR IS DONE, 2 TSOM'S ARE LOADED INTO THE TX SILO, THE LINE
1551                    *   UNIT IS CLOCKED UNTIL THE TRANSMITTER IS ACTIVE, AND REG 13 IS READ AND
1552                    *   CHECKED FOR RTS, CS, CARR SET.
1553                    ;*************************************************************************
1554
1555
1556
```

```
1557
1558
1559                    ;*************************************************************************
1560                        TEST 24 - DATA TEST - BIT MODE, NO ERR DET
1561                    *
1562                    * A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH ERROR DETECTION
1563                    * INHIBITED. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
1564                    * AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
1565                    * THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
1566                    * IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
1567                    * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
1568                    * TEST WILL NOT BE RUN.
1569                    *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
1570                    *               375,373,367,357,337,277,177
1571                    * 8-BIT CHARACTERS ARE USED.
1572                    * TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
1573                    * RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
1574                    ;*************************************************************************
1575
1576
1577
1578
1579
1580                    ;*************************************************************************
1581                        TEST 25 - DATA TEST - CHAR MODE, NO ERR DET
1582                    *
1583                    * A MESSAGE IS INITIATED IN CHAR MODE, WITH ERROR DETECTION
1584                    * INHIBITED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
1585                    * AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
1586                    * THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
1587                    * IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
1588                    * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
1589                    * TEST WILL NOT BE RUN.
1590                    *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
1591                    *               375,373,367,357,337,277,177
1592                    * 8-BIT CHARACTERS ARE USED.
1593                    * TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
1594                    * RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
1595                    ;*************************************************************************
1596
1597
1598
1599
1600
1601                    ;*************************************************************************
1602                        TEST 26 - DATA TEST - BIT MODE, CRC-CCITT-1
1603                    *
1604                    * A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH CRC-CCITT-1 ERROR
1605                    * DETECTION. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
1606                    * AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
1607                    * THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
1608                    * IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
1609                    * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
1610                    * TEST WILL NOT BE RUN.
1611                    *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
1612                    *               375,373,367,357,337,277,177
```

```
1613                    * 8-BIT CHARACTERS ARE USED.
1614                    * TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
1615                    * RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
1616                    ;*************************************************************************
1617
1618
1619
1620
1621
1622                    ;*************************************************************************
1623                        TEST 27 - DATA TEST - BIT MODE, CRC-CCITT-0
1624                    *
1625                    * A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH CRC-CCITT-0 ERROR
1626                    * DETECTION. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
1627                    * AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
1628                    * THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
1629                    * IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
1630                    * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
1631                    * TEST WILL NOT BE RUN.
1632                    *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
1633                    *               375,373,367,357,337,277,177
1634                    * 8-BIT CHARACTERS ARE USED.
1635                    * TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
1636                    * RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
1637                    ;*************************************************************************
1638
1639
1640
1641
1642
1643                    ;*************************************************************************
1644                        TEST 28 - DATA TEST - CHAR MODE, CRC-16
1645                    *
1646                    * A MESSAGE IS INITIATED IN CHAR MODE, WITH CRC-16 ERROR
1647                    * DETECTION. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
1648                    * AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
1649                    * THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
1650                    * IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
1651                    * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
1652                    * TEST WILL NOT BE RUN.
1653                    *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
1654                    *               375,373,367,357,337,277,177
1655                    * 8-BIT CHARACTERS ARE USED.
1656                    * TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
1657                    * RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
1658                    ;*************************************************************************
1659
1660
1661
1662
1663
1664                    ;*************************************************************************
1665                        TEST 29 - DATA TEST - CHAR MODE, ODD VRC
1666                    *
1667                    * A MESSAGE IS INITIATED IN CHAR MODE, WITH ODD VRC ERROR DETECTION
1668                    * SELECTED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
```

```
1669                    * AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
1670                    * THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
1671                    * IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
1672                    * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
1673                    * TEST WILL NOT BE RUN.
1674                    *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
1675                    *               375,373,367,357,337,277,177
1676                    * 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).
1677                    * TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
1678                    * RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
1679                    ;*************************************************************************
1680
1681
1682
1683
1684
1685                    ;*************************************************************************
1686                        TEST 30 - DATA TEST - CHAR MODE, EVEN VRC
1687                    *
1688                    * A MESSAGE IS INITIATED IN CHAR MODE, WITH EVEN VRC ERROR DETECTION
1689                    * SELECTED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
1690                    * AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
1691                    * THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
1692                    * IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
1693                    * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
1694                    * TEST WILL NOT BE RUN.
1695                    *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
1696                    *               375,373,367,357,337,277,177
1697                    * 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).
1698                    * TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
1699                    * RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
1700                    ;*************************************************************************
1701
1702
1703
1704
1705
1706                    ;*************************************************************************
1707                        TEST 31 - CONTIGUOUS ONES IN SEC. STA. ADRS. MODE, BIT MODE
1708                    *
1709                    * IN THIS TEST, A MESSAGE CONSISTING OF 5 ONES CHARS (377 OCT)
1710                    * IS SENT IN SECONDARY STATION ADDRESS MODE, WITH THE STATION ADRS
1711                    * FOR THIS LINE = 377. THE PROGRAM CHECKS FOR CORRECT RECEPTION OF
1712                    * THE FIRST CHARACTER (STATION ADDRESS) AND THE REMAINING 4
1713                    * ONES CHARACTERS (DATA). THIS TEST EXERCISES THE SECONDARY STATION
1714                    * ADDRESS LOGIC, AND CHECKS THAT THE SEC. STA. ADRS. CAN BE BIT-STUFFED
1715                    * AND TRANSMITTED AND RECEIVED CORRECTLY.
1716                    ;*************************************************************************
1717
1718
1719
1720
1721
1722                    ;*************************************************************************
1723                        TEST 32 - DDCMP MESSAGE TEST - CHAR MODE
1724                    *
```

```
1725                    * IN THIS TEST, THREE USYRT MESSAGES ARE SENT TO SIMULATE A DDCMP HEADER,
1726                    * DDCMP DATA MESSAGE, AND THE START OF A NEW DDCMP HEADER.
1727                    * FIRST, THE DATA IN PATTERN A IS TRANSMITTED AND RECEIVED
1728                    * AND THEN CRC (CRC-16) IS SENT, FOLLOWED BY THE DATA IN PATTERN A
1729                    * AGAIN AND THE CRC ON THAT DATA, AND FINALLY THE DATA IN 'MSG1" IS
1730                    * SENT WITH ITS CORRESPONDING CRC.
1731                    *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
1732                    *               375,373,367,357,337,277,177
1733                    *   MSG1 = SYNCH,SYNCH,SYNCH,SYNCH,000,125,252,377,SYNCH,SYNCH
1734                    ;*****************************************************************************
1735
1736
1737
1738
1739
1740                    8.1 DATA PATTERNS USED
1741
1742
1743
1744                    ***** DATA PATTERN A *****
1745                    PATA:
1746                            .BYTE   125
1747                            .BYTE   252
1748                            .BYTE   000
1749                            .BYTE   377
1750                            .BYTE   001
1751                            .BYTE   002
1752                            .BYTE   004
1753                            .BYTE   010
1754                            .BYTE   020
1755                            .BYTE   040
1756                            .BYTE   100
1757                            .BYTE   200
1758                            .BYTE   376
1759                            .BYTE   375
1760                            .BYTE   373
1761                            .BYTE   367
1762                            .BYTE   357
1763                            .BYTE   337
1764                            .BYTE   277
1765                            .BYTE   177
1766
1767                    ***** DATA PATTERN Q *****
1768                    PATQ:   .BYTE   000
1769                            .BYTE   120
1770                            .BYTE   125
1771                            .BYTE   137
1772                            .BYTE   040
1773                            .BYTE   052
1774                            .BYTE   057
1775                            .BYTE   177
1776
1777                    ***** DATA PATTERN R *****
1778                    PATR:   .BYTE   000
1779                            .BYTE   100
1780                            .BYTE   120
```

CZDMSF.P11 30-SEP-81 15:40 PROGRAM DOCUMENT

```
1781                            .BYTE   124
1782                            .BYTE   164
1783                            .BYTE   172
1784                            .BYTE   176
1785                            .BYTE   177
1786                            .BYTE   000
1787                            .BYTE   100
1788                            .BYTE   120
1789                            .BYTE   124
1790                            .BYTE   164
1791                            .BYTE   172
1792                            .BYTE   176
1793
1794                    ***** DATA PATTERN S *****
1795                    PATS:   .BYTE   000
1796                            .BYTE   017
1797                            .BYTE   036
1798                            .BYTE   074
1799                            .BYTE   170
1800                            .BYTE   360
1801                            .BYTE   037
1802                            .BYTE   076
1803                            .BYTE   174
1804                            .BYTE   370
1805                            .BYTE   077
1806                            .BYTE   176
1807                            .BYTE   374
1808                            .BYTE   177
1809                            .BYTE   376
1810                            .BYTE   377
1811
1812                    ***** DATA PATTERN T *****
1813                    PATT:   .BYTE   000
1814                            .BYTE   125
1815                            .BYTE   252
1816                            .BYTE   176
1817                            .BYTE   177
1818
1819
1820
```

1821

9.0 ERROR INFORMATION

9.1 ERROR REPORTING

ERRORS ARE REPORTED BY THE PROGRAM AS THEY OCCUR (IF NOT INHIBITED). THE REPORT CONFORMS TO THE DIAGNOSTIC SUPERVISOR ERROR REPORT FORMAT, AND CONSISTS OF A DESCRIPTION OF THE ERROR, THE TEST NUMBER, SUBTEST NUMBER, PC OF THE ERROR CALL, DEVICE ADDRESS, AND BASIC AND EXTENDED ERROR INFORMATION.

THE FOLLOWING EXAMPLE PROVIDES A TYPICAL ERROR REPORT, WHICH DESCRIBES AN "IRDY NOT SET" ERROR, AND PROVIDES THE PC OF THE ERROR CALL AND THE PC OF THE CALL TO THE SUBROUTINE REPORTING IT, THE FAILING REGISTER NAME, AND DEVICE REGISTER CONTENTS :

```
  CZDMS DVC FTL ERR 00017 ON UNIT 00 TST 034 SUB 000 PC:  006210
  IRDY NOT SET
  PC OF SUBR CALL:  030044
  DEVICE CSR ADDRESS : 160170

  FAILING REG:  INBUS/OUTBUS REG 12

  LINE UNIT INBUS REGS:
  REG10   REG11   REG12   REG13
  000     120     000     257
      REG14   REG15   REG16   REG17
      024     377     377     035

  LINE UNIT EXTENDED REGS:
  AX0-15  AX0-16  AX1-15  AX1-16
  000     000     000     000
      AX2-15  AX2-16  AX3-15  AX3-16
      000     000     000     000
```

FOR OTHER ERRORS, THE REPORT MAY BE MORE EXTENSIVE, AND REQUIRE ADDITIONAL DATA TO BE REPORTED.

IF EXTENDED ERROR INFORMATION HAD BEEN INHIBITED USING THE IXE FLAG PRIOR TO RUNNING THE TEST, THE ABOVE ERROR WOULD HAVE BEEN REPORTED IN THE FOLLOWING SHORTENED FORM :

```
  CZDMS DVC FTL ERR 00017 ON UNIT 00 TST 034 SUB 000 PC:006210
  IRDY NOT SET
  PC OF SUBR CALL:  030044
  DEVICE CSR ADDRESS : 160170

  FAILING REG:  INBUS/OUTBUS REG 12
```

```
1874
1875
1876
1877
1878                    a
1879
1880
1881
1882
1883
```

```
CZDMSF.P11    30-SEP-81 15:40          PROGRAM DOCUMENT

                                       .TITLE CZDMSF M8203 STATIC TESTS #2
          002000                               .=2000



                                               .MCALL  SVC
002000                                         SVC                                   ; INITIALIZE SUPERVISOR MACROS


002000                                         BGNMOD  LU2MOD

          000001                       $LSTIN= 1
          000001                       $LSTTAG= 1
          000001                       SVCINS= 1       ; LIST INSTRUCTIONS, SHIFTED RIGHT
          000001                       SVCTST= 1       ; LIST TEST TAGS, SHIFTED RIGHT
          000001                       SVCSUB= 1       ; LIST SUBTEST TAGS, SHIFTED RIGHT
          000001                       SVCGBL= 1       ; LIST GLOBAL TAGS, SHIFTED RIGHT
          000001                       SVCTAG= 1       ; LIST OTHER TAGS, SHIFTED RIGHT

                                       ;     CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
                                       ;     TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS.  CHANGE THE
                                       ;     SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS.  YOU MAY
                                       ;     CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
```

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

002000                          POINTER BGNSW,BGNSFT,BGNAU,BGNDU

                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                        ;       IF ANY OPTIONAL POINTERS ARE TO BE USED IN THE 'HEADER', CHANGE
                        ;       'POINTER' TO CONTAIN THE CORRECT ARGUMENTS.  IF ALL OPTIONAL
                        ;       POINTERS ARE TO BE USED, CHANGE 'POINTER' TO BE 'POINTER ALL'.
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

002000                          HEADER  CZDMS,F,0,45.,0
002000                                                  L$NAME::
002000     103                                                  .ASCII /C/
002001     132                                                  .ASCII /Z/
002002     104                                                  .ASCII /D/
002003     115                                                  .ASCII /M/
002004     123                                                  .ASCII /S/
002005     000                                                  .BYTE   0
002006     000                                                  .BYTE   0
002007     000                                                  .BYTE   0
002010                                                  L$REV::
002010     106                                                  .ASCII  /F/
002011                                                  L$DEPO::
002011     060                                                  .ASCII  /0/
002012                                                  L$UNIT::
002012  000000                                                  .WORD   0
002014                                                  L$TIML::
002014  000055                                                  .WORD   45.
002016                                                  L$HPCP::
002016  036710                                                  .WORD   L$HARD
002020                                                  L$SPCP::
002020  037634                                                  .WORD   L$SOFT
002022                                                  L$HPTP::
002022  002226                                                  .WORD   L$HW
002024                                                  L$SPTP::
002024  002244                                                  .WORD   L$SW
002026                                                  L$LADP::
002026  040244                                                  .WORD   L$LAST
002030                                                  L$STA::
002030  000000                                                  .WORD   0
002032                                                  L$CO::
002032  000000                                                  .WORD   0
002034                                                  L$DTYP::
002034  000000                                                  .WORD   0
002036                                                  L$APT::
002036  000000                                                  .WORD   0
002040                                                  L$DTP::
002040  002124                                                  .WORD   L$DISPATCH
002042                                                  L$PRIO::
002042  000000                                                  .WORD   0
002044                                                  L$ENVI::
002044  000000                                                  .WORD   0
```

```
002046                  L$EXP1::
002046  000000                  .WORD   0
002050                  L$MREV::
002050     003                  .BYTE   C$REVISION
002051     003                  .BYTE   C$EDIT
002052                  L$EF::
002052  000000                  .WORD   0
002054  000000                  .WORD   0
002056                  L$SPC::
002056  000000                  .WORD   0
002060                  L$DEVP::
002060  003146                  .WORD   L$DVTYP
002062                  L$REPP::
002062  000000                  .WORD   0
002064                  L$EXP4::
002064  000000                  .WORD   0
002066                  L$EXP5::
002066  000000                  .WORD   0
002070                  L$AUT::
002070  023434                  .WORD   L$AU
002072                  L$DUT::
002072  023352                  .WORD   L$DU
002074                  L$LUN::
002074  000000                  .WORD   0
002076                  L$DESP::
002076  003154                  .WORD   L$DESC
002100                  L$LOAD::
002100  104035                  EMT     E$LOAD
002102                  L$ETP::
002102  000000                  .WORD   0
002104                  L$ICP::
002104  022144                  .WORD   L$INIT
002106                  L$CCP::
002106  023350                  .WORD   L$CLEAN
002110                  L$ACP::
002110  023270                  .WORD   L$AUTO
002112                  L$PRT::
002112  022136                  .WORD   L$PROT
002114                  L$TEST::
002114  000000                  .WORD   0
002116                  L$DLY::
002116  000000                  .WORD   0
002120                  L$HIME::
002120  000000                  .WORD   0

                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                ;       CHANGE THE 'HEADER' TO CONTAIN THE PROPER ARGUMENTS.
                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                .EVEN
```

```
CZDMSF.P11    30-SEP-81 15:40          DISPATCH TABLE

  2028                                 .SBTTL  DISPATCH TABLE
  2029
  2030                                 ;////////////////////////////////////////////////////////////////////////////////
  2031                                 ;/ THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
  2032                                 ;/ IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
  2033                                 ;////////////////////////////////////////////////////////////////////////////////
  2034
  2035  002122                               DISPATCH 32
  2036  002122  000040                                                          .WORD   32
  2037  002124                                                         L$DISPATCH::
  2038  002124  023436                                                          .WORD   T1
  2039  002126  023566                                                          .WORD   T2
  2040  002130  024224                                                          .WORD   T3
  2041  002132  024456                                                          .WORD   T4
  2042  002134  024540                                                          .WORD   T5
  2043  002136  024630                                                          .WORD   T6
  2044  002140  024734                                                          .WORD   T7
  2045  002142  025504                                                          .WORD   T8
  2046  002144  025744                                                          .WORD   T9
  2047  002146  026150                                                          .WORD   T10
  2048  002150  026304                                                          .WORD   T11
  2049  002152  026756                                                          .WORD   T12
  2050  002154  027072                                                          .WORD   T13
  2051  002156  027430                                                          .WORD   T14
  2052  002160  030040                                                          .WORD   T15
  2053  002162  030512                                                          .WORD   T16
  2054  002164  031214                                                          .WORD   T17
  2055  002166  031476                                                          .WORD   T18
  2056  002170  031750                                                          .WORD   T19
  2057  002172  032134                                                          .WORD   T20
  2058  002174  032320                                                          .WORD   T21
  2059  002176  032424                                                          .WORD   T22
  2060  002200  032642                                                          .WORD   T23
  2061  002202  034272                                                          .WORD   T24
  2062  002204  034464                                                          .WORD   T25
  2063  002206  034656                                                          .WORD   T26
  2064  002210  035050                                                          .WORD   T27
  2065  002212  035242                                                          .WORD   T28
  2066  002214  035434                                                          .WORD   T29
  2067  002216  035622                                                          .WORD   T30
  2068  002220  036010                                                          .WORD   T31
  2069  002222  036122                                                          .WORD   T32
  2070
  2071                                 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  2072                                 ;     CHANGE THE ARGUMENT OF "DISPATCH" TO BE THE
  2073                                 ;     NUMBER OF HARDWARE TESTS IN YOUR PROGRAM.
  2074                                 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  2075
  2076
  2077
  2078
  2079
```

```
CZDMSF.P11    30-SEP-81 15:40          DEFAULT HARDWARE P-TABLE

                                       .SBTTL  DEFAULT HARDWARE P-TABLE

                                       ;//////////////////////////////////////////////////////////////////////////////
                                       ;/ THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
                                       ;/ THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
                                       ;/ IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
                                       ;//////////////////////////////////////////////////////////////////////////////

002224                                         BGNHW   DFPTBL
002224  000006                                                                .WORD   L10000-L$HW/2
002226                                                                  L$HW::
002226                                                                  DFPTBL::

002226  160170                                 .WORD   160170          ;M8207 CSR UNIBUS ADDRESS
002230  000000                                 .WORD      000          ;M8203 REG 11 (E134 SW10,9 , E121 SW9,10)
002232  000000                                 .WORD      000          ;M8203 REG 15 (E134 SW8-1)
002234  000000                                 .WORD      000          ;M8203 REG 16 (E121 SW8-1)
002236  000000                                 .WORD        0          ;H3254&H3255 OR OTHER LOOPBACK USED
002240  000004                                 .WORD        4          ;BAUD RATE = 56 K

002242                                         ENDHW
002242                                                                  L10000:
```

```
CZDMSF.P11    30-SEP-81 15:40          SOFTWARE P-TABLE

                                       .SBTTL  SOFTWARE P-TABLE

                                       ;////////////////////////////////////////////////////////////////////
                                       ;/ THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
                                       ;/ PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
                                       ;////////////////////////////////////////////////////////////////////

002242                                         BGNSW   SFPTBL
002242  000004                                                                .WORD   L10001-L$SW/2
002244                                                                 L$SW::
002244                                                                 SFPTBL::

002244  000000                         MIFLAG: .WORD   0       ; =1 IF MAN. INTERVENTION DESIRED, =0 IF NOT
002246  000000                         PRNFLG: .WORD   0       ; =1 IF SW PACK AND AX3-15 PRINTOUT ALLOWED ALWAYS
002250  000000                         SWIFLG: .WORD   0       ; =1 IF SWITCH PACK VERIFICATION TEST SHOULD BE RUN
002252  000000                         TCOUNT: .WORD   0       ;INITIAL MSG TIME-OUT VALUE (0=LONGEST TIME-OUT)

002254                                         ENDSW
002254                                                                 L10001:
```

```
                                .SBTTL  GLOBAL EQUATES SECTION

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

002254                                  EQUALS
                                ;
                                ; BIT DIFINITIONS
                                ;
100000                          BIT15== 100000
040000                          BIT14== 40000
020000                          BIT13== 20000
010000                          BIT12== 10000
004000                          BIT11== 4000
002000                          BIT10== 2000
001000                          BIT09== 1000
000400                          BIT08== 400
000200                          BIT07== 200
000100                          BIT06== 100
000040                          BIT05== 40
000020                          BIT04== 20
000010                          BIT03== 10
000004                          BIT02== 4
000002                          BIT01== 2
000001                          BIT00== 1
                                ;
001000                          BIT9==  BIT09
000400                          BIT8==  BIT08
000200                          BIT7==  BIT07
000100                          BIT6==  BIT06
000040                          BIT5==  BIT05
000020                          BIT4==  BIT04
000010                          BIT3==  BIT03
000004                          BIT2==  BIT02
000002                          BIT1==  BIT01
000001                          BIT0==  BIT00
                                ;
                                ; EVENT FLAG DEFINITIONS
                                ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                ;
000040                          EF.START==      32.                             ; START COMMAND WAS ISSUED
000037                          EF.RESTART==    31.                             ; RESTART COMMAND WAS ISSUED
```

```
CZDMSF.P11    30-SEP-81 15:40          GLOBAL EQUATES SECTION

  2188          000036                EF.CONTINUE==   30.                    ; CONTINUE COMMAND WAS ISSUED
  2189          000035                EF.NEW==        29.                    ; A NEW PASS HAS BEEN STARTED
  2190          000034                EF.PWR==        28.                    ; A POWER-FAIL/POWER-UP OCCURRED
  2191                                ;
  2192                                ;
  2193                                ; PRIORITY LEVEL DEFINITIONS
  2194                                ;
  2195          000340                PRI07== 340
  2196          000300                PRI06== 300
  2197          000240                PRI05== 240
  2198          000200                PRI04== 200
  2199          000140                PRI03== 140
  2200          000100                PRI02== 100
  2201          000040                PRI01== 40
  2202          000000                PRI00== 0
  2203                                ;
  2204                                ;OPERATOR FLAG BITS
  2205                                ;
  2206          000004                EVL==       4
  2207          000010                LOT==      10
  2208          000020                ADR==      20
  2209          000040                IDU==      40
  2210          000100                ISR==     100
  2211          000200                UAM==     200
  2212          000400                BOE==     400
  2213          001000                PNT==    1000
  2214          002000                PRI==    2000
  2215          004000                IXE==    4000
  2216          010000                IBE==   10000
  2217          020000                IER==   20000
  2218          040000                LOE==   40000
  2219          100000                HOE==  100000
  2220
  2221
  2222
  2223
  2224                                ;;*****************************************************************************
  2225                                ;* PROGRAM EVENT FLAG DEFINITIONS
  2226                                ;;*****************************************************************************
  2227
  2228
  2229
  2230
  2231
  2232                                ;;*****************************************************************************
  2233                                ;* MAINTENANCE REGISTER - BSEL1
  2234                                ;;*****************************************************************************
  2235          000200                RUN     = BIT7
  2236          000100                MCLR    = BIT6
  2237          000020                STEPLU  = BIT4
  2238          000010                LULOOP  = BIT3
  2239          000004                ROMO    = BIT2
  2240          000002                ROMI    = BIT1
  2241          000001                STEPMP  = BIT0
  2242
  2243                                ;;*****************************************************************************
```

```
CZDMSF.P11    30-SEP-81 15:40          GLOBAL EQUATES SECTION

  2244                                 ;* OBUS REG 10 - TRANSMITTER BUFFER
  2245                                 ;;*****************************************************************************
  2246          000200                 TX7      = BIT7
  2247          000100                 TX6      = BIT6
  2248          000040                 TX5      = BIT5
  2249          000020                 TX4      = BIT4
  2250          000010                 TX3      = BIT3
  2251          000004                 TX2      = BIT2
  2252          000002                 TX1      = BIT1
  2253          000001                 TX0      = BIT0
  2254
  2255                                 ;;*****************************************************************************
  2256                                 ;* OBUS REG 11
  2257                                 ;;*****************************************************************************
  2258          000200                 OC       = BIT7
  2259          000010                 GOAH     = BIT3
  2260          000004                 ABORT    = BIT2
  2261          000002                 EOM      = BIT1
  2262          000001                 SOM      = BIT0
  2263
  2264                                 ;;*****************************************************************************
  2265                                 ;* OBUS REG 12
  2266                                 ;;*****************************************************************************
  2267          000200                 IC       = BIT7
  2268          000100                 BPOLL    = BIT6
  2269          000040                 LULP     = BIT5
  2270
  2271                                 ;;*****************************************************************************
  2272                                 ;* OBUS REG 13
  2273                                 ;;*****************************************************************************
  2274          000200                 POLL     = BIT7
  2275          000100                 DTR      = BIT6
  2276          000040                 SELFR    = BIT5
  2277          000020                 HDX      = BIT4
  2278          000010                 MAINT1   = BIT3
  2279          000004                 MAINT2   = BIT2
  2280          000002                 SELSBY   = BIT1
  2281
  2282                                 ;;*****************************************************************************
  2283                                 ;* OBUS REG 14
  2284                                 ;;*****************************************************************************
  2285          000100                 TXEN     = BIT6
  2286          000040                 DISSI    = BIT5
  2287          000020                 RDAX     = BIT4
  2288          000010                 WAX      = BIT3
  2289          000004                 ENAX     = BIT2
  2290          000002                 AX2      = BIT1
  2291          000001                 AX1      = BIT0
  2292
  2293                                 ;;*****************************************************************************
  2294                                 ;* OBUS REG 17
  2295                                 ;;*****************************************************************************
  2296          000200                 CRC2     = BIT7
  2297          000100                 CRC1     = BIT6
  2298          000040                 IDLE     = BIT5
  2299          000020                 SECA     = BIT4
```

CZDMSF.P11 30-SEP-81 15:40 GLOBAL EQUATES SECTION

```
2300            000010          STRIP   = BIT3
2301            000004          RDALL   = BIT2
2302            000002          IERR    = BIT1
2303            000001          DDCMP   = BIT0
2304
2305                            ;;*****************************************************************************
2306                            ;* IBUS REG 10 - RECEIVER BUFFER
2307                            ;;*****************************************************************************
2308            000200          RX7     = BIT7
2309            000100          RX6     = BIT6
2310            000040          RX5     = BIT5
2311            000020          RX4     = BIT4
2312            000010          RX3     = BIT3
2313            000004          RX2     = BIT2
2314            000002          RX1     = BIT1
2315            000001          RX0     = BIT0
2316
2317                            ;;*****************************************************************************
2318                            ;* IBUS REG 11
2319                            ;;*****************************************************************************
2320            000200          OC      = BIT7
2321            000100          OACT    = BIT6
2322            000040          SW3     = BIT5
2323            000020          ORDY    = BIT4
2324            000010          SW2     = BIT3
2325            000004          SW1     = BIT2
2326            000002          SW0     = BIT1
2327            000001          UNRR    = BIT0
2328
2329                            ;;*****************************************************************************
2330                            ;* IBUS REG 12
2331                            ;;*****************************************************************************
2332            000200          IC      = BIT7
2333            000100          IACT    = BIT6
2334            000040          LULP    = BIT5
2335            000020          IRDY    = BIT4
2336            000010          OVRR    = BIT3
2337            000004          RAB     = BIT2
2338            000002          EBLK    = BIT1
2339            000001          BCC     = BIT0
2340
2341                            ;;*****************************************************************************
2342                            ;* IBUS REG 13
2343                            ;;*****************************************************************************
2344            000200          RING    = BIT7
2345            000100          DTR     = BIT6
2346            000040          RTS     = BIT5
2347            000020          HDX     = BIT4
2348            000010          MODR    = BIT3
2349            000004          CS      = BIT2
2350            000002          STBY    = BIT1
2351            000001          CARR    = BIT0
2352
2353                            ;;*****************************************************************************
2354                            ;* IBUS REG 14
2355                            ;;*****************************************************************************
```

```
CZDMSF.P11    30-SEP-81 15:40          GLOBAL EQUATES SECTION

  2356          000200                READY   = BIT7
  2357          000100                TXEN    = BIT6
  2358          000040                DISSI   = BIT5
  2359          000020                RDAX    = BIT4
  2360          000010                WAX     = BIT3
  2361          000004                ENAX    = BIT2
  2362          000002                AX2     = BIT1
  2363          000001                AX1     = BIT0
  2364
  2365                                ;;**************************************************************************************
  2366                                ;* IBUS REG 17
  2367                                ;;**************************************************************************************
  2368          000200                SIGR    = BIT7
  2369          000100                SIGQ    = BIT6
  2370          000040                TXDATA  = BIT5
  2371          000020                OCOR    = BIT4
  2372          000010                ICIR    = BIT3
  2373          000004                TESTMD  = BIT2
  2374          000002                MCLK    = BIT1
  2375          000001                DDCMP   = BIT0
  2376
  2377                                ;;**************************************************************************************
  2378                                ;* AX0-15 - USYRT REG 0 (READ ONLY)
  2379                                ;;**************************************************************************************
  2380          000200                RX7     = BIT7
  2381          000100                RX6     = BIT6
  2382          000040                RX5     = BIT5
  2383          000020                RX4     = BIT4
  2384          000010                RX3     = BIT3
  2385          000004                RX2     = BIT2
  2386          000002                RX1     = BIT1
  2387          000001                RX0     = BIT0
  2388
  2389                                ;;**************************************************************************************
  2390                                ;* AX0-16 - USYRT REG 1 (READ ONLY)
  2391                                ;;**************************************************************************************
  2392          000200                RERR    = BIT7
  2393          000100                ASBC2   = BIT6
  2394          000040                ASBC1   = BIT5
  2395          000020                ASBC0   = BIT4
  2396          000010                ROR     = BIT3
  2397          000004                RABT    = BIT2
  2398          000002                REOM    = BIT1
  2399          000001                RSOM    = BIT0
  2400
  2401                                ;;**************************************************************************************
  2402                                ;* AX1-15 - USYRT REG 2
  2403                                ;;**************************************************************************************
  2404          000200                TX7     = BIT7
  2405          000100                TX6     = BIT6
  2406          000040                TX5     = BIT5
  2407          000020                TX4     = BIT4
  2408          000010                TX3     = BIT3
  2409          000004                TX2     = BIT2
  2410          000002                TX1     = BIT1
  2411          000001                TX0     = BIT0
```

```
2412
2413                            ;;*****************************************************************************
2414                            ;* AX1-16 - USYRT REG 3
2415                            ;;*****************************************************************************
2416    000200                  TERR    = BIT7
2417    000010                  TXGA    = BIT3
2418    000004                  TXAB    = BIT2
2419    000002                  TEOM    = BIT1
2420    000001                  TSOM    = BIT0
2421
2422                            ;;*****************************************************************************
2423                            ;* AX2-15 - USYRT REG 4
2424                            ;;*****************************************************************************
2425    000200                  SYN7    = BIT7
2426    000100                  SYN6    = BIT6
2427    000040                  SYN5    = BIT5
2428    000020                  SYN4    = BIT4
2429    000010                  SYN3    = BIT3
2430    000004                  SYN2    = BIT2
2431    000002                  SYN1    = BIT1
2432    000001                  SYN0    = BIT0
2433    000226                  SYNCH   = 226
2434
2435                            ;;*****************************************************************************
2436                            ;* AX2-16 - USYRT REG 5
2437                            ;;*****************************************************************************
2438    000200                  APA     = BIT7
2439    000100                  DDC     = BIT6
2440    000040                  STR     = BIT5
2441    000020                  SEC     = BIT4
2442    000010                  IDL     = BIT3
2443    000004                  CRCTY2  = BIT2
2444    000002                  CRCTY1  = BIT1
2445    000001                  CRCTY0  = BIT0
2446
2447                            ;;*****************************************************************************
2448                            ;* AX3-15 - USYRT REG 6
2449                            ;;*****************************************************************************
2450    000200                  I422    = BIT7
2451    000100                  XYZ     = BIT6
2452    000040                  C32BCC  = BIT5
2453    000020                  V35     = BIT4
2454    000010                  INTGRL  = BIT3
2455    000004                  C32ENB  = BIT2
2456    000002                  OP      = BIT1
2457    000001                  TEST    = BIT0
2458    000372                  AX315U  = I422!XYZ!C32BCC!V35!INTGRL!OP
2459
2460                            ;;*****************************************************************************
2461                            ;* AX3-16 - USYRT REG 7
2462                            ;;*****************************************************************************
2463    000200                  TXLEN2  = BIT7
2464    000100                  TXLEN1  = BIT6
2465    000040                  TXLEN0  = BIT5
2466    000004                  RXLEN2  = BIT2
2467    000002                  RXLEN1  = BIT1
```

```
CZDMSF.P11    30-SEP-81 15:40          GLOBAL EQUATES SECTION

  2468          000001                RXLEN0  = BIT0
  2469
  2470
  2471
  2472
  2473
  2474                                ;*****************************************************************************
  2475                                ;* TX CONTROL BITS DEFINED ON WORD BASIS
  2476                                ;*****************************************************************************
  2477          004000                TXGOA   = BIT11
  2478          002000                TXABT   = BIT10
  2479          001000                TXEOM   = BIT9
  2480          000400                TXSOM   = BIT8
  2481
  2482
  2483
  2484
  2485
  2486                                ;*****************************************************************
  2487                                ;* RCV CONTROL BITS DEFINED ON WORD BASIS
  2488                                ;*****************************************************************
  2489          004000                RXOVR   = BIT11
  2490          002000                RXABT   = BIT10
  2491          001000                RXEBL   = BIT9
  2492          000400                RXBCC   = BIT8
  2493
  2494
  2495
  2496
  2497                                ;*****************************************************************************
  2498                                ;* ADDRESS EQUATES FOR REGISTER STORAGE TABLE (LUREG:)
  2499                                ;*****************************************************************************
  2500          002254                LUR10   = LUREG+0       ;LINE UNIT IBUS REG 10
  2501          002256                LUR11   = LUREG+2       ;LINE UNIT IBUS REG 11
  2502          002260                LUR12   = LUREG+4       ;LINE UNIT IBUS REG 12
  2503          002262                LUR13   = LUREG+6       ;LINE UNIT IBUS REG 13
  2504          002264                LUR14   = LUREG+10      ;LINE UNIT IBUS REG 14
  2505          002266                LUR15   = LUREG+12      ;LINE UNIT IBUS REG 15
  2506          002270                LUR16   = LUREG+14      ;LINE UNIT IBUS REG 16
  2507          002272                LUR17   = LUREG+16      ;LINE UNIT IBUS REG 17
  2508          002274                AX0.15  = LUREG+20      ;USYRT REG 0
  2509          002276                AX0.16  = LUREG+22      ;USYRT REG 1
  2510          002300                AX1.15  = LUREG+24      ;USYRT REG 2
  2511          002302                AX1.16  = LUREG+26      ;USYRT REG 3
  2512          002304                AX2.15  = LUREG+30      ;USYRT REG 4
  2513          002306                AX2.16  = LUREG+32      ;USYRT REG 5
  2514          002310                AX3.15  = LUREG+34      ;USYRT REG 6
  2515          002312                AX3.16  = LUREG+36      ;USYRT REG 7
  2516
  2517
  2518
  2519
  2520
  2521          100000                CHPCHK  = BIT15
  2522
  2523          100000                BCCCHK  = BIT15
```

```
CZDMSF.P11    30-SEP-81 15:40          GLOBAL EQUATES SECTION

  2524          100000                CRCCHK  = BIT15
  2525
  2526          100000                TCCHEK  = BIT15
  2527
  2528          000000                H3254   = 0
  2529          000001                H325    = 1
  2530          000002                H3250   = 2
  2531          000003                H3251   = 3
  2532          000004                INTMDM  = 4
  2533
  2534          000001                CBLPBK  = 1
  2535
  2536
  2537
  2538
  2539
  2540                                ;*************************************************************************
  2541                                ;* MICROINSTRUCTION DEFINITIONS
  2542                                ;*************************************************************************
  2543          021000                MVIOX   = 021000                ;MOVE IBUS TO OBUS*
  2544          122000                MVIXO   = 122000                ;MOVE IBUS* TO OBUS
  2545
  2546
  2547
  2548                                ;***** ERROR1 BIT FLAG DEFINITIONS *****
  2549          000001                RRDYTO  = BIT0
  2550          000002                WRDYTO  = BIT1
  2551
  2552
  2553
  2554
  2555
```

```
                                .SBTTL  GLOBAL DATA SECTION

                                ;////////////////////////////////////////////////////////////////////////////////
                                ;/      THE GLOBAL DATA SECTION C  TAINS DATA THAT ARE USED
                                ;/      IN MORE THAN ONE TEST.
                                ;////////////////////////////////////////////////////////////////////////////////

                                ;;*****************************************************************************
                                ;* STORAGE FOR DEVICE REGISTERS
                                ;;*****************************************************************************
002254  000020                  LUREG:  .BLKW   16.

                                ;*****************************************************************************
                                ;* MISCELLANEOUS STORAGE
                                ;*****************************************************************************
002314  000000                  SCRACH: .WORD   0               ;GEN'L PURPOSE SCRATCH WORD
002316  000000                  LOGDEV: .WORD   0               ;LOGICAL DEVICE NUMBER
002320  000000                  PSTACK: .WORD   0               ;CONTAINS BASE LEVEL PROGRAM STACK POINTER
002322  000000                  PRIOR:  .WORD   0               ;CPU PRIORITY FOR PRINTOUT
002324  000000                  SUBRPC: .WORD   0               ;PC OF SUBR CALL FOR ERROR REPORTS
002326  000000                  INTFLG: .WORD   0               ;INTERRUPT RECEIVED FLAGS
                                                                ;  BIT 0 FOR TX, BIT 1 FOR RCV
002330  000000                  ERRFLG: .WORD   0               ;SUBROUTINE ERROR FLAG
002332  000000                  TIMFLG: .WORD   0               ;EVENT TIME-OUT FLAG
002334  000000                  RETADR: .WORD   0               ;SUBR ERROR RETURN ADDRESS
002336  000000                  REDBYT: .WORD   0               ;LO BYTE CONTAINS BYTE READ FROM LU REG
002340  000000                  WRIBYT: .WORD   0               ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
002342  000000                  RAX15:  .WORD   0               ;LO BYTE CONTAINS BYTE READ FROM REG 15
002344  000000                  RAX16:  .WORD   0               ;LO BYTE CONTAINS BYTE READ FROM REG 16
002346  000000                  WAX15:  .WORD   0               ;LO BYTE CONTAINS BYTE TO LOAD INTO REG 15
002350  000000                  WAX16:  .WORD   0               ;LO BYTE CONTAINS BYTE TO LOAD INTO REG 16
002352  000000                  REGNUM: .WORD   0               ;NUMBER (10-17) OF LINE UNIT REG BEING TESTED
002354  000000                  AXNUM:  .WORD   0               ;NUMBER (0-7) OF EXTENDED REG BYTE BEING TESTED
002356  000000                  GOODAT: .WORD   0               ;STORAGE FOR EXPECTED DATA
002360  000000                  BADDAT: .WORD   0               ;STORAGE FOR ACTUAL DATA
002362  000000                  LOADAT: .WORD   0               ;CONTAINS TEST DATA LOADED INTO REG
002364  000000                  FRSTIM: .WORD   0               ;FLAG=0 IF PROGRAM JUST LOADED
002366  000000                  FRSPAS: .WORD   0               ;FLAG=0 IF FIRST PASS AFTER LOAD
002370  000000                  STARES: .WORD   0               ;FLAG=0 IF FIRST TIME THRU AFTER STA OR RES
002372  000000                  SAVE4:  .WORD   0               ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
002374  000000                  SAVE6:  .WORD   0               ;SAVE LOC 6 HERE (ERROR TRAP VECTOR)
002376  000000                  ERROR1: .WORD   0               ;SUBR ERROR BIT FLAGS (DEF'D IN GLOBAL EQUATES)
002400  000000                  TXWORD: .WORD   0               ;BITS 0-11 CONTAIN DATA TO LOAD INTO TX SILO
002402  000000                  RXWORD: .WORD   0               ;BITS 0-11 CONTAIN DATA READ FROM RCV SILO
002404  000000                  DISILO: .WORD   0               ;CONTAINS CURRENT STATE OF DISSI IN BIT 5
002406  000000                  CHPTYP: .WORD   0               ;USYRT CHIP TYPE, =0 FOR SIG, ELSE =1
002410  000000                  MODINT: .WORD   0               ;MODEM INTERFACE SELECTION
002412  000000                  SAVLEN: .WORD   0               ;SAVED TX AND RCV CHAR LENGTHS
002414  000000                  DEVMAP: .WORD   0               ;BIT MAP OF ACTIVE DEVICES
002416  000000                  DEVPTR: .WORD   0               ;DEVICE MAP BIT POINTER
002420  000000                  UNIT:   .WORD   0               ;CONTAINS UNIT NUMBER (1 TO N)
002422  000000                  TSTNUM: .WORD   0               ;CONTAINS TEST NUMBER FOR SOME TESTS

                                ;;*********  CURRENT DEVICE PARAMETERS  *************
002424  160170                  MPCSR:  .WORD   160170          ;POINTER TO MICROPROCESSOR CSR'S
002426  160171                  BSEL1:  .WORD   160171          ;POINTER TO BSEL1
```

CZDMSF.P11 30-SEP-81 15:40 GLOBAL DATA SECTION

```
2612  002430                  BSEL4:
2613  002430  160174          SEL4:    .WORD   160174          ;POINTER TO SEL4
2614  002432  160176          SEL6:    .WORD   160176          ;POINTER TO SEL6
2615  002434  000000          LUSWI1:  .WORD        0          ;LINE UNIT SWITCH PACK #1
2616  002436  000000          LUSWI2:  .WORD        0          ;LINE UNIT SWITCH PACK #2
2617  002440  000000          LUSWI3:  .WORD        0          ;LINE UNIT SWITCH PACK #3
2618  002442  000000          TSTCON:  .WORD        0          ;TEST CONNECTOR INDICATOR
2619  002444  000000          LPBCON:  .WORD        0          ;LOOP BACK CONNECTOR INDICATOR
2620  002446  000004          BDRATE:  .WORD        4          ;BAUD RATE
2621  002450  000000          MLWBYT:  .WORD        0          ;INITIAL WRIBYT FOR REG 13 MAINT1/2 BITS
2622
2623                          ;***** STORAGE FOR DATA READ IN ADDRESS TESTS *****
2624  002452      000         REDDAT:  .BYTE   0
2625  002453      000                  .BYTE   0
2626  002454      000                  .BYTE   0
2627  002455      000                  .BYTE   0
2628  002456      000                  .BYTE   0
2629  002457      000                  .BYTE   0
2630  002460      000                  .BYTE   0
2631  002461      000                  .BYTE   0
2632
2633                          ;;****  GEN'L PURPOSE SCRATCH STORAGE  *****
2634  002462  000000          REG0:    .WORD   0
2635  002464  000000          REG1:    .WORD   0
2636  002466  000000          REG2:    .WORD   0
2637  002470  000000          REG3:    .WORD   0
2638  002472  000000          REG4:    .WORD   0
2639  002474  000000          REG5:    .WORD   0
2640  002476  000000          REG6:    .WORD   0
2641  002500  000000          REG7:    .WORD   0
2642
2643                          ;;****  SCRATCH STORAGE FOR MESSAGE REPORTING   *****
2644  002502  000000          TMP0:    .WORD   0
2645  002504  000000          TMP1:    .WORD   0
2646  002506  000000          TMP2:    .WORD   0
2647  002510  000000          TMP3:    .WORD   0
2648  002512  000000          TMP4:    .WORD   0
2649  002514  000000          TMP5:    .WORD   0
2650  002516  000000          TMP6:    .WORD   0
2651  002520  000000          TMP7:    .WORD   0
2652
2653                          ;***** INBUS LU REG BIT MASKS FOR UNPREDICTABLE BITS *****
2654  002522                  UPBITS:
2655  002522      000                  .BYTE   000             ;MASK FOR REG 10
2656  002523      056                  .BYTE   056             ;MASK FOR REG 11
2657  002524      000                  .BYTE   000             ;MASK FOR REG 12
2658  002525      257                  .BYTE   257             ;MASK FOR REG 13
2659  002526      100                  .BYTE   100             ;MASK FOR REG 14
2660  002527      377                  .BYTE   377             ;MASK FOR REG 15
2661  002530      377                  .BYTE   377             ;MASK FOR REG 16
2662  002531      306                  .BYTE   306             ;MASK FOR REG 17
2663
2664  002532      200         R14NRW:  .BYTE   200             ;REG 14 NON-R/W BITS
2665
2666                          ;***** MASKS FOR EXTENDED REGISTER NON-READ/WRITE BITS *****
2667  002533                  ANBITS:
```

CZDMSF.P11 30-SEP-81 15:40 GLOBAL DATA SECTION

```
2668  002533  377                .BYTE   377             ;MASK FOR AX0-15
2669  002534  377                .BYTE   377             ;MASK FOR AX0-16
2670  002535  000                .BYTE   000             ;MASK FOR AX1-15
2671  002536  360                .BYTE   360             ;MASK FOR AX1-16
2672  002537  000                .BYTE   000             ;MASK FOR AX2-15
2673  002540  000                .BYTE   000             ;MASK FOR AX2-16
2674  002541  004                .BYTE   004             ;MASK FOR AX3-15
2675  002542  030                .BYTE   030             ;MASK FOR AX3-16
2676
2677                             ;***** DATA PATTERN A *****
2678  002543                     PATA:
2679  002543  125                .BYTE   125
2680  002544  252                .BYTE   252
2681  002545  000                .BYTE   000
2682  002546  377                .BYTE   377
2683  002547  001                .BYTE   001
2684  002550  002                .BYTE   002
2685  002551  004                .BYTE   004
2686  002552  010                .BYTE   010
2687  002553  020                .BYTE   020
2688  002554  040                .BYTE   040
2689  002555  100                .BYTE   100
2690  002556  200                .BYTE   200
2691  002557  376                .BYTE   376
2692  002560  375                .BYTE   375
2693  002561  373                .BYTE   373
2694  002562  367                .BYTE   367
2695  002563  357                .BYTE   357
2696  002564  337                .BYTE   337
2697  002565  277                .BYTE   277
2698  002566  177                .BYTE   177
2699
2700                             ;***** DATA PATTERN B *****
2701  002567                     PATB:
2702  002567  000                .BYTE   000
2703  002570  000                .BYTE   000
2704  002571  040                .BYTE   040
2705  002572  100                .BYTE   100
2706  002573  220                .BYTE   220
2707  002574  000                .BYTE   000
2708  002575  000                .BYTE   000
2709  002576  051                .BYTE   051
2710
2711
2712                             ;***** DATA PATTERN Q *****
2713  002577  000                PATQ:   .BYTE   000
2714  002600  120                .BYTE   120
2715  002601  125                .BYTE   125
2716  002602  137                .BYTE   137
2717  002603  040                .BYTE   040
2718  002604  052                .BYTE   052
2719  002605  057                .BYTE   057
2720  002606  177                .BYTE   177
2721
2722                             ;***** DATA PATTERN R *****
2723  002607  000                PATR:   .BYTE   000
```

```
CZDMSF.P11    30-SEP-81 15:40          GLOBAL DATA SECTION

  2724  002610      100                        .BYTE   100
  2725  002611      120                        .BYTE   120
  2726  002612      124                        .BYTE   124
  2727  002613      164                        .BYTE   164
  2728  002614      172                        .BYTE   172
  2729  002615      176                        .BYTE   176
  2730  002616      177                        .BYTE   177
  2731  002617      000                        .BYTE   000
  2732  002620      100                        .BYTE   100
  2733  002621      120                        .BYTE   120
  2734  002622      124                        .BYTE   124
  2735  002623      164                        .BYTE   164
  2736  002624      172                        .BYTE   172
  2737  002625      176                        .BYTE   176
  2738
  2739                                   ;***** DATA PATTERN S *****
  2740  002626      000                  PATS:   .BYTE   000
  2741  002627      017                          .BYTE   017
  2742  002630      036                          .BYTE   036
  2743  002631      074                          .BYTE   074
  2744  002632      170                          .BYTE   170
  2745  002633      360                          .BYTE   360
  2746  002634      037                          .BYTE   037
  2747  002635      076                          .BYTE   076
  2748  002636      174                          .BYTE   174
  2749  002637      370                          .BYTE   370
  2750  002640      077                          .BYTE   077
  2751  002641      176                          .BYTE   176
  2752  002642      374                          .BYTE   374
  2753  002643      177                          .BYTE   177
  2754  002644      376                          .BYTE   376
  2755  002645      377                          .BYTE   377
  2756
  2757                                   ;***** DATA PATTERN T *****
  2758  002646      000                  PATT:   .BYTE   000
  2759  002647      125                          .BYTE   125
  2760  002650      252                          .BYTE   252
  2761  002651      176                          .BYTE   176
  2762  002652      177                          .BYTE   177
  2763
  2764  002653                           ENDPAT:
  2765              002654               .EVEN
  2766
  2767
  2768
  2769
  2770
  2771                                   ;***  TEST MESSAGES TO BE TRANSMITTED  ***
  2772
  2773  002654  000400                   MSG1:   TXSOM
  2774  002656  000400                           TXSOM
  2775  002660  000000                           000
  2776  002662  000125                           125
  2777  002664  000252                           252
  2778  002666  000377                           377
  2779  002670  000000                           000
```

```
 2780  002672  001000                        TXEOM
 2781  002674  001000                        TXEOM
 2782  002676  001000                        TXEOM
 2783  002700  001000                        TXEOM
 2784
 2785  002702  000400              MSG2:     TXSOM
 2786  002704  000400                        TXSOM
 2787  002706  000000                        000
 2788  002710  000377                        377
 2789  002712  001000                        TXEOM
 2790  002714  001000                        TXEOM
 2791
 2792  002716  000001              MSG3:     001
 2793  002720  000001                        001
 2794  002722  000001                        001
 2795  002724  000001                        001
 2796  002726  002000                        TXABT
 2797  002730  000400                        TXSOM
 2798  002732  000400                        TXSOM
 2799  002734  000003                        003
 2800  002736  000003                        003
 2801  002740  000003                        003
 2802  002742  000003                        003
 2803  002744  000003                        003
 2804
 2805
 2806
 2807
 2808
 2809                              ;*** RECEIVED DATA BUFFER (64. WORDS) ***
 2810  002746  000100              RCVBUF: .BLKW   64.
 2811
 2812
 2813
 2814
 2815
 2816
 2817
 2818
 2819
 2820
```

```
                                        .SBTTL  GLOBAL TEXT SECTION

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

                                        ;;*****************************************************************************
                                        ;* NAMES OF DEVICES SUPPORTED BY PROGRAM
                                        ;;*****************************************************************************
003146                                          DEVTYP  <M8203>
003146                                                                          L$DVTYP::
003146  034115  030062  000063                                                          .ASCIZ  /M8203/
                                                                                        .EVEN

                                        ;;*******************************************************************************
                                        ;* TITLE OF PROGRAM
                                        ;;*******************************************************************************
003154                                          DESCRIPT        <M8203 STATIC LOGIC TESTS - PART 2 OF 2>
003154                                                                          L$DESC::
003154  034115  030062  020063                                                          .ASCIZ  /M8203 STATIC LO
003162  052123  052101  041511
003170  046040  043517  041511
003176  052040  051505  051524
003204  026440  050040  051101
003212  020124  020062  043117
003220  031040     000
        003224                                                                          .EVEN



                                        ;
                                        ; FORMAT STATEMENTS USED IN PRINT CALLS
                                        ;

                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                        ;     INSERT THE FORMAT STATEMENTS USED IN THE VARIOUS PRINT CALLS.
                                        ;     USE THE .ASCIZ STATEMENT.
                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
                                .SBTTL  GLOBAL SUBROUTINES

                                ;////////////////////////////////////////////////////////////////////////////////
                                ;/      THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
                                ;////////////////////////////////////////////////////////////////////////////////



                                ;*******************************************************************************
                                ;* STPCLK - THIS SUBROUTINE FORCES THE DMC11 OR KMC11 MICROPROCESSOR TO
                                ;*      EXECUTE AN INSTRUCTION WHICH IS PASSED IN THE WORD FOLLOWING THE CALL.
                                ;*******************************************************************************
003224                          STPCLK:
003224  152777  000006  177174          BISB    #ROMO!ROMI,@BSEL1       ;SET ROMO, ROMI BITS IN BSEL1
003232  017677  000000  177172          MOV     @(SP),@SEL6     ;PUT INSTRUCTION INTO SEL6
003240  152777  000007  177160          BISB    #ROMO!ROMI!STEPMP,@BSEL1 ;SET ROMO, ROMI, STEPMP IN BSEL1
003246  142777  000007  177152          BICB    #ROMO!ROMI!STEPMP,@BSEL1 ;CLEAR ROMO, ROMI, STEPMP IN BSEL1
003254  062716  000002                  ADD     #2,(SP)         ;FIX UP RETURN PC
003260  000207                          RTS     PC              ;RETURN



                                ;*******************************************************************************
                                ;* MSTCLR - THIS SUBROUTINE ISSUES A MASTER CLEAR AND SETS LULOOP
                                ;*******************************************************************************
003262                          MSTCLR:
003262  010146                          MOV     R1,-(SP)        ;SAVE R1
003264  013746  002352                  MOV     REGNUM,-(SP)    ;SAVE LU REG. NO.
003270  112777  000100  177130          MOVB    #MCLR,@BSEL1    ;SET MASTER CLEAR BIT
003276  142777  000300  177122          BICB    #RUN!MCLR,@BSEL1 ;CLEAR RUN AND MCLR BITS
003304  012701  000024                  MOV     #20.,R1         ;INITIALIZE STALL COUNTER
003310  000240                  2$:     NOP                     ;STALL IN LOOP FOR SEVERAL MICRO-SEC
003312  005301                          DEC     R1
003314  001375                          BNE     2$
003316  152777  000010  177102          BISB    #LULOOP,@BSEL1  ;SET LU LOOP
003324  012737  000013  002352          MOV     #13,REGNUM      ;SET LU REG NO. = 13
003332  113737  002450  002340          MOVB    MLWBYT,WRIBYT   ;MANAGE MAINT BITS FOR MODEM LOOPBACK
003340  004737  003436                  JSR     PC,WRITLU       ;CLEAR REG 13 (EXCEPT DTR & MAINT IF MOD LOOPBK)
003344  012637  002352                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO
003350  012601                          MOV     (SP)+,R1        ;RESTORE R1
003352  005037  002412                  CLR     SAVLEN          ;CLEAR SAVED CHAR LENGTH FROM SETUP
003356  000207                          RTS     PC              ;RETURN



                                ;*******************************************************************************
                                ;* READLU - THIS SUBROUTINE FORCES THE DMC11 OR KMC11 MICROPROCESSOR
                                ;*      TO EXECUTE AN INSTRUCTION WHICH READS THE LINE UNIT REG WHOSE
                                ;*      NUMBER IS PASSED IN REGNUM, INTO REDBYT.
                                ;*******************************************************************************
```

CZDMSF.P11 30-SEP-81 15:40 GLOBAL SUBROUTINES

```
2921  003360                          READLU:
2922  003360  010146                         MOV     R1,-(SP)        ;SAVE R1
2923  003362  013701  002352                 MOV     REGNUM,R1       ;GET LINE UNIT REG NUMBER
2924  003366  006301                         ASL     R1              ;SHIFT INTO SOURCE BITS 4-7
2925  003370  006301                         ASL     R1
2926  003372  006301                         ASL     R1
2927  003374  006301                         ASL     R1
2928  003376  052701  000004                 BIS     #4,R1           ;SET DESTINATION = BSEL4
2929  003402  052701  021000                 BIS     #MVIOX,R1       ;SET REST OF MOVE INSTRUCTION
2930  003406  010137  003416                 MOV     R1,2$           ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
2931  003412  004737  003224                 JSR     PC,STPCLK       ;EXECUTE MOVE INSTRUCTION
2932  003416  000000                  2$:    .WORD   0               ;INSTRUCTION GOES HERE
2933  003420  117737  177004  002336         MOVB    @BSEL4,REDBYT   ;GET LU REG CONTENTS INTO REDBYT
2934  003426  105037  002337                 CLRB    REDBYT+1        ;CLR HI BYTE OF STORAGE
2935  003432  012601                         MOV     (SP)+,R1        ;RESTORE R1
2936  003434  000207                         RTS     PC              ;RETURN
2937
2938
2939
2940
2941
2942                                  ;*********************************************************************************
2943                                  ;* WRITLU - THIS SUBROUTINE FORCES THE DMC11 OR KMC11 MICROPROCESSOR TO
2944                                  ;*      EXECUTE AN INSTRUCTION WHICH LOADS THE BYTE CONTAINED IN WRIBYT
2945                                  ;*      INTO THE LU REG WHOSE NUMBER IS PASSED IN REGNUM.
2946                                  ;*********************************************************************************
2947  003436                          WRITLU:
2948  003436  010146                         MOV     R1,-(SP)        ;SAVE R1
2949  003440  013701  002352                 MOV     REGNUM,R1       ;GET LINE UNIT REG NUMBER
2950  003444  052701  000100                 BIS     #100,R1         ;SET SOURCE = BSEL4
2951  003450  052701  122000                 BIS     #MVIXO,R1       ;SET REST OF MOVE INSTRUCTION
2952  003454  010137  003476                 MOV     R1,2$           ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
2953  003460  105037  002341                 CLRB    WRIBYT+1        ;CLR HI BYTE OF STORAGE
2954  003464  113777  002340  176736         MOVB    WRIBYT,@BSEL4   ;LOAD BYTE INTO BSEL4
2955  003472  004737  003224                 JSR     PC,STPCLK       ;EXECUTE MOVE INSTRUCTION
2956  003476  000000                  2$:    .WORD   0
2957  003500  012601                         MOV     (SP)+,R1        ;RESTORE R1
2958  003502  000207                         RTS     PC              ;RETURN
2959
2960
2961
2962
2963
2964                                  ;*********************************************************************************
2965                                  ;* GETREG - THIS SUBROUTINE READS THE LINE UNIT REGISTERS 10-17 INTO THE
2966                                  ;*      REGISTER STORAGE TABLE (LUREG:).
2967                                  ;*********************************************************************************
2968  003504  010146                  GETREG: MOV     R1,-(SP)        ;SAVE R1
2969  003506  013746  002352                 MOV     REGNUM,-(SP)    ;SAVE CURRENT REG NO.
2970  003512  012701  002254                 MOV     #LUR10,R1       ;INIT POINTER TO REG STORAGE TABLE
2971  003516  012737  000010  002352         MOV     #10,REGNUM      ;INIT LU REG NO. TO 10
2972  003524  004737  003360          3$:    JSR     PC,READLU       ;READ A LINE UNIT REG
2973  003530  113721  002336                 MOVB    REDBYT,(R1)+    ;PUT BYTE READ INTO TABLE
2974  003534  105021                         CLRB    (R1)+           ;CLEAR UPPER BYTE OF TABLE ENTRY
2975  003536  005237  002352                 INC     REGNUM          ;INCREMENT REG NO.
2976  003542  023727  002352  000020         CMP     REGNUM,#20      ;SEE IF ALL REGS READ YET
```

```
                        BLT     3$              ;BR IF NOT
                        MOV     (SP)+,REGNUM    ;RESTORE CURRENT REG NO.
                        MOV     (SP)+,R1        ;RESTORE R1
                        RTS     PC              ;RETURN
```

```
;*******************************************************************************
;* LOOPIN - THIS SUBROUTINE PLACES THE MICROPROCESSOR IN A LOOP ON AN
;*      INSTRUCTION, BY MOVING THE INSTRUCTION FROM THE WORD FOLLOWING THE CALL
;*      INTO SEL6, AND SETTING RUN AND ROMI IN BSEL1. THE SUBROUTINE RETURNS
;*      WITH THE MICROPROCESSOR STUCK IN THE LOOP, AND IF IT IS DESIRED TO
;*      TERMINATE THE LOOP, THE PDP-11 PROGRAM MUST CLEAR THE RUN BIT IN
;*      BSEL1, OR CALL SUBROUTINE MSTCLR TO DO THIS.
;*******************************************************************************
LOOPIN:
        BISB    #ROMO!ROMI,@BSEL1       ;SET ROMO, ROMI BITS IN BSEL1
        MOV     @(SP),@SEL6     ;PUT MICROINSTRUCTION INTO SEL6
        BISB    #RUN!ROMO!ROMI,@BSEL1 ;SET RUN, ROMO, ROMI IN BSEL1
        ADD     #2,(SP)         ;FIX UP RETURN PC
        RTS     PC              ;RETURN WITH MICROPROCESSOR STUCK IN SINGLE
                                ;    INSTRUCTION LOOP
```

```
;*******************************************************************************
;* READAX - THIS SUBROUTINE READS THE USYRT REG PAIR WHOSE NUMBER (0-3)
;*      IS PASSED IN BITS 1,2 OF AXNUM ON ENTRY, AND RETURNS THE BYTES READ IN
;*      RAX15 AND RAX16. IF THE LINE UNIT DOES NOT RESPOND WITH READY IN REG 14,
;*      RRDYTO BIT IS SET IN ERROR1 ON RETURN.
;*******************************************************************************
READAX: MOV     R1,-(SP)        ;SAVE R1
        MOV     REGNUM,-(SP)    ;STORE CURRENT REG NO.
        BIC     #RRDYTO,ERROR1  ;CLEAR ERROR BIT
        MOV     #14,REGNUM      ;SET LU REG NO. = 14
        MOVB    AXNUM,WRIBYT    ;SET UP AX REG NO. BITS
        ASR     WRIBYT
        BISB    #RDAX!ENAX,WRIBYT ;SET UP BITS TO LOAD INTO REG 14
        BIS     DISILO,WRIBYT   ;SET PROPER STATE OF DISSI BIT
        JSR     PC,WRITLU       ;SET RDAX AND ENAX IN REG 14
        CLR     R1              ;INIT TIMER
6$:     JSR     PC,READLU       ;READ REG 14
        BITB    #READY,REDBYT   ;SEE IF READY BIT SET IN REG 14 YET
        BNE     9$              ;BR IF READY SET
        INC     R1              ;INCR TIMER
        BNE     6$              ;BR IF TIMER DIDN'T TIME OUT YET
        BIS     #RRDYTO,ERROR1  ;SET ERROR FLAG FOR TIME OUT ON READ RDY
        BR      12$             ;BR TO RETURN
9$:     MOV     #15,REGNUM      ;SET REG NO. = 15
        JSR     PC,READLU       ;READ REG 15
        MOVB    REDBYT,RAX15    ;STORE REG AX-15
        CLRB    RAX15+1         ;CLR HI BYTE OF STORAGE
```

```
3033  003744  012737  000016  002352          MOV     #16,REGNUM      ;SET REG NO. = 16
3034  003752  004737  003360                  JSR     PC,READLU       ;READ REG 16
3035  003756  113737  002336  002344          MOVB    REDBYT,RAX16    ;STORE REG AX-16
3036  003764  105037  002345                  CLRB    RAX16+1         ;CLR HI BYTE OF STORAGE
3037  003770  012637  002352          12$:    MOV     (SP)+,REGNUM    ;RESTORE CURRENT REG NO.
3038  003774  012601                          MOV     (SP)+,R1        ;RESTORE R1
3039  003776  000207                          RTS     PC              ;RETURN
3040
3041
3042
3043
3044                                  ;*********************************************************************************
3045                                  ;* WRITAX - THIS SUBROUTINE WRITES THE USYRT REG PAIR WHOSE NUMBER (0-3) IS
3046                                  ;*       PASSED IN BITS 1,2 OF AXNUM ON ENTRY, WITH THE DATA FROM WAX15 AND
3047                                  ;*       WAX16. IF LINE UNIT DOES NOT RESPOND WITH READY IN REG 14, WRDYTO BIT
3048                                  ;*       IS SET IN ERROR1 ON RETURN.
3049                                  ;*********************************************************************************
3050  004000  010146          WRITAX: MOV     R1,-(SP)        ;SAVE R1
3051  004002  013746  002352          MOV     REGNUM,-(SP)    ;SAVE CURRENT REG NO.
3052  004006  042737  000002  002376          BIC     #WRDYTO,ERROR1  ;CLEAR ERROR BIT
3053  004014  012737  000014  002352          MOV     #14,REGNUM      ;SET LU REG NO. = 14
3054  004022  113737  002354  002340          MOVB    AXNUM,WRIBYT    ;SET AX REG NO. BITS
3055  004030  006237  002340                  ASR     WRIBYT
3056  004034  053737  002404  002340          BIS     DISILO,WRIBYT   ;SET PROPER STATE OF DISSI BIT
3057  004042  004737  003436                  JSR     PC,WRITLU       ;SET AX NO. BITS IN REG 14
3058  004046  012737  000015  002352          MOV     #15,REGNUM      ;SET REG NO. = 15
3059  004054  105037  002347                  CLRB    WAX15+1         ;CLR HI BYTE OF STORAGE
3060  004060  113737  002346  002340          MOVB    WAX15,WRIBYT    ;SET UP BYTE TO WRITE INTO REG 15
3061  004066  004737  003436                  JSR     PC,WRITLU       ;WRITE BYTE INTO REG 15
3062  004072  005237  002352                  INC     REGNUM          ;SET REG NO. = 16
3063  004076  105037  002351                  CLRB    WAX16+1         ;CLR HI BYTE OF STORAGE
3064  004102  113737  002350  002340          MOVB    WAX16,WRIBYT    ;SET UP BYTE TO WRITE INTO REG 16
3065  004110  004737  003436                  JSR     PC,WRITLU       ;WRITE BYTE INTO REG 16
3066  004114  012737  000014  002352          MOV     #14,REGNUM      ;SET REG NO. = 14
3067  004122  113737  002354  002340          MOVB    AXNUM,WRIBYT    ;SET AX REG NO. BITS
3068  004130  006237  002340                  ASR     WRIBYT
3069  004134  152737  000014  002340          BISB    #ENAX!WAX,WRIBYT ;SET UP BITS TO LOAD INTO REG 14
3070  004142  053737  002404  002340          BIS     DISILO,WRIBYT   ;SET PROPER STATE OF DISSI BIT
3071  004150  004737  003436                  JSR     PC,WRITLU       ;SET ENAX AND WAX IN REG 14
3072  004154  005001                          CLR     R1              ;INIT PROGRAM TIMER
3073  004156  004737  003360          6$:     JSR     PC,READLU       ;READ REG 14
3074  004162  132737  000200  002336          BITB    #READY,REDBYT   ;SEE IF READY BIT SET IN REG 14 YET
3075  004170  001005                          BNE     9$              ;BR IF READY SET
3076  004172  005201                          INC     R1              ;INCR TIMER
3077  004174  001370                          BNE     6$              ;BR IF TIMER DIDN'T TIME OUT YET
3078  004176  052737  000002  002376          BIS     #WRDYTO,ERROR1  ;SET ERROR FLAG BIT FOR TIME OUT ON WRITE RDY
3079  004204  012637  002352          9$:     MOV     (SP)+,REGNUM    ;RESTORE CURRENT REG NO.
3080  004210  012601                          MOV     (SP)+,R1        ;RESTORE R1
3081  004212  000207                          RTS     PC              ;RETURN
3082
3083
3084
3085
3086
3087                                  ;*********************************************************************************
3088                                  ;* GETALL - THIS SUBROUTINE READS THE LINE UNIT REGS 10-17 AND THE EXTENDED
```

```
3089                                    ;*      REGISTERS AX0-AX3 INTO REGISTER STORAGE TABLE (LUREG:).
3090                                    ;*************************************************************************
3091  004214  010146                    GETALL: MOV     R1,-(SP)        ;SAVE R1
3092  004216  013746  002354                    MOV     AXNUM,-(SP)     ;SAVE CURRENT AX REG BYTE NO.
3093  004222  012737  015310  002502            MOV     #DH5,TMP0       ;SET AX LO BYTE NO.
3094  004230  032737  000001  002354            BIT     #BIT0,AXNUM     ;SEE IF LO OR HI BYTE
3095  004236  001403                            BEQ     1$              ;BR IF LO BYTE
3096  004240  012737  015313  002502            MOV     #DH6,TMP0       ;SET AX HI BYTE NO.
3097  004246  004737  003504            1$:     JSR     PC,GETREG       ;READ AND STORE REGS 10-17
3098  004252  142777  000010  176146            BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
3099  004260  012701  002274                    MOV     #AX0,R1         ;INIT POINTER TO REG STORAGE TABLE
3100  004264  005037  002354                    CLR     AXNUM           ;INIT AX REG BYTE NO. TO 0
3101  004270  004737  003612            3$:     JSR     PC,READAX       ;READ 2 AX REG BYTES
3102  004274  113721  002342                    MOVB    RAX15,(R1)+     ;PUT LO BYTE READ INTO TABLE
3103  004300  105021                            CLRB    (R1)+           ;CLEAR UPPER BYTE OF TABLE ENTRY
3104  004302  113721  002344                    MOVB    RAX16,(R1)+     ;PUT HI BYTE READ INTO TABLE
3105  004306  105021                            CLRB    (R1)+           ;CLEAR UPPER BYTE OF TABLE ENTRY
3106  004310  062737  000002  002354            ADD     #2,AXNUM        ;INCR AX REG BYTE NO.
3107  004316  023727  002354  000010            CMP     AXNUM,#10       ;SEE IF ALL REGS READ YET
3108  004324  002761                            BLT     3$              ;BR IF NOT
3109  004326  012637  002354                    MOV     (SP)+,AXNUM     ;RESTORE CURRENT AX REG BYTE NO.
3110  004332  012601                            MOV     (SP)+,R1        ;RESTORE R1
3111  004334  013737  002354  002504            MOV     AXNUM,TMP1
3112  004342  006237  002504                    ASR     TMP1            ;GET EXTENDED REG NO. FOR PRINTOUT
3113  004346  000207                            RTS     PC              ;RETURN
3114
3115
3116
3117
3118
3119                                    ;*************************************************************************
3120                                    ;* OSIRDY - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF ORDY (REG 11)
3121                                    ;*      AND OCOR (REG 17) AND REPORTS AN ERROR IF EITHER IS NOT PROPERLY SET
3122                                    ;*      AS PASSED IN BIT 0 (ORDY) AND BIT 1 (OCOR) OF THE WORD FOLLOWING THE
3123                                    ;*      CALL.
3124                                    ;*      IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST, AT THE ADDRESS IN
3125                                    ;*      RETADR.
3126                                    ;*************************************************************************
3127  004350  013746  002352            OSIRDY: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
3128  004354  013746  002324                    MOV     SUBRPC,-(SP)
3129  004360  005737  002324                    TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
3130  004364  001006                            BNE     1$              ;BR IF YES
3131  004366  016637  000004  002324            MOV     4(SP),SUBRPC
3132  004374  162737  000004  002324            SUB     #4,SUBRPC       ;GET PC OF SUBROUTINE CALL
3133  004402  012737  000011  002352    1$:     MOV     #11,REGNUM      ;SET REG NO. TO 11
3134  004410  004737  003360                    JSR     PC,READLU       ;READ REG 11
3135  004414  032776  000001  000004            BIT     #BIT0,@4(SP)    ;GET EXPECTED STATE OF ORDY
3136  004422  001413                            BEQ     3$              ;BR IF EXPECTED ORDY = 0
3137  004424  132737  000020  002336            BITB    #ORDY,REDBYT    ;SEE IF ORDY = 1
3138  004432  001022                            BNE     9$              ;BR IF ORDY = 1
3139  004434  004737  004214                    JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3140                                  ;REPORT ORDY NOT SET
3141  004440                                    ERRDF   7,EM7,ERR4
3142  004440  104455                                                                    TRAP    C$ERDF
3143  004442  000007                                                                    .WORD   7
3144  004444  013537                                                                    .WORD   EM7
```

```
                                                                                .WORD   ERR4
                        BR      16$             ;TAKE ERROR RETURN
002336  3$:     BITB    #ORDY,REDBYT    ;SEE IF ORDY = 0
                BEQ     9$              ;BR IF ORDY = 0
004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
        ;REPORT ORDY NOT CLEARED
                ERRDF   8,EM8,ERR4
                                                                                TRAP    C$ERDF
                                                                                .WORD   8
                                                                                .WORD   EM8
                                                                                .WORD   ERR4
                        BR      16$             ;TAKE ERROR RETURN
002352  9$:     MOV     #17,REGNUM      ;SET REG NO. = 17
                JSR     PC,READLU       ;READ LU REG 17
000004          BITB    #BIT1,@4(SP)    ;GET EXPECTED STATE OF OCOR
                BEQ     12$             ;BR IF EXPECTED OCOR = 0
002336          BITB    #OCOR,REDBYT    ;SEE IF OCOR = 1
                BNE     20$             ;BR IF OCOR = 1
004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
        ;REPORT OCOR NOT SET
                ERRDF   9,EM9,ERR4
                                                                                TRAP    C$ERDF
                                                                                .WORD   9
                                                                                .WORD   EM9
                                                                                .WORD   ERR4
                        BR      16$             ;TAKE ERROR RETURN
002336  12$:    BITB    #OCOR,REDBYT    ;SEE IF OCOR = 0
                BEQ     20$             ;BR IF OCOR = 0
004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
        ;REPORT OCOR NOT CLEARED
                ERRDF   10,EM10,ERR4
                                                                                TRAP    C$ERDF
                                                                                .WORD   10
                                                                                .WORD   EM10
                                                                                .WORD   ERR4
002352  16$:    MOV     2(SP),REGNUM    ;RESTORE LU REG NO.
002320          MOV     PSTACK,SP       ;RESTORE STACK POINTER TO BASE LEVEL
002334          MOV     RETADR,-(SP)    ;FIX ERROR RETURN PC
                BR      23$
000004  20$:    ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
002324          MOV     (SP)+,SUBRPC
002352          MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
        23$:    RTS     PC              ;RETURN


        ;*******************************************************************************
        ;* WAIT50 - THIS SUBROUTINE STALLS FOR AT LEAST 50 MICRO-SEC, AND THEN RETURNS.
        ;*******************************************************************************
        WAIT50: MOV     R1,-(SP)        ;SAVE R1
000310          MOV     #200.,R1        ;INIT COUNTER
        3$:     DEC     R1              ;DECREMENT COUNTER
                BNE     3$              ;BR IF NOT DONE YET
                MOV     (SP)+,R1        ;RESTORE R1
```

```
3201  004650  000207                        RTS     PC              ;RETURN
3202
3203
3204
3205
3206
3207                                ;*********************************************************************************
3208                                ;* STALL - THIS SUBROUTINE STALLS FOR ABOUT A MICRO-SEC.
3209                                ;*********************************************************************************
3210  004652  000240                STALL:  NOP
3211  004654  000240                        NOP
3212  004656  000240                        NOP
3213  004660  000207                        RTS     PC
3214
3215
3216
3217
3218
3219                                ;*********************************************************************************
3220                                ;* LDTXSI - THIS SUBROUTINE LOADS THE TX SILO (REGS 10,11) WITH THE DATA PASSED
3221                                ;*       IN BITS 0-11 OF TXWORD.
3222                                ;*********************************************************************************
3223  004662  013746  002352       LDTXSI: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
3224  004666  042737  170000  002400        BIC     #170000,TXWORD  ;CLEAR UNUSED BITS
3225  004674  012737  000011  002352        MOV     #11,REGNUM      ;SET REG NO. = 11
3226  004702  113737  002401  002340        MOVB    TXWORD+1,WRIBYT ;SET DATA TO BE WRITTEN INTO REG 11
3227  004710  004737  003436               JSR     PC,WRITLU       ;LOAD DATA INTO REG 11
3228  004714  012737  000010  002352        MOV     #10,REGNUM      ;SET REG NO. = 10
3229  004722  113737  002400  002340        MOVB    TXWORD,WRIBYT   ;SET DATA TO BE WRITTEN INTO REG 10
3230  004730  004737  003436               JSR     PC,WRITLU       ;LOAD DATA INTO REG 10
3231  004734  012637  002352               MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
3232  004740  000207                        RTS     PC              ;RETURN
3233
3234
3235
3236
3237
3238                                ;*********************************************************************************
3239                                ;* STPLU - THIS SUBROUTINE CLOCKS THE LINE UNIT FOR THE NO. OF CYCLES PASSED
3240                                ;*       IN BITS 0-14 OF THE WORD FOLLOWING THE CALL.
3241                                ;*       IF BIT 15 = 1, A CHECK IS MADE TO DETERMINE IF THE USYRT CHIP TYPE
3242                                ;*       REQUIRES DECREMENTING THE NO. OF CYCLES BY 1.
3243                                ;*********************************************************************************
3244  004742  010146                STPLU:  MOV     R1,-(SP)        ;SAVE R1
3245  004744  017601  000002               MOV     @2(SP),R1       ;GET DESIRED NO. OF CYCLES
3246  004750  001426                        BEQ     6$              ;IF DESIRED CYCLES = 0, RETURN
3247  004752  100006                        BPL     2$              ;BR IF CHIP TYPE CHECK NOT NECESSARY
3248  004754  042701  100000               BIC     #BIT15,R1       ;CLEAR FLAG BIT
3249  004760  005737  002406               TST     CHPTYP          ;SEE IF SIG USYRT
3250  004764  001401                        BEQ     2$              ;BR IF YES
3251  004766  005301                        DEC     R1              ;DECREMENT CYCLE COUNT
3252  004770  152777  000010  175430  2$:   BISB    #LULOOP,@BSEL1  ;SET LU LOOP BIT
3253  004776  152777  000020  175422  3$:   BISB    #STEPLU,@BSEL1  ;SET THE STEPLU BIT (CLOCK THE TRANSMITTER)
3254  005004  004737  004652               JSR     PC,STALL        ;STALL
3255  005010  142777  000020  175410        BICB    #STEPLU,@BSEL1  ;CLEAR THE STEPLU BIT (CLOCK THE RECEIVER)
3256  005016  004737  004652               JSR     PC,STALL        ;STALL
```

```
3257  005022  005301                          DEC     R1              ;DECREMENT CYCLE COUNTER
3258  005024  001364                          BNE     3$              ;BR IF NOT DONE YET
3259  005026  062766  000002  000002  6$:     ADD     #2,2(SP)        ;FIX UP RETURN PC
3260  005034  012601                          MOV     (SP)+,R1        ;RESTORE R1
3261  005036  000207                          RTS     PC              ;RETURN
3262
3263
3264
3265
3266
3267                                  ;*****************************************************************************
3268                                  ;* OACTIV - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF OACT (REG 11) AND
3269                                  ;*      REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE STATE OF BIT 0 IN THE
3270                                  ;*      WORD FOLLOWING THE CALL.
3271                                  ;*****************************************************************************
3272  005040  013746  002352          OACTIV: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
3273  005044  013746  002324                  MOV     SUBRPC,-(SP)
3274  005050  005737  002324                  TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
3275  005054  001006                          BNE     1$              ;BR IF YES
3276  005056  016637  000004  002324          MOV     4(SP),SUBRPC
3277  005064  162737  000004  002324          SUB     #4,SUBRPC       ;GET PC OF SUBROUTINE CALL
3278  005072  012737  000011  002352  1$:     MOV     #11,REGNUM      ;SET REG NO. = 11
3279  005100  004737  003360                  JSR     PC,READLU       ;READ REG 11
3280  005104  032776  000001  000004          BIT     #BIT0,@4(SP)    ;GET EXPECTED STATE OF OACT
3281  005112  001413                          BEQ     3$              ;BR IF EXPECTED OACT = 0
3282  005114  132737  000100  002336          BITB    #OACT,REDBYT    ;SEE IF OACT = 1
3283  005122  001031                          BNE     9$              ;BR IF OACT = 1
3284  005124  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3285                                  ;REPORT OACT NOT SET
3286  005130                                  ERRDF   11,EM11,ERR4
3287  005130  104455                                                                  TRAP    C$ERDF
3288  005132  000013                                                                  .WORD   11
3289  005134  013623                                                                  .WORD   EM11
3290  005136  016474                                                                  .WORD   ERR4
3291  005140  000412                          BR      6$              ;TAKE ERROR RETURN
3292  005142  132737  000100  002336  3$:     BITB    #OACT,REDBYT    ;SEE IF OACT = 0
3293  005150  001416                          BEQ     9$              ;BR IF OACT = 0
3294  005152  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3295                                  ;REPORT OACT NOT CLEARED
3296  005156                                  ERRDF   12,EM12,ERR4
3297  005156  104455                                                                  TRAP    C$ERDF
3298  005160  000014                                                                  .WORD   12
3299  005162  013640                                                                  .WORD   EM12
3300  005164  016474                                                                  .WORD   ERR4
3301  005166  016637  000002  002352  6$:     MOV     2(SP),REGNUM    ;RESTORE LU REG NO.
3302  005174  013706  002320                  MOV     PSTACK,SP       ;RESTORE PROGRAM STACK TO BASE LEVEL
3303  005200  013746  002334                  MOV     RETADR,-(SP)    ;FIX UP ERROR RETURN PC
3304  005204  000407                          BR      12$
3305  005206  062766  000002  000004  9$:     ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
3306  005214  012637  002324                  MOV     (SP)+,SUBRPC
3307  005220  012637  002352                  MOV     (SP)+,REGNUM    :RESTORE LU REG NO.
3308  005224  000207                  12$:    RTS     PC              ;RETURN
3309
3310
3311
3312
```

```
3313
3314                                    ;*************************************************************************
3315                                    ;* INITRN - THIS SUBROUTINE INITIATES TRANSMISSION OF A MESSAGE, BY DOING A
3316                                    ;*      MASTER CLEAR, LOADING AX2-15 AND REG 17 WITH THE DATA PASSED IN THE 2
3317                                    ;*      WORDS FOLLOWING THE CALL, LOADING 2 SOM CHARS INTO THE TX SILO, AND
3318                                    ;*      CLOCKING THE LINE UNIT UNTIL THE FIRST SYNCH OR FLAG HAS BEEN SERIALIZED
3319                                    ;*      IN THE USYRT. THE PROGRAM MONITORS ORDY,OCOR, AND OACT FOR VALID STATES,
3320                                    ;*      THROUGHOUT THE PROCESS.
3321                                    ;*      IF THE SUBROUTINE DETECTS AN ERROR, A RETURN IS MADE TO THE TEST, AT THE
3322                                    ;*      ADDRESS CONTAINED IN RETADR.
3323                                    ;*************************************************************************
3324  005226  010146                   INITRN: MOV     R1,-(SP)        ;SAVE R1
3325  005230  013746  002352                   MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
3326  005234  013746  002354                   MOV     AXNUM,-(SP)     ;SAVE AX BYTE NO.
3327  005240  016637  000006  002324           MOV     6(SP),SUBRPC
3328  005246  162737  000004  002324           SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
3329  005254  004737  003262                   JSR     PC,MSTCLR       ;ISSUE A MASTER CLEAR
3330  005260  004737  004350                   JSR     PC,OSIRDY       ;CHECK ORDY=1, OCOR=0
3331  005264  000001                           1
3332  005266  004737  005040                   JSR     PC,OACTIV       ;CHK OACT=0
3333  005272  000000                           0
3334  005274  012737  000004  002354           MOV     #4,AXNUM        ;SET AX BYTE NO. = 4 FOR AX2
3335  005302  117637  000006  002346           MOVB    @6(SP),WAX15    ;SET DATA BYTE TO LOAD INTO AX2-15
3336  005310  012737  000400  002400           MOV     #TXSOM,TXWORD   ;SET TSOM BIT
3337  005316  113737  002346  002400           MOVB    WAX15,TXWORD    ;SET SYNCH CHAR
3338  005324  005037  002350                   CLR     WAX16
3339  005330  004737  004000                   JSR     PC,WRITAX       ;LOAD AX2
3340  005334  012737  000017  002352           MOV     #17,REGNUM      ;SET REG NO. = 17
3341  005342  062766  000002  000006           ADD     #2,6(SP)        ;INCR POINTER TO NEXT DATA BYTE
3342  005350  117637  000006  002340           MOVB    @6(SP),WRIBYT   ;SET DATA BYTE TO LOAD INTO REG 17
3343  005356  004737  003436                   JSR     PC,WRITLU       ;LOAD REG 17
3344  005362  004737  004662                   JSR     PC,LDTXSI       ;LOAD THE SILO WITH SOM CHAR
3345  005366  004737  004662                   JSR     PC,LDTXSI       ;LOAD ANOTHER SOM INTO SILO
3346  005372  004737  004634                   JSR     PC,WAIT50       ;WAIT FOR DATA TO RIPPLE
3347  005376  004737  004350                   JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=1
3348  005402  000003                           3
3349  005404  004737  005040                   JSR     PC,OACTIV       ;CHK FOR OACT = 0
3350  005410  000000                           0
3351  005412  005001                           CLR     R1              ;INIT CYCLE COUNTER
3352  005414  012737  000011  002352           MOV     #11,REGNUM      ;SET LU REG NO. = 11
3353  005422  152777  000010  174776  6$:      BISB    #LULOOP,@BSEL1  ;SET LINE UNIT LOOP BIT
3354  005430  152777  000020  174770           BISB    #STEPLU,@BSEL1  ;SET CLOCK BIT
3355  005436  004737  004652                   JSR     PC,STALL        ;STALL FOR MICRO-SEC
3356  005442  004737  003360                   JSR     PC,READLU       ;READ REG 11
3357  005446  132737  000100  002336           BITB    #OACT,REDBYT    ;SEE IF OACT = 1 YET
3358  005454  001014                           BNE     9$              ;BR IF OACT = 1
3359  005456  142777  000020  174742           BICB    #STEPLU,@BSEL1  ;CLEAR CLOCK BIT
3360  005464  004737  004652                   JSR     PC,STALL        ;STALL FOR A MICRO-SEC
3361  005470  005201                           INC     R1              ;INCR CYCLE COUNT
3362  005472  020127  000003                   CMP     R1,#3           ;SEE IF 3 CYCLES DONE YET
3363  005476  002751                           BLT     6$              ;BR IF NOT
3364  005500  004737  005040                   JSR     PC,OACTIV       ;CHK FOR OACT = 1
3365  005504  000001                           1
3366  005506  012737  000017  002352  9$:      MOV     #17,REGNUM      ;SET REG NO. = 17
3367  005514  005037  002406                   CLR     CHPTYP          ;CLEAR USYRT CHIP INDICATOR
3368  005520  004737  003360                   JSR     PC,READLU       ;READ REG 17
```

```
3369  005524  132737  000020  002336          BITB    #OCOR,REDBYT    ;CHK FOR OCOR CLEARED YET
3370  005532  001403                          BEQ     12$             ;BR IF YES - IT IS SIG CHIP
3371  005534  012737  000001  002406          MOV     #1,CHPTYP       ;SET INDICATOR FOR OTHER CHIP TYPE
3372  005542  142777  000020  174656  12$:    BICB    #STEPLU,@BSEL1  ;CLEAR CLOCK BIT
3373  005550  004737  004652                  JSR     PC,STALL        ;STALL FOR MICRO-SEC
3374  005554  004737  004350                  JSR     PC,OSIRDY       ;CHK FOR ORDY = 1, OCOR = 0
3375  005560  000001                          1
3376  005562  062766  000002  000006          ADD     #2,6(SP)        ;FIX UP RETURN PC
3377  005570  012637  002354                  MOV     (SP)+,AXNUM     ;RESTORE AX BYTE NO.
3378  005574  012637  002352                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
3379  005600  012601                          MOV     (SP)+,R1        ;RESTORE R1
3380  005602  005037  002324                  CLR     SUBRPC          ;CLEAR SUBR CALL PC
3381  005606  000207                          RTS     PC              ;RETURN
3382
3383
3384
3385
3386
3387                                  ;*****************************************************************************
3388                                  ;* TXCHAR - THIS SUBROUTINE INITIATES TRANSMISSION OF A CHARACTER, BY LOADING
3389                                  ;*      THE TX SILO WITH DATA PASSED IN BITS 0-11 OF THE WORD FOLLOWING THE CALL
3390                                  ;*      AND CLOCKS THE LINE UNIT WITH THE NUMBER OF CYCLES PASSED IN BITS 0-14
3391                                  ;*      OF THE SECOND WORD FOLLOWING THE CALL. IF BIT 15 = 1, A CHK IS MADE TO
3392                                  ;*      DETERMINE IF THE USYRT CHIP TYPE REQUIRES DECREMENTING THE NO. OF CYCLES
3393                                  ;*      BY 1. THE PROGRAM CHECKS FOR VALID STATES OF ORDY,
3394                                  ;*      OCOR, AND OACT THROUGHOUT THE PROCESS.
3395                                  ;*      IF AN ERROR IS DETECTED, A RETURN IS MADE TO THE TEST, AT THE ADDRESS
3396                                  ;*      CONTAINED IN RETADR.
3397                                  ;*****************************************************************************
3398  005610  010146                  TXCHAR: MOV     R1,-(SP)        ;SAVE R1
3399  005612  010246                          MOV     R2,-(SP)        ;SAVE R2
3400  005614  016637  000004  002324          MOV     4(SP),SUBRPC
3401  005622  162737  000004  002324          SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
3402  005630  017637  000004  002400          MOV     @4(SP),TXWORD   ;GET DATA TO BE TRANSMITTED
3403  005636  004737  004662                  JSR     PC,LDTXSI       ;LOAD THE TX SILO WITH THE DATA
3404  005642  004737  004634                  JSR     PC,WAIT50       ;WAIT FOR DATA TO RIPPLE DOWN SILO
3405  005646  062766  000002  000004          ADD     #2,4(SP)        ;INCR POINTER
3406  005654  005001                          CLR     R1              ;INIT CYCLE COUNT
3407  005656  017602  000004                  MOV     @4(SP),R2       ;GET DESIRED NO. OF CYCLES
3408  005662  005702                          TST     R2              ;SEE IF CHIP TYPE CHK SHOULD BE MADE
3409  005664  100006                          BPL     9$              ;BR IF NOT
3410  005666  042702  100000                  BIC     #BIT15,R2       ;CLEAR FLAG BIT
3411  005672  005737  002406                  TST     CHPTYP          ;SEE IF SIG USYRT
3412  005676  001401                          BEQ     9$              ;BR IF YES
3413  005700  005302                          DEC     R2              ;DECREMENT NO. OF CYCLES
3414  005702  004737  005040          9$:     JSR     PC,OACTIV       ;CHK OACT = 1
3415  005706  000001                          1
3416  005710  020102                          CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
3417  005712  001410                          BEQ     12$             ;BR IF YES
3418  005714  004737  004350                  JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=1
3419  005720  000003                          3
3420  005722  004737  004742                  JSR     PC,STPLU        ;STEP LU ONE CYCLE
3421  005726  000001                          1
3422  005730  005201                          INC     R1              ;INCR CYCLE COUNT
3423  005732  000763                          BR      9$
3424  005734  004737  004350          12$:    JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=0
```

```
CZDMSF.P11    30-SEP-81 15:40          GLOBAL SUBROUTINES

  3425  005740  000001                          1
  3426  005742  062766  000002  000004          ADD     #2,4(SP)        ;FIX UP RETURN PC
  3427  005750  005037  002324                  CLR     SUBRPC          ;CLEAR SUBR CALL PC
  3428  005754  012602                          MOV     (SP)+,R2        ;RESTORE R2
  3429  005756  012601                          MOV     (SP)+,R1        ;RESTORE R1
  3430  005760  000207                          RTS     PC              ;RETURN
  3431
  3432
  3433
  3434
  3435
  3436                                  ;*************************************************************************
  3437                                  ;* ISIRDY - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF ICIR (REG 17)
  3438                                  ;*       AND IRDY (REG 12) AND REPORTS AN ERROR IF EITHER IS NOT PROPERLY SET
  3439                                  ;*       AS PASSED IN BIT 0 (ICIR) AND BIT 1 (IRDY) OF THE WORD FOLLOWING THE
  3440                                  ;*       CALL.
  3441                                  ;*       IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS
  3442                                  ;*       IN RETADR.
  3443                                  ;*************************************************************************
  3444  005762  013746  002352          ISIRDY: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
  3445  005766  013746  002324                  MOV     SUBRPC,-(SP)
  3446  005772  005737  002324                  TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
  3447  005776  001006                          BNE     1$              ;BR IF YES
  3448  006000  016637  000004  002324          MOV     4(SP),SUBRPC
  3449  006006  162737  000004  002324          SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
  3450  006014  012737  000012  002352  1$:     MOV     #12,REGNUM      ;SET REG NO. TO 12
  3451  006022  004737  003360                  JSR     PC,READLU       ;READ REG 12
  3452  006026  032776  000002  000004          BIT     #BIT1,@4(SP)    ;GET EXPECTED STATE OF IRDY
  3453  006034  001413                          BEQ     3$              ;BR IF EXPECTED IRDY = 0
  3454  006036  132737  000020  002336          BITB    #IRDY,REDBYT    ;SEE IF IRDY = 1
  3455  006044  001022                          BNE     9$              ;BR IF IRDY = 1
  3456  006046  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
  3457                                  ;REPORT IRDY NOT SET
  3458  006052                                  ERRDF   17,EM17,ERR4
  3459  006052  104455                                                                  TRAP    C$ERDF
  3460  006054  000021                                                                  .WORD   17
  3461  006056  013655                                                                  .WORD   EM17
  3462  006060  016474                                                                  .WORD   ERR4
  3463  006062  000451                          BR      16$             ;TAKE ERROR EXIT
  3464  006064  132737  000020  002336  3$:     BITB    #IRDY,REDBYT    ;SEE IF IRDY = 0
  3465  006072  001407                          BEQ     9$              ;BR IF IRDY = 0
  3466  006074  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
  3467                                  ;REPORT IRDY NOT CLEARED
  3468  006100                                  ERRDF   18,EM18,ERR4
  3469  006100  104455                                                                  TRAP    C$ERDF
  3470  006102  000022                                                                  .WORD   18
  3471  006104  013672                                                                  .WORD   EM18
  3472  006106  016474                                                                  .WORD   ERR4
  3473  006110  000436                          BR      16$             ;TAKE ERROR RETURN
  3474  006112  012737  000017  002352  9$:     MOV     #17,REGNUM      ;SET REG NO. = 17
  3475  006120  004737  003360                  JSR     PC,READLU       ;READ REG 17
  3476  006124  132776  000001  000004          BITB    #BIT0,@4(SP)    ;GET EXPECTED STATE OF ICIR
  3477  006132  001413                          BEQ     12$             ;BR IF EXPECTED ICIR = 0
  3478  006134  132737  000010  002336          BITB    #ICIR,REDBYT    ;SEE IF ICIR = 1
  3479  006142  001031                          BNE     20$             ;BR IF ICIR = 1
  3480  006144  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
```

```
CZDMSF.P11    30-SEP-81 15:40          GLOBAL SUBROUTINES

 3481                                  ;REPORT ICIR NOT SET
 3482  006150                                 ERRDF   19,EM19,ERR4
 3483  006150  104455                                                                TRAP    C$ERDF
 3484  006152  000023                                                                .WORD   19
 3485  006154  013707                                                                .WORD   EM19
 3486  006156  016474                                                                .WORD   ERR4
 3487  006160  000412                        BR      16$             ;TAKE ERROR RETURN
 3488  006162  132737  000010  002336  12$:  BITB    #ICIR,REDBYT    ;SEE IF ICIR = 0
 3489  006170  001416                        BEQ     20$             ;BR IF ICIR = 0
 3490  006172  004737  004214                JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 3491                                  ;REPORT ICIR NOT CLEARED
 3492  006176                                 ERRDF   20,EM20,ERR4
 3493  006176  104455                                                                TRAP    C$ERDF
 3494  006200  000024                                                                .WORD   20
 3495  006202  013724                                                                .WORD   EM20
 3496  006204  016474                                                                .WORD   ERR4
 3497  006206  016637  000002  002352  16$:  MOV     2(SP),REGNUM    ;RESTORE LU REG NO.
 3498  006214  013706  002320                MOV     PSTACK,SP       ;RESTORE STACK POINTER TO BASE LEVEL
 3499  006220  013746  002334                MOV     RETADR,-(SP)    ;FIX ERROR RETURN PC
 3500  006224  000407                        BR      23$
 3501  006226  062766  000002  000004  20$:  ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
 3502  006234  012637  002324                MOV     (SP)+,SUBRPC
 3503  006240  012637  002352                MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
 3504  006244  000207                  23$:  RTS     PC              ;RETURN
 3505
 3506
 3507
 3508
 3509
 3510                                  ;*****************************************************************************
 3511                                  ;* IACTIV - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF IACT (REG 12) AND
 3512                                  ;*      REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE STATE OF BIT 0 IN THE
 3513                                  ;*      WORD FOLLOWING THE CALL.
 3514                                  ;*      IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS IN
 3515                                  ;*      RETADR.
 3516                                  ;*****************************************************************************
 3517  006246  013746  002352          IACTIV: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
 3518  006252  013746  002324                  MOV     SUBRPC,-(SP)
 3519  006256  005737  002324                  TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
 3520  006262  001006                          BNE     1$              ;BR IF YES
 3521  006264  016637  000004  002324          MOV     4(SP),SUBRPC
 3522  006272  162737  000004  002324          SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
 3523  006300  012737  000012  002352  1$:     MOV     #12,REGNUM      ;SET REG NO. = 12
 3524  006306  004737  003360                  JSR     PC,READLU       ;READ REG 12
 3525  006312  032776  000001  000004          BIT     #BIT0,@4(SP)    ;GET EXPECTED STATE OF IACT
 3526  006320  001413                          BEQ     3$              ;BR IF EXPECTED IACT = 0
 3527  006322  132737  000100  002336          BITB    #IACT,REDBYT    ;SEE IF IACT = 1
 3528  006330  001031                          BNE     9$              ;BR IF IACT = 1
 3529  006332  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 3530                                  ;REPORT IACT NOT SET
 3531  006336                                 ERRDF   21,EM21,ERR4
 3532  006336  104455                                                                TRAP    C$ERDF
 3533  006340  000025                                                                .WORD   21
 3534  006342  013741                                                                .WORD   EM21
 3535  006344  016474                                                                .WORD   ERR4
 3536  006346  000412                          BR      6$              ;TAKE ERROR EXIT
```

```
3537  006350  132737  000100  002336  3$:     BITB    #IACT,REDBYT    ;SEE IF IACT = 0
3538  006356  001416                          BEQ     9$              ;BR IF IACT = 0
3539  006360  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3540                                  ;REPORT IACT NOT CLEARED
3541  006364                                  ERRDF   22,EM22,ERR4
3542  006364  104455                                                                 TRAP    C$ERDF
3543  006366  000026                                                                 .WORD   22
3544  006370  013756                                                                 .WORD   EM22
3545  006372  016474                                                                 .WORD   ERR4
3546  006374  016637  000002  002352  6$:     MOV     2(SP),REGNUM    ;RESTORE LU REG NO.
3547  006402  013706  002320                  MOV     PSTACK,SP       ;RESTORE PROGRAM STACK TO BASE LEVEL
3548  006406  013746  002334                  MOV     RETADR,-(SP)    ;FIX UP ERROR RETURN PC
3549  006412  000407                          BR      12$
3550  006414  062766  000002  000004  9$:     ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
3551  006422  012637  002324                  MOV     (SP)+,SUBRPC
3552  006426  012637  002352                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
3553  006432  000207                  12$:    RTS     PC              ;RETURN
3554
3555
3556
3557
3558
3559                                  ;*********************************************************************************
3560                                  ;* RSEOM - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF RSOM AND REOM IN
3561                                  ;* AX0-16, AND REPORTS AN ERROR IF EITHER IS NOT SET TO THE STATE PASSED IN BITS
3562                                  ;* 0,1, RESPECTIVELY, OF THE WORD FOLLOWING THE CALL.
3563                                  ;* IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS IN RETADR.
3564                                  ;*********************************************************************************
3565  006434  013746  002354          RSEOM:  MOV     AXNUM,-(SP)     ;SAVE AX BYTE NO.
3566  006440  013746  002324                  MOV     SUBRPC,-(SP)
3567  006444  005737  002324                  TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
3568  006450  001006                          BNE     1$              ;BR IF YES
3569  006452  016637  000004  002324          MOV     4(SP),SUBRPC
3570  006460  162737  000004  002324          SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
3571  006466  012737  000001  002354  1$:     MOV     #1,AXNUM        ;SET AX BYTE NO. FOR AX0-16
3572  006474  004737  003612                  JSR     PC,READAX       ;READ AX0
3573  006500  032776  000001  000004          BIT     #BIT0,@4(SP)    ;GET EXPECTED STATE OF RSOM
3574  006506  001413                          BEQ     3$              ;BR IF EXPECTED RSOM = 0
3575  006510  132737  000001  002344          BITB    #RSOM,RAX16     ;SEE IF RSOM = 1
3576  006516  001022                          BNE     9$              ;BR IF RSOM = 1
3577  006520  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3578                                  ;REPORT RSOM NOT SET
3579  006524                                  ERRDF   29,EM29,ERR6
3580  006524  104455                                                                 TRAP    C$ERDF
3581  006526  000035                                                                 .WORD   29
3582  006530  014010                                                                 .WORD   EM29
3583  006532  017664                                                                 .WORD   ERR6
3584  006534  000444                          BR      16$             ;TAKE ERROR EXIT
3585  006536  132737  000001  002344  3$:     BITB    #RSOM,RAX16     ;SEE IF RSOM = 0
3586  006544  001407                          BEQ     9$              ;BR IF RSOM = 0
3587  006546  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3588                                  ;REPORT RSOM NOT CLEARED
3589  006552                                  ERRDF   28,EM28,ERR6
3590  006552  104455                                                                 TRAP    C$ERDF
3591  006554  000034                                                                 .WORD   28
3592  006556  013773                                                                 .WORD   EM28
```

```
3593  006560  017664                                                                    .WORD   ERR6
3594  006562  000431                    BR      16$             ;TAKE ERROR RETURN
3595  006564  132776  000002  000004  9$:   BITB    #BIT1,@4(SP)    ;GET EXPECTED STATE OF REOM
3596  006572  001413                    BEQ     12$             ;BR IF EXPECTED REOM = 0
3597  006574  132737  000002  002344        BITB    #REOM,RAX16     ;SEE IF REOM = 1
3598  006602  001031                    BNE     20$             ;BR IF REOM = 1
3599  006604  004737  004214            JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3600                                  ;REPORT REOM NOT SET
3601  006610                          ERRDF   31,EM31,ERR6
3602  006610  104455                                                                    TRAP    C$ERDF
3603  006612  000037                                                                    .WORD   31
3604  006614  014042                                                                    .WORD   EM31
3605  006616  017664                                                                    .WORD   ERR6
3606  006620  000412                    BR      16$             ;TAKE ERROR RETURN
3607  006622  132737  000002  002344  12$:   BITB    #REOM,RAX16     ;SEE IF REOM = 0
3608  006630  001416                    BEQ     20$             ;BR IF REOM = 0
3609  006632  004737  004214            JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3610                                  ;REPORT REOM NOT CLEARED
3611  006636                          ERRDF   30,EM30,ERR6
3612  006636  104455                                                                    TRAP    C$ERDF
3613  006640  000036                                                                    .WORD   30
3614  006642  014025                                                                    .WORD   EM30
3615  006644  017664                                                                    .WORD   ERR6
3616  006646  016637  000002  002354  16$:   MOV     2(SP),AXNUM     ;RESTORE AX BYTE NO.
3617  006654  013706  002320            MOV     PSTACK,SP       ;RESTORE STACK POINTER TO BASE LEVEL
3618  006660  013746  002334            MOV     RETADR,-(SP)    ;FIX ERROR RETURN PC
3619  006664  000407                    BR      23$
3620  006666  062766  000002  000004  20$:   ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
3621  006674  012637  002324            MOV     (SP)+,SUBRPC
3622  006700  012637  002354            MOV     (SP)+,AXNUM     ;RESTORE AX BYTE NO.
3623  006704  000207                23$:   RTS     PC              ;RETURN
3624
3625
3626
3627
3628
3629                                  ;*****************************************************************************
3630                                  ;* RDRXSI - THIS SUBROUTINE READS THE RCV SILO (REGS 10,12) AND RETURNS THE
3631                                  ;*       SILO ENTRY IN BITS 0-11 OF RXWORD.
3632                                  ;*****************************************************************************
3633  006706  013746  002352        RDRXSI: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
3634  006712  012737  000012  002352        MOV     #12,REGNUM      ;SET REG NO. = 12
3635  006720  004737  003360            JSR     PC,READLU       ;READ LU REG 12
3636  006724  113737  002336  002403        MOVB    REDBYT,RXWORD+1 ;GET HI BITS OF SILO ENTRY
3637  006732  042737  170000  002402        BIC     #170000,RXWORD  ;CLEAR UNUSED BITS
3638  006740  012737  000010  002352        MOV     #10,REGNUM      ;SET REG NO. = 10
3639  006746  004737  003360            JSR     PC,READLU       ;READ REG 10
3640  006752  113737  002336  002402        MOVB    REDBYT,RXWORD   ;GET LOW BITS OF SILO ENTRY
3641  006760  012637  002352            MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
3642  006764  000207                    RTS     PC              ;RETURN
3643
3644
3645
3646
3647
3648                                  ;*****************************************************************************
```

```
3649                                    ;* RCV1ST - THIS SUBROUTINE RECEIVES THE FIRST CHAR OF A MESSAGE, AND MONITORS
3650                                    ;*      STATUS OF THE RECEIVER. FIRST, A CHECK IS MADE FOR IACT = 0, IRDY = 0,
3651                                    ;*      ICIR = 1, AND RSOM = 0. THEN, THE LINE UNIT IS CLOCKED USING
3652                                    ;*      STEPLU UNTIL IRDY = 1. THE PROGRAM CHECKS FOR THIS TO OCCUR WITHIN 3
3653                                    ;*      CYCLES AFTER THE NO. OF CYCLES PASSED IN THE WORD FOLLOWING THE CALL.
3654                                    ;*      IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST, AT THE ADDRESS
3655                                    ;*      CONTAINED IN RETADR.
3656                                    ;*****************************************************************************
3657  006766  010146                   RCV1ST: MOV     R1,-(SP)        ;SAVE R1
3658  006770  010346                           MOV     R3,-(SP)        ;SAVE R3
3659  006772  013746  002352                   MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
3660  006776  016637  000006  002324           MOV     6(SP),SUBRPC
3661  007004  162737  000004  002324           SUB     #4,SUBRPC       ;GET PC OF SUBROUTINE CALL
3662  007012  012737  000012  002352           MOV     #12,REGNUM      ;SET LU REG NO. = 12
3663  007020  005001                           CLR     R1              ;INIT CYCLE COUNT TO 0
3664  007022  017603  000006                   MOV     @6(SP),R3       ;GET CYCLE COUNT LIMIT
3665  007026  062703  000003                   ADD     #3,R3
3666  007032  005776  000006                   TST     @6(SP)          ;SEE IF DESIRED CYCLES = 0
3667  007036  001414                           BEQ     8$              ;BR IF YES
3668  007040  004737  006246                   JSR     PC,IACTIV       ;CHK FOR IACT = 0
3669  007044  000000                           0
3670  007046  004737  005762                   JSR     PC,ISIRDY       ;CHK FOR ICIR = 1, IRDY = 0
3671  007052  000001                           1
3672  007054  004737  006434                   JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0 IN AX0-16
3673  007060  000000                           0
3674  007062  004737  004742           6$:     JSR     PC,STPLU        ;CLOCK LU FOR 1 CYCLE
3675  007066  000001                           1
3676  007070  004737  004634           8$:     JSR     PC,WAIT50       ;ALLOW SILO DATA TO RIPPLE
3677  007074  005201                           INC     R1              ;INCREMENT CYCLE COUNT
3678  007076  004737  003360                   JSR     PC,READLU       ;READ REG 12
3679  007102  132737  000020  002336           BITB    #IRDY,REDBYT    ;SEE IF IRDY = 1 YET
3680  007110  001005                           BNE     9$              ;BR IF IRDY = 1
3681  007112  020103                           CMP     R1,R3           ;SEE IF LIMIT EXCEEDED
3682  007114  002762                           BLT     6$              ;BR IF NOT YET
3683  007116  004737  005762                   JSR     PC,ISIRDY       ;CHK FOR ICIR = 1, IRDY = 1
3684  007122  000003                           3
3685  007124  020176  000006           9$:     CMP     R1,@6(SP)       ;SEE IF LESS THAN REQUIRED CYCLES
3686  007130  002003                           BGE     12$             ;BR IF NOT
3687  007132  004737  005762                   JSR     PC,ISIRDY       ;CHK FOR ICIR = 1, IRDY = 0
3688  007136  000001                           1
3689  007140  004737  006246           12$:    JSR     PC,IACTIV       ;CHK FOR IACT = 1
3690  007144  000001                           1
3691  007146  004737  005762                   JSR     PC,ISIRDY       ;CHK FOR ICIR = 1, IRDY = 1
3692  007152  000003                           3
3693  007154  062766  000002  000006           ADD     #2,6(SP)        ;FIX UP RETURN PC
3694  007162  012637  002352                   MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
3695  007166  012603                           MOV     (SP)+,R3        ;RESTORE R3
3696  007170  012601                           MOV     (SP)+,R1        ;RESTORE R1
3697  007172  005037  002324                   CLR     SUBRPC          ;CLEAR SUBR CALL PC
3698  007176  000207                           RTS     PC              ;RETURN
3699
3700
3701
3702
3703
3704                                    ;*****************************************************************************
```

```
3705                                    ;* STPERR - THIS SUBROUTINE LOADS THE CONTENTS OF THE FIRST WORD FOLLOWING THE
3706                                    ;*      CALL INTO REG 17, AND SETS THE IERR BIT, AND CLOCKS THE LINE UNIT
3707                                    ;*      FOR THE NO. OF CYCLES PASSED IN THE 2ND WORD FOLLOWING THE CALL. THEN,
3708                                    ;*      IT RESTORES REG 17 TO ITS ORIGINAL CONTENTS, CLEARING THE IERR BIT.
3709                                    ;*****************************************************************************
3710  007200  013746  002352           STPERR: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
3711  007204  012737  000017  002352           MOV     #17,REGNUM      ;SET LU REG NO. = 17
3712  007212  017637  000002  002340           MOV     @2(SP),WRIBYT
3713  007220  152737  000002  002340           BISB    #IERR,WRIBYT
3714  007226  004737  003436                    JSR     PC,WRITLU       ;SET IERR BIT IN REG 17
3715  007232  062766  000002  000002           ADD     #2,2(SP)        ;INCREMENT SUBR ARGUMENT POINTER
3716  007240  017637  000002  007252           MOV     @2(SP),3$       ;GET DESIRED NO. OF CYCLES
3717  007246  004737  004742                    JSR     PC,STPLU        ;CLOCK LU FOR DESIRED NO. OF CYCLES
3718  007252  000000                    3$:     .WORD   0               ;NO. OF CYCLES GOES HERE
3719  007254  142737  000002  002340           BICB    #IERR,WRIBYT
3720  007262  004737  003436                    JSR     PC,WRITLU       ;CLEAR IERR BIT IN REG 17
3721  007266  062766  000002  000002           ADD     #2,2(SP)        ;FIX UP RETURN PC
3722  007274  012637  002352                    MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
3723  007300  000207                           RTS     PC              ;RETURN
3724
3725
3726
3727
3728
3729                                    ;*****************************************************************************
3730                                    ;* CKDATA - THIS SUBROUTINE READS THE RCV SILO AND COMPARES THE SILO ENTRY
3731                                    ;*      TO BITS 0-11 OF THE FIRST WORD FOLLOWING THE CALL. IF THERE IS A
3732                                    ;*      MISMATCH, THE ERROR IS REPORTED AND A RETURN IS MADE TO THE TEST AT THE
3733                                    ;*      ADDRESS CONTAINED IN RETADR. IF BIT 15 = 0 IN THE FIRST WORD
3734                                    ;*      FOLLOWING THE CALL, THE SUBROUTINE WILL NOT CHECK THE BCC BIT (SILO
3735                                    ;*      BIT 8). IF THERE ARE NO ERRORS, THE LINE UNIT IS CLOCKED FOR THE
3736                                    ;*      NUMBER OF CYCLES PASSED IN THE SECOND WORD FOLLOWING THE CALL.
3737                                    ;*****************************************************************************
3738  007302  010146                    CKDATA: MOV     R1,-(SP)        ;SAVE R1
3739  007304  013746  002352                    MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
3740  007310  016637  000004  002324           MOV     4(SP),SUBRPC
3741  007316  162737  000004  002324           SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
3742  007324  017601  000004                    MOV     @4(SP),R1       ;GET EXPECTED SILO ENTRY
3743  007330  042701  170000                    BIC     #170000,R1      ;CLEAR UNUSED BITS FOR COMPARE
3744  007334  004737  006706                    JSR     PC,RDRXSI       ;READ RCV SILO
3745  007340  023727  002412  000347           CMP     SAVLEN,#TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0
3746  007346  001005                           BNE     4$              ;BR IF CHAR LENGTH NOT = 7
3747  007350  042701  000200                    BIC     #BIT7,R1        ;MASK OFF 8TH BIT
3748  007354  042737  000200  002402           BIC     #BIT7,RXWORD
3749  007362  120137  002402           4$:     CMPB    R1,RXWORD       ;COMPARE EXPECTED BITS 0-7 TO ACTUAL
3750  007366  001445                           BEQ     6$              ;BR IF MATCH
3751  007370  005037  002356                    CLR     GOODAT
3752  007374  110137  002356                    MOVB    R1,GOODAT       ;GET EXPECTED DATA
3753  007400  005037  002360                    CLR     BADDAT
3754  007404  113737  002402  002360           MOVB    RXWORD,BADDAT   ;GET ACTUAL DATA
3755  007412  012737  000011  002352           MOV     #11,REGNUM      ;SET REG NO. = 11
3756  007420  004737  003360                    JSR     PC,READLU       ;READ REG 11
3757  007424  132737  000001  002336           BITB    #UNRR,REDBYT    ;SEE IF TX UNDERRUN ERROR
3758  007432  001410                           BEQ     5$              ;BR IF NOT,TO REPORT DATA ERROR
3759  007434  004737  004214                    JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3760                                  ;REPORT TX UNDERRUN ERROR
```

```
007440                                       ERRDF   54,EM54,ERR4
007440  104455                                                                      TRAP    C$ERDF
007442  000066                                                                      .WORD   54
007444  014667                                                                      .WORD   EM54
007446  016474                                                                      .WORD   ERR4
007450  000137  010132                       JMP     36$             ;TAKE ERROR EXIT
007454  012737  000010  002352   5$:         MOV     #10,REGNUM      ;SET REG NO. = 10
007462  004737  004214                       JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                 ;REPORT RCV'D DATA MISCOMPARE
007466                                       ERRDF   34,EM34,ERR8
007466  104455                                                                      TRAP    C$ERDF
007470  000042                                                                      .WORD   34
007472  014057                                                                      .WORD   EM34
007474  020774                                                                      .WORD   ERR8
007476  000137  010132                       JMP     36$             ;TAKE ERROR EXIT
007502  000301                   6$:         SWAB    R1
007504  012737  000012  002352               MOV     #12,REGNUM      ;SET LU REG NO. FOR ERROR REPORTS
007512  120137  002403                       CMPB    R1,RXWORD+1     ;COMPARE EXPECTED SILO BITS 8-11 TO ACTUAL
007516  001002                               BNE     7$              ;BR IF MISMATCH
007520  000137  010106                       JMP     22$             ;CONTINUE
007524  005037  002356           7$:         CLR     GOODAT
007530  110137  002356                       MOVB    R1,GOODAT       ;SET EXPECTED DATA
007534  005037  002360                       CLR     BADDAT
007540  113737  002403  002360               MOVB    RXWORD+1,BADDAT ;SET ACTUAL DATA
007546  032776  100000  000004               BIT     #BCCCHK,@4(SP)  ;SEE IF BCC SHOULD BE IGNORED
007554  001433                               BEQ     10$             ;BR IF YES
007556  132701  000001                       BITB    #BCC,R1         ;SEE IF EXPECTED BIT = 1
007562  001014                               BNE     8$              ;BR IF YES
007564  132737  000001  002403               BITB    #BCC,RXWORD+1   ;SEE IF ACTUAL BIT = 0
007572  001424                               BEQ     10$             ;BR IF YES
007574  004737  004214                       JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                 ;REPORT BCC NOT CLEARED
007600                                       ERRDF   35,EM35,ERR8
007600  104455                                                                      TRAP    C$ERDF
007602  000043                                                                      .WORD   35
007604  014105                                                                      .WORD   EM35
007606  020774                                                                      .WORD   ERR8
007610  000137  010132                       JMP     36$             ;TAKE ERROR EXIT
007614  132737  000001  002403   8$:         BITB    #BCC,RXWORD+1   ;SEE IF ACTUAL BIT = 1
007622  001010                               BNE     10$             ;BR IF YES
007624  004737  004214                       JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                 ;REPORT BCC NOT SET
007630                                       ERRDF   36,EM36,ERR8
007630  104455                                                                      TRAP    C$ERDF
007632  000044                                                                      .WORD   36
007634  014121                                                                      .WORD   EM36
007636  020774                                                                      .WORD   ERR8
007640  000137  010132                       JMP     36$             ;TAKE ERROR EXIT
007644                           10$:
007644  132701  000002                       BITB    #EBLK,R1        ;SEE IF EXPECTED BIT = 1
007650  001014                               BNE     12$             ;BR IF YES
007652  132737  000002  002403               BITB    #EBLK,RXWORD+1  ;SEE IF ACTUAL BIT = 0
007660  001424                               BEQ     14$             ;BR IF YES
007662  004737  004214                       JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                 ;REPORT EBLK NOT CLEARED
007666                                       ERRDF   37,EM37,ERR8
```

```
3817  007666  104455                                                                  TRAP    C$ERDF
3818  007670  000045                                                                  .WORD   37
3819  007672  014135                                                                  .WORD   EM37
3820  007674  020774                                                                  .WORD   ERR8
3821  007676  000137  010132          JMP     36$             ;TAKE ERROR EXIT
3822  007702  132737  000002  002403  12$:    BITB    #EBLK,RXWORD+1  ;SEE IF ACTUAL BIT = 1
3823  007710  001010                  BNE     14$             ;BR IF YES
3824  007712  004737  004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3825                          ;REPORT EBLK NOT SET
3826  007716                  ERRDF   38,EM38,ERR8
3827  007716  104455                                                                  TRAP    C$ERDF
3828  007720  000046                                                                  .WORD   38
3829  007722  014152                                                                  .WORD   EM38
3830  007724  020774                                                                  .WORD   ERR8
3831  007726  000137  010132          JMP     36$             ;TAKE ERROR EXIT
3832  007732                  14$:
3833  007732  132701  000004          BITB    #RAB,R1         ;SEE IF EXPECTED BIT = 1
3834  007736  001014                  BNE     16$             ;BR IF YES
3835  007740  132737  000004  002403  BITB    #RAB,RXWORD+1   ;SEE IF ACTUAL BIT = 0
3836  007746  001424                  BEQ     18$             ;BR IF YES
3837  007750  004737  004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3838                          ;REPORT RAB NOT CLEARED
3839  007754                  ERRDF   39,EM39,ERR8
3840  007754  104455                                                                  TRAP    C$ERDF
3841  007756  000047                                                                  .WORD   39
3842  007760  014167                                                                  .WORD   EM39
3843  007762  020774                                                                  .WORD   ERR8
3844  007764  000137  010132          JMP     36$             ;TAKE ERROR EXIT
3845  007770  132737  000004  002403  16$:    BITB    #RAB,RXWORD+1   ;SEE IF ACTUAL BIT = 1
3846  007776  001010                  BNE     18$             ;BR IF YES
3847  010000  004737  004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3848                          ;REPORT RAB NOT SET
3849  010004                  ERRDF   40,EM40,ERR8
3850  010004  104455                                                                  TRAP    C$ERDF
3851  010006  000050                                                                  .WORD   40
3852  010010  014203                                                                  .WORD   EM40
3853  010012  020774                                                                  .WORD   ERR8
3854  010014  000137  010132          JMP     36$             ;TAKE ERROR EXIT
3855  010020                  18$:
3856  010020  132701  000010          BITB    #OVRR,R1        ;SEE IF EXPECTED BIT = 1
3857  010024  001014                  BNE     20$             ;BR IF YES
3858  010026  132737  000010  002403  BITB    #OVRR,RXWORD+1  ;SEE IF ACTUAL BIT = 0
3859  010034  001424                  BEQ     22$             ;BR IF YES
3860  010036  004737  004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3861                          ;REPORT OVRR NOT CLEARED
3862  010042                  ERRDF   41,EM41,ERR8
3863  010042  104455                                                                  TRAP    C$ERDF
3864  010044  000051                                                                  .WORD   41
3865  010046  014217                                                                  .WORD   EM41
3866  010050  020774                                                                  .WORD   ERR8
3867  010052  000137  010132          JMP     36$             ;TAKE ERROR EXIT
3868  010056  132737  000010  002403  20$:    BITB    #OVRR,RXWORD+1  ;SEE IF ACTUAL BIT = 1
3869  010064  001010                  BNE     22$             ;BR IF YES
3870  010066  004737  004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3871                          ;REPORT OVRR NOT SET
3872  010072                  ERRDF   42,EM42,ERR8
```

```
                                                                                TRAP    C$ERDF
                                                                                .WORD   42
                                                                                .WORD   EM42
                                                                                .WORD   ERR8
                        JMP     36$             ;TAKE ERROR EXIT
22$:
                        ADD     #2,4(SP)        ;INCR SUBROUTINE ARGUMENT POINTER
                        MOV     @4(SP),24$      ;GET DESIRED CYCLE COUNT
                        JSR     PC,STPLU        ;CLOCK LU FOR DESIRED CYCLES
24$:                    .WORD   0
                        BR      38$             ;TAKE ERROR-FREE EXIT
36$:                    MOV     (SP),REGNUM     ;RESTORE LU REG NO.
                        MOV     PSTACK,SP       ;RESTORE PROGRAM STACK TO BASE LEVEL
                        MOV     RETADR,-(SP)    ;FIX UP ERROR RETURN PC
                        BR      40$
38$:                    ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
                        MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
                        MOV     (SP)+,R1        ;RESTORE R1
40$:                    CLR     SUBRPC          ;CLEAR SUBROUTINE PC
                        RTS     PC              ;RETURN


;*********************************************************************************
;* LODATA - THIS SUBROUTINE LOADS THE TRANSMITTER SILO WITH 5 SOM'S, THE DATA
;*      IN PATTERN A REPEATED 2 TIMES (40 CHARS), AND 2 EOM'S. IN ADDITION, THE
;*      DATA CHARS ARE ALSO LOADED INTO THE RECEIVED MSG BUFFER (RCVBUF:), AS
;*      EXPECTED DATA FOR LATER COMPARISON.
;*********************************************************************************
LODATA: MOV     R1,-(SP)        ;SAVE R1
        MOV     R3,-(SP)        ;SAVE R3
        MOV     R4,-(SP)        ;SAVE R4
        JSR     PC,LODSIL       ;LOAD 5 SOM'S INTO TX SILO
        TXSOM
        5
        MOV     #2,R1           ;INIT COUNTER
        MOV     #RCVBUF,R4      ;GET POINTER TO RCV BUF
3$:     MOV     #PATA,R3        ;GET POINTER TO DATA PATTERN
6$:     CLR     TXWORD
        MOVB    (R3)+,TXWORD    ;GET A DATA CHAR
        MOV     TXWORD,(R4)+    ;LOAD A DATA CHAR INTO RCV BUF
        JSR     PC,LDTXSI       ;LOAD DATA CHAR INTO TX SILO
        CMP     R3,#PATB        ;SEE IF AT END OF PATTERN A YET
        BLO     6$              ;BR IF NOT YET
        DEC     R1              ;DECREMENT COUNTER
        BNE     3$              ;BR IF NOT DONE YET
        BIS     #CRCCHK!RXBCC,-2(R4) ;SET UP TO CHK BCC = 1 ON LAST DATA CHAR
        MOV     #TXEOM,TXWORD
        JSR     PC,LDTXSI       ;LOAD AN EOM INTO TX SILO
        JSR     PC,LDTXSI       ;LOAD ANOTHER EOM
        MOV     (SP)+,R4        ;RESTORE R4
        MOV     (SP)+,R3        ;RESTORE R3
        MOV     (SP)+,R1        ;RESTORE R1
        RTS     PC              ;RETURN
```

```
3929
3930
3931
3932
3933
3934                                    ;*************************************************************************
3935                                    ;* SETUP - THIS SUBROUTINE LOADS THE FIRST WORD AFTER THE CALL INTO AX2-15
3936                                    ;*       (SYNCH CHAR), LOADS THE SECOND WORD AFTER THE CALL INTO REG 17
3937                                    ;*       LOADS THE THIRD WORD INTO AX3-15, AND LOADS THE FOURTH WORD INTO AX3-16.
3938                                    ;*************************************************************************
3939  010312  013746  002354           SETUP:  MOV     AXNUM,-(SP)     ;SAVE AX BYTE NO.
3940  010316  013746  002352                   MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
3941  010322  012737  000004  002354           MOV     #4,AXNUM        ;SET AX BYTE NO. FOR AX2
3942  010330  017637  000004  002346           MOV     @4(SP),WAX15
3943  010336  005037  002350                   CLR     WAX16
3944  010342  004737  004000                   JSR     PC,WRITAX       ;SET SYNCH CHAR IN AX2-15, CLEAR AX2-16
3945  010346  012737  000017  002352           MOV     #17,REGNUM      ;SET LU REG NO. = 17
3946  010354  062766  000002  000004           ADD     #2,4(SP)        ;INCREMENT ARGUMENT POINTER
3947  010362  017637  000004  002340           MOV     @4(SP),WRIBYT
3948  010370  004737  003436                   JSR     PC,WRITLU       ;LOAD REG 17
3949  010374  012737  000006  002354           MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX3
3950  010402  062766  000002  000004           ADD     #2,4(SP)        ;INCREMENT ARGUMENT POINTER
3951  010410  017637  000004  002346           MOV     @4(SP),WAX15
3952  010416  062766  000002  000004           ADD     #2,4(SP)        ;INCREMENT ARGUMENT POINTER
3953  010424  017637  000004  002350           MOV     @4(SP),WAX16
3954  010432  013737  002350  002412           MOV     WAX16,SAVLEN    ;STORE TX AND RCV CHAR LENGTHS
3955  010440  142777  000010  171760           BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
3956  010446  004737  004000                   JSR     PC,WRITAX       ;LOAD AX3-15, AX3-16
3957  010452  152777  000010  171746           BISB    #LULOOP,@BSEL1  ;SET LULOOP
3958  010460  062766  000002  000004           ADD     #2,4(SP)        ;FIX RETURN PC
3959  010466  012637  002352                   MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
3960  010472  012637  002354                   MOV     (SP)+,AXNUM     ;RESTORE AX BYTE NO.
3961  010476  005037  002324                   CLR     SUBRPC          ;CLEAR SUBROUTINE PC STORAGE
3962  010502  000207                           RTS     PC              ;RETURN
3963
3964
3965
3966
3967
3968                                    ;*************************************************************************
3969                                    ;* LODMSG - THIS SUBROUTINE LOADS THE NO. OF WORDS PASSED IN THE SECOND WORD
3970                                    ;*       FOLLOWING THE CALL FROM THE MSG BUFFER WHOSE ADDRESS IS IN THE FIRST
3971                                    ;*       WORD FOLLOWING THE CALL, INTO THE TRANSMITTER SILO.
3972                                    ;*************************************************************************
3973  010504  010146                   LODMSG: MOV     R1,-(SP)        ;SAVE R1
3974  010506  010246                           MOV     R2,-(SP)        ;SAVE R2
3975  010510  017601  000004                   MOV     @4(SP),R1       ;GET MSG POINTER INTO R1
3976  010514  062766  000002  000004           ADD     #2,4(SP)        ;INCR ARG POINTER
3977  010522  017602  000004                   MOV     @4(SP),R2       ;GET WORD COUNT INTO R2
3978  010526  062766  000002  000004           ADD     #2,4(SP)        ;FIX UP RETURN PC
3979  010534  012137  002400           6$:     MOV     (R1)+,TXWORD    ;GET NEXT MSG WORD
3980  010540  004737  004662                   JSR     PC,LDTXSI       ;LOAD A WORD INTO TX SILO
3981  010544  005302                           DEC     R2              ;DECR COUNT
3982  010546  001372                           BNE     6$              ;BR IF NOT DONE YET
3983  010550  004737  004634                   JSR     PC,WAIT50       ;WAIT FOR SILO TO RIPPLE
3984  010554  012602                           MOV     (SP)+,R2        ;RESTORE R2
```

```
3985  010556  012601                        MOV     (SP)+,R1        ;RESTORE R1
3986  010560  000207                        RTS     PC              ;RETURN
3987
3988
3989
3990
3991
3992                                  ;*****************************************************************************
3993                                  ;* LDBYTS - THIS SUBROUTINE LOADS THE NO. OF BYTES PASSED IN THE SECOND WORD
3994                                  ;*       FOLLOWING THE CALL FROM THE MSG BUFFER WHOSE ADDRESS IS IN THE FIRST
3995                                  ;*       WORD FOLLOWING THE CALL, INTO THE LOW BYTE OF THE TX SILO. FOR EACH
3996                                  ;*       BYTE LOADED, A 0 IS LOADED INTO THE HI 4 BITS OF THE TX SILO.
3997                                  ;*****************************************************************************
3998  010562  010146                  LDBYTS: MOV     R1,-(SP)        ;SAVE R1
3999  010564  010246                          MOV     R2,-(SP)        ;SAVE R2
4000  010566  017601  000004                 MOV     @4(SP),R1       ;GET DATA POINTER INTO R1
4001  010572  062766  000002  000004         ADD     #2,4(SP)        ;INCR ARGUMENT POINTER
4002  010600  017602  000004                 MOV     @4(SP),R2       ;GET BYTE COUNT INTO R2
4003  010604  062766  000002  000004         ADD     #2,4(SP)        ;FIX UP RETURN PC
4004  010612  112137  002400          6$:    MOVB    (R1)+,TXWORD    ;GET NEXT DATA BYTE
4005  010616  105037  002401                 CLRB    TXWORD+1        ;CLEAR HI BYTE
4006  010622  004737  004662                 JSR     PC,LDTXSI       ;LOAD A SILO ENTRY
4007  010626  005302                         DEC     R2              ;DECR BYTE COUNT
4008  010630  001370                         BNE     6$              ;BR IF NOT DONE YET
4009  010632  004737  004634                 JSR     PC,WAIT50       ;WAIT FOR SILO TO RIPPLE
4010  010636  012602                         MOV     (SP)+,R2        ;RESTORE R2
4011  010640  012601                         MOV     (SP)+,R1        ;RESTORE R1
4012  010642  000207                         RTS     PC              ;RETURN
4013
4014
4015
4016
4017
4018                                  ;*****************************************************************************
4019                                  ;* LDMSG1 - THIS SUBROUTINE LOADS THE TRANSMITTER SILO WITH MSG1, AND LOADS
4020                                  ;*       THE DATA CHARS INTO THE RCV MSG BUFFER (RCVBUF:), AS EXPECTED DATA
4021                                  ;*       FOR LATER COMPARISON.
4022                                  ;*****************************************************************************
4023  010644  010146                  LDMSG1: MOV     R1,-(SP)        ;SAVE R1
4024  010646  010246                          MOV     R2,-(SP)        ;SAVE R2
4025  010650  004737  010734                 JSR     PC,LODSIL       ;LOAD 3 SOM'S INTO TX SILO
4026  010654  000400                         TXSOM
4027  010656  000003                         3
4028  010660  004737  010504                 JSR     PC,LODMSG       ;LOAD MSG1 INTO TX SILO (WITH 2 SOM'S, 1 EOM)
4029  010664  002654                         MSG1
4030  010666  000010                         8.
4031  010670  012701  002660                 MOV     #MSG1+4,R1      ;GET POINTER TO MSG1
4032  010674  012702  002746                 MOV     #RCVBUF,R2      ;GET POINTER TO MSG BUF
4033  010700  012122                  3$:    MOV     (R1)+,(R2)+     ;LOAD A CHAR INTO MSG BUF
4034  010702  020127  002672                 CMP     R1,#MSG1+14.    ;SEE IF DID LAST DATA CHAR YET
4035  010706  103774                         BLO     3$              ;BR IF NOT
4036  010710  052762  100400  177776         BIS     #CRCCHK!RXBCC,-2(R2) ;SET EXPECTED BCC
4037  010716  012722  000160                 MOV     #160,(R2)+      ;LOAD HI CRC BYTE
4038  010722  012722  000034                 MOV     #034,(R2)+      ;LOAD LO CRC BYTE
4039  010726  012602                         MOV     (SP)+,R2        ;RESTORE R2
4040  010730  012601                         MOV     (SP)+,R1        ;RESTORE R1
```

```
4041  010732  000207                          RTS     PC              ;RETURN
4042
4043
4044
4045
4046
4047                                  ;*************************************************************************************
4048                                  ;* LODSIL - THIS SUBROUTINE REPEATEDLY LOADS THE DATA PASSED IN THE FIRST WORD
4049                                  ;*       FOLLOWING THE CALL INTO THE TX SILO, FOR THE NO. OF TIMES PASSED IN THE
4050                                  ;*       SECOND WORD FOLLOWING THE CALL.
4051                                  ;*************************************************************************************
4052  010734  010146                  LODSIL: MOV     R1,-(SP)        ;SAVE R1
4053  010736  017637  000002  002400          MOV     @2(SP),TXWORD   ;GET DATA
4054  010744  062766  000002  000002          ADD     #2,2(SP)        ;INCR ARGUMENT POINTER
4055  010752  017601  000002                  MOV     @2(SP),R1       ;GET COUNT
4056  010756  004737  004662          3$:     JSR     PC,LDTXSI       ;LOAD WORD INTO TX SILO
4057  010762  005301                          DEC     R1              ;DECR COUNT
4058  010764  001374                          BNE     3$              ;BR IF NOT ALL LOADED YET
4059  010766  004737  004634                  JSR     PC,WAIT50       ;ALLOW SILO DATA TO RIPPLE
4060  010772  062766  000002  000002          ADD     #2,2(SP)        ;FIX UP RETURN PC
4061  011000  012601                          MOV     (SP)+,R1        ;RESTORE R1
4062  011002  000207                          RTS     PC              ;RETURN
4063
4064
4065
4066
4067
4068                                  ;*************************************************************************************
4069                                  ;* CKLPBK - THIS SUBROUTINE DETERMINES IF THE TEST CALLING IT CAN BE RUN. THE
4070                                  ;*       TEST PASSES THE DESIRED MODEM INTERFACE TYPE IN THE WORD FOLLOWING THE
4071                                  ;*       CALL, AND IF THE PROPER EXTERNAL LOOPBACK HAS BEEN PROVIDED BY THE
4072                                  ;*       OPERATOR FOR THAT INTERFACE, AND IF THE BAUD RATE IS CORRECT, A NORMAL
4073                                  ;*       RETURN IS MADE TO RUN THE TEST. IF NOT, A RETURN IS MADE TO THE TEST,
4074                                  ;*       AT THE ADDRESS IN RETADR (RETADR CONTAINS THE TEST EXIT ADDRESS, SO
4075                                  ;*       THE TEST GETS SKIPPED).
4076                                  ;*       IF BIT 15 IS SET IN THE WORD FOLLOWING THE CALL, THE TEST WILL NOT
4077                                  ;*       BE RUN UNLESS THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED. IF THE
4078                                  ;*       TEST IS TO BE RUN, THE SUBROUTINE RETURNS THE MODEM SELECT BITS FOR
4079                                  ;*       AX3-15 IN MODINT. IF NECESSARY, THE SUBROUTINE WILL SET MAINT1, MAINT2,
4080                                  ;*       OR DTR.
4081                                  ;*       IF THE PROGRAM PASSES '0' IN THE WORD FOLLOWING THE CALL, THE SUBROUTINE
4082                                  ;*       WILL ATTEMPT TO RUN WHICHEVER MODEM INTERFACE IS SELECTED BY CABLE,
4083                                  ;*       SWITCH, OR TEST CONNECTOR. IF SUCCESSFUL, THE SELECTED INTERFACE WILL
4084                                  ;*       BE PASSED BACK TO THE TEST IN MODINT.
4085                                  ;*************************************************************************************
4086  011004  013746  002352          CKLPBK: MOV     REGNUM,-(SP)    ;SAVE REG NO.
4087  011010  013746  002354                  MOV     AXNUM,-(SP)     ;SAVE AX BYTE NO.
4088  011014  010246                          MOV     R2,-(SP)        ;SAVE R2
4089  011016  016637  000006  002324          MOV     6(SP),SUBRPC
4090  011024  162737  000004  002324          SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
4091  011032  012737  000333  002410          MOV     #I422!XYZ!V35!INTGRL!OP!TEST,MODINT ;INIT MODEM SELECT BITS
4092  011040  032776  100000  000006          BIT     #TCCHEK,@6(SP)  ;SEE IF H3254,5 CHECK IS DESIRED
4093  011046  001405                          BEQ     1$              ;BR IF NOT
4094  011050  005737  002442                  TST     TSTCON          ;SEE IF H3254,5 INSTALLED
4095  011054  001064                          BNE     7$              ;BR IF NOT, TO SKIP TEST
4096  011056  000137  011706                  JMP     32$             ;BR TO RUN TEST
```

```
CZDMSF.P11    30-SEP-81 15:40          GLOBAL SUBROUTINES

 4097                                 ;IF NO EXTERNAL LPBK, SKIP TEST
 4098  011062  023727  002442  000004 1$:     CMP     TSTCON,#4       ;SEE IF NO LPBK
 4099  011070  001456                        BEQ     7$              ;BR IF NO LPBK, TO SKIP TEST
 4100  011072  042777  000010  171326        BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
 4101  011100  012737  000006  002354        MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX3-15
 4102  011106  004737  003612                JSR     PC,READAX       ;READ AX3-15
 4103
 4104                                 ;*** SEE IF AN INTERFACE IS REQUESTED ***
 4105
 4106  011112  027627  000006  000010        CMP     @6(SP),#INTGRL  ;SEE IF INTEGRAL MODEM REQUESTED
 4107  011120  001422                        BEQ     4$              ;BR IF INTGRL MODEM REQUESTED
 4108  011122  027627  000006  000020        CMP     @6(SP),#V35     ;SEE IF V.35 REQUESTED
 4109  011130  001502                        BEQ     10$             ;BR IF V.35 REQUESTED
 4110  011132  027627  000006  000200        CMP     @6(SP),#I422    ;SEE IF 422 REQUESTED
 4111  011140  001002                        BNE     2$              ;BR IF 422 NOT REQUESTED
 4112  011142  000137  011422                JMP     14$             ;422 REQUESTED
 4113  011146  027627  000006  000100 2$:     CMP     @6(SP),#XYZ     ;SEE IF XYZ REQUESTED
 4114  011154  001002                        BNE     3$              ;BR IF XYZ NOT REQUESTED
 4115  011156  000137  011456                JMP     17$             ;XYZ REQUESTED
 4116  011162  000137  011550        3$:     JMP     21$             ;NONE REQUESTED, FIND AN INTERFACE TO TEST
 4117
 4118                                 ;SEE IF INTEGRAL MODEM CAN BE RUN
 4119  011166  005737  002442        4$:     TST     TSTCON          ;SEE IF H3254 AND H3255 USED
 4120  011172  001040                        BNE     8$              ;BR IF NOT
 4121  011174  023727  002446  000004 5$:     CMP     BDRATE,#4       ;SEE IF BAUD RATE > OR = 56K
 4122  011202  002405                        BLT     6$              ;BR IF BAUD RATE TOO SLOW FOR INTGRL MODM
 4123  011204  042737  000010  002410        BIC     #INTGRL,MODINT  ;ASSERT INTEGRAL MODEM
 4124  011212  000137  011706                JMP     32$             ;GO TO RUN TEST
 4125  011216                         6$:
 4126  011216  023727  002370  000001        CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
 4127  011224  001016                        BNE     40$             ;BR IF NOT, TO SKIP PRINTING
 4128                                 ;PRINT 'TEST XX NOT RUN"
 4129  011226                         7$:
 4130  011226  023727  002370  000001        CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
 4131  011234  001012                        BNE     40$             ;BR IF NOT, TO SKIP PRINTING
 4132  011236                                PRINTF  #FMT19,TSTNUM
 4133  011236  013746  002422                                                MOV     TSTNUM,-(SP)
 4134  011242  012746  013367                                                MOV     #FMT19,-(SP)
 4135  011246  012746  000002                                                MOV     #2,-(SP)
 4136  011252  010600                                                        MOV     SP,R0
 4137  011254  104417                                                        TRAP    C$PNTF
 4138  011256  062706  000006                                                ADD     #6,SP
 4139  011262  013766  002334  000006 40$:     MOV     RETADR,6(SP)    ;SET TEST EXIT ADDRESS FOR ERRORS
 4140  011270  000137  011714                JMP     33$             ;GO TO SKIP TEST
 4141  011274  032737  000010  002342 8$:     BIT     #INTGRL,RAX15   ;SEE IF INTEGRAL MODEM IS SELECTED
 4142  011302  001334                        BNE     5$              ;BR IF YES, TO CHECK BAUD RATE
 4143                                 ;PRINT 'MODEM INTERFACE NOT SELECTED"
 4144  011304                         9$:
 4145  011304  023727  002370  000001        CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
 4146  011312  001363                        BNE     40$             ;BR IF NOT, TO SKIP PRINTING
 4147  011314                                PRINTF  #FMT26
 4148  011314  012746  013420                                                MOV     #FMT26,-(SP)
 4149  011320  012746  000001                                                MOV     #1,-(SP)
 4150  011324  010600                                                        MOV     SP,R0
 4151  011326  104417                                                        TRAP    C$PNTF
 4152  011330  062706  000004                                                ADD     #4,SP
```

```
 4153  011334  000734                          BR      7$              ;GO TO PRINT "TEST NOT RUN"
 4154
 4155                                  ;SEE IF V.35 CAN BE RUN
 4156  011336  032737  000200  002342  10$:    BIT     #I422,RAX15     ;SEE IF 422 IS SELECTED
 4157  011344  001357                          BNE     9$              ;BR IF YES, TO SKIP TEST
 4158  011346  005737  002442                  TST     TSTCON          ;SEE IF H3254 AND H3255 USED
 4159  011352  001010                          BNE     12$             ;BR IF H3254 AND H3255 NOT USED
 4160  011354  012737  000004  002340  11$:    MOV     #MAINT2,WRIBYT  ;SET MAINT2 FOR MANUFACTURING TEST CONN.
 4161  011362  042737  000020  002410          BIC     #V35,MODINT     ;ASSERT V.35
 4162  011370  000137  011652                  JMP     42$             ;GO SET DTR AND RUN THE TEST
 4163  011374  032737  000020  002342  12$:    BIT     #V35,RAX15      ;SEE IF V.35 IS SELECTED
 4164  011402  001405                          BEQ     13$             ;BR IF NOT
 4165  011404  042737  000020  002410          BIC     #V35,MODINT     ;ASSERT V.35
 4166  011412  000137  011646                  JMP     27$             ;GO SET DTR AND RUN THE TEST
 4167  011416  000137  011304          13$:    JMP     9$              ;WRONG INTRFCE, GO SKIP TEST
 4168
 4169                                  ;SEE IF 422 CAN BE RUN
 4170  011422  005737  002442          14$:    TST     TSTCON          ;SEE IF H3254 AND H3255 USED
 4171  011426  001005                          BNE     16$             ;BR IF NOT
 4172  011430  042737  000200  002410  15$:    BIC     #I422,MODINT    ;ASSERT 422
 4173  011436  000137  011646                  JMP     27$             ;GO TO RUN TEST
 4174  011442  032737  000200  002342  16$:    BIT     #I422,RAX15     ;SEE IF 422 IS SELECTED
 4175  011450  001367                          BNE     15$             ;IF YES, GO ASSERT 422 AND RUN TEST
 4176  011452  000137  011304                  JMP     9$              ;WRONG INTRFCE, GO SKIP TEST
 4177
 4178                                  ;SEE IF XYZ CAN BE RUN
 4179  011456  032737  000200  002342  17$:    BIT     #I422,RAX15     ;SEE IF 422 IS SELECTED
 4180  011464  001402                          BEQ     18$             ;BR IF NOT
 4181  011466  000137  011304                  JMP     9$              ;WRONG INTRFCE, GO SKIP TEST
 4182  011472  032737  000100  002342  18$:    BIT     #XYZ,RAX15      ;SEE IF XYZ IS SELECTED
 4183  011500  001002                          BNE     19$             ;BR IF YES
 4184  011502  000137  011304                  JMP     9$              ;WRONG INTRFCE, GO SKIP TEST
 4185  011506  023727  002446  000004  19$:    CMP     BDRATE,#4       ;SEE IF BAUD RATE < OR = 56K
 4186  011514  003402                          BLE     20$             ;BR IF YES
 4187  011516  000137  011216                  JMP     6$              ;BAUD RATE TOO FAST FOR XYZ
 4188  011522  042737  000100  002410  20$:    BIC     #XYZ,MODINT     ;ASSERT XYZ
 4189  011530  005737  002442                  TST     TSTCON          ;SEE IF H3254,5 BEING USED
 4190  011534  001044                          BNE     27$             ;BR IF NOT
 4191  011536  012737  000004  002340          MOV     #MAINT2,WRIBYT
 4192  011544  000137  011652                  JMP     42$             ;GO SET MAINT2 FOR MANUFACTURING TEST CONN.
 4193
 4194                                  ;*** NO INTERFACE REQUESTED - FIND ONE TO TEST ***
 4195
 4196  011550  032737  000010  002342  21$:    BIT     #INTGRL,RAX15   ;SEE IF INTEGRAL MODEM SELECTED
 4197  011556  001402                          BEQ     22$             ;BR IF NOT
 4198  011560  000137  011174                  JMP     5$              ;SEE IF INTEGRAL MODEM CAN BE RUN
 4199  011564  032737  000020  002342  22$:    BIT     #V35,RAX15      ;SEE IF V.35 SELECTED
 4200  011572  001402                          BEQ     23$             ;BR IF NOT
 4201  011574  000137  011336                  JMP     10$             ;GO SEE IF V.35 CAN BE RUN
 4202  011600  032737  000200  002342  23$:    BIT     #I422,RAX15     ;SEE IF 422 SELECTED
 4203  011606  001402                          BEQ     24$             ;BR IF NOT
 4204  011610  000137  011430                  JMP     15$             ;GO ASSERT AND RUN 422
 4205  011614  005737  002442          24$:    TST     TSTCON          ;SEE IF H3254 AND H3255 USED
 4206  011620  001002                          BNE     25$             ;BR IF NOT
 4207  011622  000137  011354                  JMP     11$             ;GO ASSERT AND RUN V.35 BY DEFAULT
 4208  011626  032737  000100  002342  25$:    BIT     #XYZ,RAX15      ;SEE IF XYZ SELECTED
```

```
4209  011634  001402                          BEQ     26$             ;BR IF NOT
4210  011636  000137  011506                  JMP     19$             ;GO SEE IF XYZ CAN BE RUN
4211  011642  000137  011304          26$:    JMP     9$              ;GO SKIP TEST
4212
4213  011646  005037  002340          27$:    CLR     WRIBYT          ;INITIALIZE WRITBYT FOR WRITE TO SET DTR
4214  011652  012737  000013  002352  42$:    MOV     #13,REGNUM      ;SET REG NO. = 13
4215  011660  153737  002450  002340          BISB    MLWBYT,WRIBYT   ;SET MAINT1/2 BIT IF MODEM LOOPBACK
4216
4217                                  ;SET DTR
4218  011666  012737  000013  002352          MOV     #13,REGNUM      ;SET REG NO. = 13
4219  011674  152737  000100  002340          BISB    #DTR,WRIBYT     ;SET DTR BIT TO BE WRITTEN
4220  011702  004737  003436                  JSR     PC,WRITLU       ;LOAD REG 13
4221
4222                                  ;*** BRANCH HERE TO RUN TEST ***
4223  011706  062766  000002  000006  32$:    ADD     #2,6(SP)        ;INCREMENT RETURN ADRS
4224
4225                                  ;*** BRANCH HERE TO SKIP TEST ***
4226  011714  012602                  33$:    MOV     (SP)+,R2        ;RESTORE R2
4227  011716  012637  002354                  MOV     (SP)+,AXNUM     ;RESTORE AX BYTE NO.
4228  011722  012637  002352                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
4229  011726  152777  000010  170472          BISB    #LULOOP,@BSEL1  ;SET LULOOP
4230  011734  005037  002324                  CLR     SUBRPC          ;CLEAR SUBROUTINE CALL PC
4231  011740  000207                          RTS     PC              ;RETURN
4232
4233
4234
4235
4236
4237                    ;*********************************************************************************
4238                    ;* CHKABT - THIS SUBROUTINE READS AX0-16 AND CHECKS FOR RAB, REOM SET. IF
4239                    ;*      EITHER IS NOT SET, A RETURN IS MADE TO THE TEST, AT THE ADDRESS
4240                    ;*      CONTAINED IN RETADR.
4241                    ;*********************************************************************************
4242  011742  013746  002354  CHKABT: MOV     AXNUM,-(SP)     ;SAVE AX BYTE NO.
4243  011746  016637  000002  002324          MOV     2(SP),SUBRPC
4244  011754  162737  000004  002324          SUB     #4,SUBRPC       ;GET PC OF SUBROUTINE CALL
4245  011762  005037  002354                  CLR     AXNUM           ;SET AX0 ADDRESS
4246  011766  004737  003612                  JSR     PC,READAX       ;READ REG AX0
4247  011772  032737  000004  002344          BIT     #RABT,RAX16     ;CHK FOR RAB SET
4248  012000  001007                          BNE     6$              ;BR IF RAB SET
4249  012002  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4250                                  ;REPORT RAB NOT SET
4251  012006                                  ERRDF   40,EM40,ERR6
4252  012006  104455                                                                  TRAP    C$ERDF
4253  012010  000050                                                                  .WORD   40
4254  012012  014203                                                                  .WORD   EM40
4255  012014  017664                                                                  .WORD   ERR6
4256  012016  000412                          BR      8$
4257  012020  032737  000002  002344  6$:     BIT     #REOM,RAX16     ;CHK FOR REOM SET
4258  012026  001015                          BNE     9$              ;BR IF REOM SET
4259  012030  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4260                                  ;REPORT REOM NOT SET
4261  012034                                  ERRDF   31,EM31,ERR6
4262  012034  104455                                                                  TRAP    C$ERDF
4263  012036  000037                                                                  .WORD   31
4264  012040  014042                                                                  .WORD   EM31
```

```
012042  017664                                                                  .WORD   ERR6
012044  011637  002354          8$:     MOV     (SP),AXNUM      ;RESTORE AX BYTE NO.
012050  013706  002320                  MOV     PSTACK,SP       ;RESTORE STACK POINTER TO BASE LEVEL
012054  013746  002334                  MOV     RETADR,-(SP)    ;FIX ERROR RETURN PC
012060  000402                          BR      16$
012062  012637  002354          9$:     MOV     (SP)+,AXNUM     ;RESTORE AX BYTE NO.
012066  000207                  16$:    RTS     PC              ;RETURN




                                ;*******************************************************************************
                                ;* SETMTM - IF MODEM LOOPBACK IS REQUESTED, THIS SUBROUTINE WAITS FOR TESTMD TO
                                ;*         BE SET.  IF REMOTE MODEM LOOPBACK IS BEING USED, THIS ROUTINE LOOPS
                                ;*         INDEFINITELY UNTIL THE PHONE LINK IS ESTABLISHED.
                                ;*******************************************************************************
012070  133727  002442  000002  SETMTM: BITB    TSTCON,#2       ;TEST FOR LOCAL OR REMOTE MOD LOOPBACK
012076  001510                          BEQ     2$              ;EXIT SUBROUTINE IF NOT MOD LOOPBACK
012100  142777  000010  170320          BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP, ENABLE MSG CLOCK
012106  011637  002324                  MOV     (SP),SUBRPC
012112  162737  000004  002324          SUB     #4,SUBRPC       ;GET PC OF SUBROUTINE CALL
012120  012737  000017  002352          MOV     #17,REGNUM      ;SET REG TO 17 FOR TESTMODE DETECTION
012126  123727  002442  000003          CMPB    TSTCON,#3       ;TEST FOR REMOTE MODEM LOOPBACK
012134  001425                          BEQ     8$              ;BR IF REMOTE MODEM LOOPBACK
012136  012702  144444                  MOV     #144444,R2      ;INITIALIZE COUNTER FOR 3 SEC WAIT
012142  004737  003360          10$:    JSR     PC,READLU       ;GET REG 17 TO TEST FOR TESTMODE SET
012146  132737  000004  002336          BITB    #TESTMD,REDBYT  ;TEST FOR TESTMD SET
012154  001061                          BNE     2$              ;BR IF TESTMD SET, TO EXIT SUBROUTINE
012156  004737  004634                  JSR     PC,WAIT50       ;DELAY 50 MICRO-SEC
012162  005202                          INC     R2              ;INC COUNTER
012164  001366                          BNE     10$             ;BR IF TIME NOT UP
012166  004737  004214                  JSR     PC,GETALL       ;GET REG CONTENTS FOR ERROR MESSAGE
                                ;REPORT TESTMODE NOT SET BY MAINT1
012172                                  ERRDF   52,EM52,ERR4
012172  104455                                                                  TRAP    C$ERDF
012174  000064                                                                  .WORD   52
012176  014633                                                                  .WORD   EM52
012200  016474                                                                  .WORD   ERR4
012202  013716  002334                  MOV     RETADR,(SP)     ;SKIP TEST BECAUSE OF ERROR
012206  000444                          BR      2$              ;EXIT SUBROUTINE

012210  004737  003360          8$:     JSR     PC,READLU       ;GET REG 17 CONTENTS
012214  132737  000004  002336          BITB    #TESTMD,REDBYT  ;TEST FOR TESTMD SET
012222  001036                          BNE     2$              ;EXIT IF TESTMD SET
012224                                  PRINTF  #LM1            ;PRINT "ESTABLISH PHONE LINK"
012224  012746  012326                                                          MOV     #LM1,-(SP)
012230  012746  000001                                                          MOV     #1,-(SP)
012234  010600                                                                  MOV     SP,R0
012236  104417                                                                  TRAP    C$PNTF
012240  062706  000004                                                          ADD     #4,SP
012244  012702  000000                  MOV     #0,R2           ;INITIALIZE TIMING COUNTER
012250  004737  003360          14$:    JSR     PC,READLU       ;GET REG 17 CONTENTS
012254  132737  000004  002336          BITB    #TESTMD,REDBYT  ;TEST FOR TESTMD SET
012262  001006                          BNE     12$             ;BR IF TESTMD SET
012264                                  BREAK                   ;ALLOW CONTROL-C TO BE RECOGNIZED IN LOOP
012264  104422                                                                  TRAP    C$BRK
```

```
4321  012266  004737  004634          JSR     PC,WAIT50       ;DELAY 50 MICRO-SEC
4322  012272  005202                  INC     R2              ;INC TIME COUNTER
4323  012274  001365                  BNE     14$             ;BR IF TIME NOT UP
4324  012276  000744                  BR      8$              ;GO BACK AND PRINT 'ESTABLISH PHONE LINK'' AGAIN
4325  012300                  12$:    PRINTF  #LM2            ;PRINT 'PHONE LINK ESTABLISHED''
4326  012300  012746  012357                                                  MOV     #LM2,-(SP)
4327  012304  012746  000001                                                  MOV     #1,-(SP)
4328  012310  010600                                                          MOV     SP,R0
4329  012312  104417                                                          TRAP    C$PNTF
4330  012314  062706  000004                                                  ADD     #4,SP
4331
4332  012320  005037  002324          2$:     CLR     SUBRPC          ;REMOVE SUBROUTINE CALL PC FROM STORAGE AREA
4333  012324  000207                  RTS     PC
4334
4335  012326  047045  040445  051505  LM1:    .ASCIZ /%N%AESTABLISH PHONE LINK/
4336  012334  040524  046102  051511
4337  012342  020110  044120  047117
4338  012350  020105  044514  045516
4339  012356     000
4340  012357     045  022516  050101  LM2:    .ASCIZ /%N%APHONE LINK ESTABLISHED/
4341  012364  047510  042516  046040
4342  012372  047111  020113  051505
4343  012400  040524  046102  051511
4344  012406  042510  000104
4345
4346
4347                                  ;*************************************************************************
4348                                  ;* TSTMLB - THIS SUBROUTINE TESTS FOR LOCAL OR REMOTE MODEM LOOPBACK. IF EITHER
4349                                  ;*      TYPE OF MODEM LOOPBACK IS BEING USED IT CHECKS FOR CS BEING SET.
4350                                  ;*************************************************************************
4351  012412  133727  002442  000002  TSTMLB: BITB    TSTCON,#2       ;TEST FOR LOC OR REM MODEM LOOPBACK
4352  012420  001004                  BNE     2$              ;BR TO START ROUTINE IF MODEM LOOPBACK
4353  012422  142777  000010  167776          BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP, ENABLE MSG CLOCK
4354  012430  000434                  BR      4$              ;SKIP SUBROUTINE IF NON-MODEM LOOPBACK
4355  012432  011637  002324          2$:     MOV     (SP),SUBRPC
4356  012436  162737  000004  002324          SUB     #4,SUBRPC       ;GET PC OF SUBROUTINE CALL
4357  012444  012737  000013  002352          MOV     #13,REGNUM      ;PREPARE FOR TEST FOR CS SET
4358  012452  012702  144444                  MOV     #144444,R2      ;INITIALIZE FOR 3 SEC WAIT
4359  012456  004737  003360          6$:     JSR     PC,READLU       ;GET REG 13 CONTENTS FOR TEST FOR CS SET
4360  012462  132737  000004  002336          BITB    #CS,REDBYT      ;TEST FOR CS SET
4361  012470  001014                  BNE     4$              ;BR IF CS SET
4362  012472  004737  004634          JSR     PC,WAIT50       ;DELAY 50 MICRO-SEC
4363  012476  005202                  INC     R2              ;INC COUNTER
4364  012500  001366                  BNE     6$              ;BR IF TIME NOT UP
4365  012502  004737  004214          JSR     PC,GETALL       ;GET REG CONTENTS FOR ERROR CALL
4366                          ;REPORT CS NOT SET
4367  012506                  ERRDF   70,EM61,ERR4
4368  012506  104455                                                          TRAP    C$ERDF
4369  012510  000106                                                          .WORD   70
4370  012512  015024                                                          .WORD   EM61
4371  012514  016474                                                          .WORD   ERR4
4372  012516  013716  002334          MOV     RETADR,(SP)     ;SKIP TEST BECAUSE OF ERROR
4373  012522  005037  002324          4$:     CLR     SUBRPC          ;CLEAR SUB CALL PC FROM STORAGE
4374  012526  000207                  RTS     PC
4375
4376
```

```
4377
4378
```

```
CZDMSF.P11    30-SEP-81 15:40          GLOBAL ERROR REPORT SECTION

                                       .SBTTL  GLOBAL ERROR REPORT SECTION

                                       ;////////////////////////////////////////////////////////////////
                                       ;/      THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
                                       ;/      THAT ARE USED IN MORE THAN ONE TEST.
                                       ;////////////////////////////////////////////////////////////////

 012530  052045  047445  022466  FMT1:   .ASCIZ  /%T%06%N/
 012536  000116
 012540  047045  040445  040506  FMT2:   .ASCIZ  /%N%AFAILING REG: /
 012546  046111  047111  020107
 012554  042522  035107  000040
 012562  040445  054105  042520  FMT3:   .ASCIZ  /%AEXPECTED: %03%S5%AACTUAL: %03%N/
 012570  052103  042105  020072
 012576  047445  022463  032523
 012604  040445  041501  052524
 012612  046101  020072  047445
 012620  022463  000116
 012624  047045  052045  047045  FMT4:   .ASCIZ  /%N%T%N%T%N/
 012632  052045  047045     000
 012637     045  031517  051445  FMT5:   .ASCIZ  /%03%S5%03%S5%03%S5%03%N/
 012644  022465  031517  051445
 012652  022465  031517  051445
 012660  022465  031517  047045
 012666     000
 012667     045  032123  047445  FMT6:   .ASCIZ  /%S4%03%S5%03%S5%03%S5%03%N/
 012674  022463  032523  047445
 012702  022463  032523  047445
 012710  022463  032523  047445
 012716  022463  000116
 012722  052045  047445  022462  FMT7:   .ASCIZ  /%T%02%N/
 012730  000116
 012732  040445  054105  042524  FMT8:   .ASCIZ  /%AEXTENDED REG AX%01%A-%T%N/
 012740  042116  042105  051040
 012746  043505  040440  022530
 012754  030517  040445  022455
 012762  022524  000116
 012766  052045  047045     000  FMT9:   .ASCIZ  /%T%N/
 012773     045  050101  020103  FMT10:  .ASCIZ  /%APC OF SUBR CALL: %06%N/
 013000  043117  051440  041125
 013006  020122  040503  046114
 013014  020072  047445  022466
 013022  000116
 013024  040445  042522  020107  FMT11:  .ASCIZ  /%AREG %02%A LOADED WITH: %03%N/
 013032  047445  022462  020101
 013040  047514  042101  042105
 013046  053440  052111  035110
 013054  022440  031517  047045
 013062     000
 013063     045  046501  031070  FMT12:  .ASCIZ  /%AM8203 REG 11 (E134 SW10,9 , E121 SW9,10) = %03%N/
 013070  031460  051040  043505
 013076  030440  020061  042450
 013104  031461  020064  053523
 013112  030061  034454  026040
 013120  042440  031061  020061
```

```
 4435  013126  053523  026071  030061
 4436  013134  020051  020075  047445
 4437  013142  022463  000116
 4438  013146  040445  034115  030062  FMT13:  .ASCIZ  /%AM8203 REG 15 (E134 SW8-1) = %03%N/
 4439  013154  020063  042522  020107
 4440  013162  032461  024040  030505
 4441  013170  032063  051440  034127
 4442  013176  030455  020051  020075
 4443  013204  047445  022463  000116
 4444  013212  040445  034115  030062  FMT14:  .ASCIZ  /%AM8203 REG 16 (E121 SW8-1) = %03%N/
 4445  013220  020063  042522  020107
 4446  013226  033061  024040  030505
 4447  013234  030462  051440  034127
 4448  013242  030455  020051  020075
 4449  013250  047445  022463  000116
 4450  013256  040445  047515  042504  FMT15:  .ASCIZ  /%AMODEM INTERFACE REG (AX3-15) = %03%N/
 4451  013264  020115  047111  042524
 4452  013272  043122  041501  020105
 4453  013300  042522  020107  040450
 4454  013306  031530  030455  024465
 4455  013314  036440  022440  031517
 4456  013322  047045     000
 4457  013325     045  022516  043101  FMT18:  .ASCIZ  /%N%AFOR DEVICE AT ADRS  %06%A .%N/
 4458  013332  051117  042040  053105
 4459  013340  041511  020105  052101
 4460  013346  040440  051104  020123
 4461  013354  022440  033117  040445
 4462  013362  026040  047045     000
 4463  013367     045  022516  052101  FMT19:  .ASCIZ  /%N%ATEST %D2%A NOT RUN%N/
 4464  013374  051505  020124  042045
 4465  013402  022462  020101  047516
 4466  013410  020124  052522  022516
 4467  013416  000116
 4468  013420  047045  040445  047515  FMT26:  .ASCIZ  /%N%AMODEM INTERFACE NOT SELECTED/
 4469  013426  042504  020115  047111
 4470  013434  042524  043122  041501
 4471  013442  020105  047516  020124
 4472  013450  042523  042514  052103
 4473  013456  042105     000
 4474
 4475
 4476
 4477  013461     122  043505  047040  EM2:    .ASCIZ  /REG NOT INITIALIZED BY MST CLR/
 4478  013466  052117  044440  044516
 4479  013474  044524  046101  055111
 4480  013502  042105  041040  020131
 4481  013510  051515  020124  046103
 4482  013516  000122
 4483  013520  042522  020107  044515  EM3:    .ASCIZ  /REG MISCOMPARE/
 4484  013526  041523  046517  040520
 4485  013534  042522     000
 4486  013537     117  042122  020131  EM7:    .ASCIZ  /ORDY NOT SET/
 4487  013544  047516  020124  042523
 4488  013552  000124
 4489  013554  051117  054504  047040  EM8:    .ASCIZ  /ORDY NOT CLR/
 4490  013562  052117  041440  051114
```

```
4491  013570     000
4492  013571     117  047503  020122  EM9:    .ASCIZ  /OCOR NOT SET/
4493  013576  047516  020124  042523
4494  013604  000124
4495  013606  041517  051117  047040  EM10:   .ASCIZ  /OCOR NOT CLR/
4496  013614  052117  041440  051114
4497  013622     000
4498  013623     117  041501  020124  EM11:   .ASCIZ  /OACT NOT SET/
4499  013630  047516  020124  042523
4500  013636  000124
4501  013640  040517  052103  047040  EM12:   .ASCIZ  /OACT NOT CLR/
4502  013646  052117  041440  051114
4503  013654     000
4504  013655     111  042122  020131  EM17:   .ASCIZ  /IRDY NOT SET/
4505  013662  047516  020124  042523
4506  013670  000124
4507  013672  051111  054504  047040  EM18:   .ASCIZ  /IRDY NOT CLR/
4508  013700  052117  041440  051114
4509  013706     000
4510  013707     111  044503  020122  EM19:   .ASCIZ  /ICIR NOT SET/
4511  013714  047516  020124  042523
4512  013722  000124
4513  013724  041511  051111  047040  EM20:   .ASCIZ  /ICIR NOT CLR/
4514  013732  052117  041440  051114
4515  013740     000
4516  013741     111  041501  020124  EM21:   .ASCIZ  /IACT NOT SET/
4517  013746  047516  020124  042523
4518  013754  000124
4519  013756  040511  052103  047040  EM22:   .ASCIZ  /IACT NOT CLR/
4520  013764  052117  041440  051114
4521  013772     000
4522  013773     122  047523  020115  EM28:   .ASCIZ  /RSOM NOT CLR/
4523  014000  047516  020124  046103
4524  014006  000122
4525  014010  051522  046517  047040  EM29:   .ASCIZ  /RSOM NOT SET/
4526  014016  052117  051440  052105
4527  014024     000
4528  014025     122  047505  020115  EM30:   .ASCIZ  /REOM NOT CLR/
4529  014032  047516  020124  046103
4530  014040  000122
4531  014042  042522  046517  047040  EM31:   .ASCIZ  /REOM NOT SET/
4532  014050  052117  051440  052105
4533  014056     000
4534  014057     122  053103  042047  EM34:   .ASCIZ  /RCV'D DATA MISCOMPARE/
4535  014064  042040  052101  020101
4536  014072  044515  041523  046517
4537  014100  040520  042522     000
4538  014105     102  041503  047040  EM35:   .ASCIZ  /BCC NOT CLR/
4539  014112  052117  041440  051114
4540  014120     000
4541  014121     102  041503  047040  EM36:   .ASCIZ  /BCC NOT SET/
4542  014126  052117  051440  052105
4543  014134     000
4544  014135     105  046102  020113  EM37:   .ASCIZ  /EBLK NOT CLR/
4545  014142  047516  020124  046103
4546  014150  000122
```

```
4547  014152  041105  045514  047040  EM38:   .ASCIZ  /EBLK NOT SET/
4548  014160  052117  051440  052105
4549  014166     000
4550  014167     122  041101  047040  EM39:   .ASCIZ  /RAB NOT CLR/
4551  014174  052117  041440  051114
4552  014202     000
4553  014203     122  041101  047040  EM40:   .ASCIZ  /RAB NOT SET/
4554  014210  052117  051440  052105
4555  014216     000
4556  014217     117  051126  020122  EM41:   .ASCIZ  /OVRR NOT CLR/
4557  014224  047516  020124  046103
4558  014232  000122
4559  014234  053117  051122  047040  EM42:   .ASCIZ  /OVRR NOT SET/
4560  014242  052117  051440  052105
4561  014250     000
4562  014251     122  043505  030440  EM43:   .ASCIZ  /REG 11 SWITCHES INCORRECT/
4563  014256  020061  053523  052111
4564  014264  044103  051505  044440
4565  014272  041516  051117  042522
4566  014300  052103     000
4567  014303     122  043505  030440  EM44:   .ASCIZ  /REG 15 SWITCHES INCORRECT/
4568  014310  020065  053523  052111
4569  014316  044103  051505  044440
4570  014324  041516  051117  042522
4571  014332  052103     000
4572  014335     122  043505  030440  EM45:   .ASCIZ  /REG 16 SWITCHES INCORRECT/
4573  014342  020066  053523  052111
4574  014350  044103  051505  044440
4575  014356  041516  051117  042522
4576  014364  052103     000
4577  014367     122  053103  051440  EM46:   .ASCIZ  /RCV SILO NOT CLEARED BY IC/
4578  014374  046111  020117  047516
4579  014402  020124  046103  040505
4580  014410  042522  020104  054502
4581  014416  044440  000103
4582  014422  051501  042523  041115  EM47:   .ASCIZ  /ASSEMB BIT COUNT INCORRECT/
4583  014430  041040  052111  041440
4584  014436  052517  052116  044440
4585  014444  041516  051117  042522
4586  014452  052103     000
4587  014455     117  042104  053040  EM48:   .ASCIZ  /ODD VRC PARITY BIT NOT SET/
4588  014462  041522  050040  051101
4589  014470  052111  020131  044502
4590  014476  020124  047516  020124
4591  014504  042523  000124
4592  014510  042117  020104  051126  EM49:   .ASCIZ  /ODD VRC PARITY BIT NOT CLR/
4593  014516  020103  040520  044522
4594  014524  054524  041040  052111
4595  014532  047040  052117  041440
4596  014540  051114     000
4597  014543     105  042526  020116  EM50:   .ASCIZ  /EVEN VRC PARITY BIT NOT SET/
4598  014550  051126  020103  040520
4599  014556  044522  054524  041040
4600  014564  052111  047040  052117
4601  014572  051440  052105     000
4602  014577     105  042526  020116  EM51:   .ASCIZ  /EVEN VRC PARITY BIT NOT CLR/
```

```
 4603  014604  051126  020103  040520
 4604  014612  044522  054524  041040
 4605  014620  052111  047040  052117
 4606  014626  041440  051114     000
 4607  014633     124  051505  020124  EM52:   .ASCIZ  /TEST MODE NOT SET BY MAINT1/
 4608  014640  047515  042504  047040
 4609  014646  052117  051440  052105
 4610  014654  041040  020131  040515
 4611  014662  047111  030524     000
 4612  014667     124  020130  047125  EM54:   .ASCIZ  /TX UNDERRUN ERROR/
 4613  014674  042504  051122  047125
 4614  014702  042440  051122  051117
 4615  014710     000
 4616  014711     104  051124  047040  EM55:   .ASCIZ  /DTR NOT SET/
 4617  014716  052117  051440  052105
 4618  014724     000
 4619  014725     122  047111  020107  EM56:   .ASCIZ  /RING NOT SET/
 4620  014732  047516  020124  042523
 4621  014740  000124
 4622  014742  047515  051104  047040  EM57:   .ASCIZ  /MODR NOT SET/
 4623  014750  052117  051440  052105
 4624  014756     000
 4625  014757     110  054104  047040  EM58:   .ASCIZ  /HDX NOT SET/
 4626  014764  052117  051440  052105
 4627  014772     000
 4628  014773     123  041124  020131  EM59:   .ASCIZ  /STBY NOT SET/
 4629  015000  047516  020124  042523
 4630  015006  000124
 4631  015010  052122  020123  047516  EM60:   .ASCIZ  /RTS NOT SET/
 4632  015016  020124  042523  000124
 4633  015024  051503  047040  052117  EM61:   .ASCIZ  /CS NOT SET/
 4634  015032  051440  052105     000
 4635  015037     103  051101  020122  EM62:   .ASCIZ  /CARR NOT SET/
 4636  015044  047516  020124  042523
 4637  015052  000124
 4638  015054  044523  050507  047040  EM63:   .ASCIZ  /SIGQ NOT SET/
 4639  015062  052117  051440  052105
 4640  015070     000
 4641  015071     123  043511  020122  EM64:   .ASCIZ  /SIGR NOT SET/
 4642  015076  047516  020124  042523
 4643  015104  000124
 4644  015106  052122  020123  047516  EM65:   .ASCIZ  /RTS NOT CLR/
 4645  015114  020124  046103  000122
 4646  015122  040503  051122  047040  EM66:   .ASCIZ  /CARR NOT CLR/
 4647  015130  052117  041440  051114
 4648  015136     000
 4649
 4650
 4651
 4652  015137     111  041116  051525  DH1:    .ASCIZ  &INBUS/OUTBUS REG &
 4653  015144  047457  052125  052502
 4654  015152  020123  042522  020107
 4655  015160     000
 4656  015161     114  047111  020105  DH2:    .ASCIZ  /LINE UNIT INBUS REGS :/
 4657  015166  047125  052111  044440
 4658  015174  041116  051525  051040
```

```
4659  015202  043505  020123  000072
4660  015210  042522  030507  020060  DH3:    .ASCIZ  /REG10   REG11   REG12   REG13/
4661  015216  020040  042522  030507
4662  015224  020061  020040  042522
4663  015232  030507  020062  020040
4664  015240  042522  030507  000063
4665  015246  020040  020040  042522  DH4:    .ASCIZ  /    REG14   REG15   REG16   REG17/
4666  015254  030507  020064  020040
4667  015262  042522  030507  020065
4668  015270  020040  042522  030507
4669  015276  020066  020040  042522
4670  015304  030507  000067
4671  015310  032461     000          DH5:    .ASCIZ  /15/
4672  015313     061  000066          DH6:    .ASCIZ  /16/
4673  015316  044514  042516  052440  DH7:    .ASCIZ  /LINE UNIT EXTENDED REGS :/
4674  015324  044516  020124  054105
4675  015332  042524  042116  042105
4676  015340  051040  043505  020123
4677  015346  000072
4678  015350  054101  026460  032461  DH8:    .ASCIZ  /AX0-15  AX0-16  AX1-15  AX1-16/
4679  015356  020040  054101  026460
4680  015364  033061  020040  054101
4681  015372  026461  032461  020040
4682  015400  054101  026461  033061
4683  015406     000
4684  015407     040  020040  040440  DH9:    .ASCIZ  /    AX2-15  AX2-16  AX3-15  AX3-16/
4685  015414  031130  030455  020065
4686  015422  040440  031130  030455
4587  015430  020066  040440  031530
4688  015436  030455  020065  040440
4689  015444  031530  030455  000066
4690
4691                                  .EVEN
4692
4693
4694
4695
4696
4697  015452                        BGNMSG  ERR1
4698  015452                                                ERR1::
4699  015452                        PRINTB  #FMT1,#ADDRES,MPCSR
4700  015452  013746  002424                                        MOV     MPCSR,-(SP)
4701  015456  012746  037002                                        MOV     #ADDRES,-(SP)
4702  015462  012746  012530                                        MOV     #FMT1,-(SP)
4703  015466  012746  000003                                        MOV     #3,-(SP)
4704  015472  010600                                                MOV     SP,R0
4705  015474  104414                                                TRAP    C$PNTB
4706  015476  062706  000010                                        ADD     #10,SP
4707  015502                        ENDMSG
4708  015502                                                L10002:
4709  015502  104423                                                TRAP    C$MSG
4710
4711
4712
4713  015504                        BGNMSG  ERR2
4714  015504                                                ERR2::
```

```
 4715  015504
 4716  015504  013746  002424          PRINTB  #FMT1,#ADDRES,MPCSR
 4717  015510  012746  037002                  MOV     MPCSR,-(SP)
 4718  015514  012746  012530                  MOV     #ADDRES,-(SP)
 4719  015520  012746  000003                  MOV     #FMT1,-(SP)
 4720  015524  010600                          MOV     #3,-(SP)
 4721  015526  104414                          MOV     SP,R0
 4722  015530  062706  000010                  TRAP    C$PNTB
 4723  015534                                  ADD     #10,SP
 4724  015534  012746  012540          PRINTB  #FMT2
 4725  015540  012746  000001                  MOV     #FMT2,-(SP)
 4726  015544  010600                          MOV     #1,-(SP)
 4727  015546  104414                          MOV     SP,R0
 4728  015550  062706  000004                  TRAP    C$PNTB
 4729  015554                                  ADD     #4,SP
 4730  015554  013746  002352          PRINTB  #FMT7,#DH1,REGNUM
 4731  015560  012746  015137                  MOV     REGNUM,-(SP)
 4732  015564  012746  012722                  MOV     #DH1,-(SP)
 4733  015570  012746  000003                  MOV     #FMT7,-(SP)
 4734  015574  010600                          MOV     #3,-(SP)
 4735  015576  104414                          MOV     SP,R0
 4736  015600  062706  000010                  TRAP    C$PNTB
 4737  015604                                  ADD     #10,SP
 4738  015604  013746  002360          PRINTB  #FMT3,GOODAT,BADDAT
 4739  015610  013746  002356                  MOV     BADDAT,-(SP)
 4740  015614  012746  012562                  MOV     GOODAT,-(SP)
 4741  015620  012746  000003                  MOV     #FMT3,-(SP)
 4742  015624  010600                          MOV     #3,-(SP)
 4743  015626  104414                          MOV     SP,R0
 4744  015630  062706  000010                  TRAP    C$PNTB
 4745  015634                                  ADD     #10,SP
 4746  015634  012746  015210          PRINTX  #FMT4,#DH2,#DH3
 4747  015640  012746  015161                  MOV     #DH3,-(SP)
 4748  015644  012746  012624                  MOV     #DH2,-(SP)
 4749  015650  012746  000003                  MOV     #FMT4,-(SP)
 4750  015654  010600                          MOV     #3,-(SP)
 4751  015656  104415                          MOV     SP,R0
 4752  015660  062706  000010                  TRAP    C$PNTX
 4753  015664                                  ADD     #10,SP
 4754  015664  013746  002262          PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
 4755  015670  013746  002260                  MOV     LUR13,-(SP)
 4756  015674  013746  002256                  MOV     LUR12,-(SP)
 4757  015700  013746  002254                  MOV     LUR11,-(SP)
 4758  015704  012746  012637                  MOV     LUR10,-(SP)
 4759  015710  012746  000005                  MOV     #FMT5,-(SP)
 4760  015714  010600                          MOV     #5,-(SP)
 4761  015716  104415                          MOV     SP,R0
 4762  015720  062706  000014                  TRAP    C$PNTX
 4763  015724                                  ADD     #14,SP
 4764  015724  012746  015246          PRINTX  #FMT9,#DH4
 4765  015730  012746  012766                  MOV     #DH4,-(SP)
 4766  015734  012746  000002                  MOV     #FMT9,-(SP)
 4767  015740  010600                          MOV     #2,-(SP)
 4768  015742  104415                          MOV     SP,R0
 4769  015744  062706  000006                  TRAP    C$PNTX
 4770  015750                                  ADD     #6,SP
                                 PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
```

```
 4771  015750  013746  002272                                                    MOV     LUR17,-(SP)
 4772  015754  013746  002270                                                    MOV     LUR16,-(SP)
 4773  015760  013746  002266                                                    MOV     LUR15,-(SP)
 4774  015764  013746  002264                                                    MOV     LUR14,-(SP)
 4775  015770  012746  012667                                                    MOV     #FMT6,-(SP)
 4776  015774  012746  000005                                                    MOV     #5,-(SP)
 4777  016000  010600                                                           MOV     SP,R0
 4778  016002  104415                                                           TRAP    C$PNTX
 4779  016004  062706  000014                                                    ADD     #14,SP
 4780  016010                      ENDMSG
 4781  016010                                                          L10003:
 4782  016010  104423                                                           TRAP    C$MSG
 4783
 4784
 4785
 4786
 4787
 4788  016012                      BGNMSG  ERR3
 4789  016012                                                          ERR3::
 4790  016012                      PRINTB  #FMT1,#ADDRES,MPCSR
 4791  016012  013746  002424                                                    MOV     MPCSR,-(SP)
 4792  016016  012746  037002                                                    MOV     #ADDRES,-(SP)
 4793  016022  012746  012530                                                    MOV     #FMT1,-(SP)
 4794  016026  012746  000003                                                    MOV     #3,-(SP)
 4795  016032  010600                                                           MOV     SP,R0
 4796  016034  104414                                                           TRAP    C$PNTB
 4797  016036  062706  000010                                                    ADD     #10,SP
 4798  016042                      PRINTB  #FMT2
 4799  016042  012746  012540                                                    MOV     #FMT2,-(SP)
 4800  016046  012746  000001                                                    MOV     #1,-(SP)
 4801  016052  010600                                                           MOV     SP,R0
 4802  016054  104414                                                           TRAP    C$PNTB
 4803  016056  062706  000004                                                    ADD     #4,SP
 4804  016062                      PRINTB  #FMT8,TMP1,TMP0
 4805  016062  013746  002502                                                    MOV     TMP0,-(SP)
 4806  016066  013746  002504                                                    MOV     TMP1,-(SP)
 4807  016072  012746  012732                                                    MOV     #FMT8,-(SP)
 4808  016076  012746  000003                                                    MOV     #3,-(SP)
 4809  016102  010600                                                           MOV     SP,R0
 4810  016104  104414                                                           TRAP    C$PNTB
 4811  016106  062706  000010                                                    ADD     #10,SP
 4812  016112                      PRINTB  #FMT3,GOODAT,BADDAT
 4813  016112  013746  002360                                                    MOV     BADDAT,-(SP)
 4814  016116  013746  002356                                                    MOV     GOODAT,-(SP)
 4815  016122  012746  012562                                                    MOV     #FMT3,-(SP)
 4816  016126  012746  000003                                                    MOV     #3,-(SP)
 4817  016132  010600                                                           MOV     SP,R0
 4818  016134  104414                                                           TRAP    C$PNTB
 4819  016136  062706  000010                                                    ADD     #10,SP
 4820  016142                      PRINTX  #FMT4,#DH2,#DH3
 4821  016142  012746  015210                                                    MOV     #DH3,-(SP)
 4822  016146  012746  015161                                                    MOV     #DH2,-(SP)
 4823  016152  012746  012624                                                    MOV     #FMT4,-(SP)
 4824  016156  012746  000003                                                    MOV     #3,-(SP)
 4825  016162  010600                                                           MOV     SP,R0
 4826  016164  104415                                                           TRAP    C$PNTX
```

```
 4827  016166  062706  000010                PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
 4828  016172                                        ADD     #10,SP
 4829  016172  013746  002262                      MOV     LUR13,-(SP)
 4830  016176  013746  002260                      MOV     LUR12,-(SP)
 4831  016202  013746  002256                      MOV     LUR11,-(SP)
 4832  016206  013746  002254                      MOV     LUR10,-(SP)
 4833  016212  012746  012637                      MOV     #FMT5,-(SP)
 4834  016216  012746  000005                      MOV     #5,-(SP)
 4835  016222  010600                            MOV     SP,R0
 4836  016224  104415                            TRAP    C$PNTX
 4837  016226  062706  000014                    ADD     #14,SP
 4838  016232                                PRINTX  #FMT9,#DH4
 4839  016232  012746  015246                      MOV     #DH4,-(SP)
 4840  016236  012746  012766                      MOV     #FMT9,-(SP)
 4841  016242  012746  000002                      MOV     #2,-(SP)
 4842  016246  010600                            MOV     SP,R0
 4843  016250  104415                            TRAP    C$PNTX
 4844  016252  062706  000006                    ADD     #6,SP
 4845  016256                                PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
 4846  016256  013746  002272                      MOV     LUR17,-(SP)
 4847  016262  013746  002270                      MOV     LUR16,-(SP)
 4848  016266  013746  002266                      MOV     LUR15,-(SP)
 4849  016272  013746  002264                      MOV     LUR14,-(SP)
 4850  016276  012746  012667                      MOV     #FMT6,-(SP)
 4851  016302  012746  000005                      MOV     #5,-(SP)
 4852  016306  010600                            MOV     SP,R0
 4853  016310  104415                            TRAP    C$PNTX
 4854  016312  062706  000014                    ADD     #14,SP
 4855  016316                                PRINTX  #FMT4,#DH7,#DH8
 4856  016316  012746  015350                      MOV     #DH8,-(SP)
 4857  016322  012746  015316                      MOV     #DH7,-(SP)
 4858  016326  012746  012624                      MOV     #FMT4,-(SP)
 4859  016332  012746  000003                      MOV     #3,-(SP)
 4860  016336  010600                            MOV     SP,R0
 4861  016340  104415                            TRAP    C$PNTX
 4862  016342  062706  000010                    ADD     #10,SP
 4863  016346                                PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
 4864  016346  013746  002302                      MOV     AX1.16,-(SP)
 4865  016352  013746  002300                      MOV     AX1.15,-(SP)
 4866  016356  013746  002276                      MOV     AX0.16,-(SP)
 4867  016362  013746  002274                      MOV     AX0.15,-(SP)
 4868  016366  012746  012637                      MOV     #FMT5,-(SP)
 4869  016372  012746  000005                      MOV     #5,-(SP)
 4870  016376  010600                            MOV     SP,R0
 4871  016400  104415                            TRAP    C$PNTX
 4872  016402  062706  000014                    ADD     #14,SP
 4873  016406                                PRINTX  #FMT9,#DH9
 4874  016406  012746  015407                      MOV     #DH9,-(SP)
 4875  016412  012746  012766                      MOV     #FMT9,-(SP)
 4876  016416  012746  000002                      MOV     #2,-(SP)
 4877  016422  010600                            MOV     SP,R0
 4878  016424  104415                            TRAP    C$PNTX
 4879  016426  062706  000006                    ADD     #6,SP
 4880  016432                                PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
 4881  016432  013746  002312                      MOV     AX3.16,-(SP)
 4882  016436  013746  002310                      MOV     AX3.15,-(SP)
```

```
 4883  016442  013746  002306                                                    MOV     AX2.16,-(SP)
 4884  016446  013746  002304                                                    MOV     AX2.15,-(SP)
 4885  016452  012746  012667                                                    MOV     #FMT6,-(SP)
 4886  016456  012746  000005                                                    MOV     #5,-(SP)
 4887  016462  010600                                                           MOV     SP,R0
 4888  016464  104415                                                           TRAP    C$PNTX
 4889  016466  062706  000014                                                   ADD     #14,SP
 4890  016472                        ENDMSG
 4891  016472                                                          L10004:
 4892  016472  104423                                                           TRAP    C$MSG
 4893
 4894
 4895
 4896
 4897
 4898  016474                        BGNMSG  ERR4
 4899  016474                                                          ERR4::
 4900  016474                        PRINTB  #FMT10,SUBRPC
 4901  016474  013746  002324                                                    MOV     SUBRPC,-(SP)
 4902  016500  012746  012773                                                    MOV     #FMT10,-(SP)
 4903  016504  012746  000002                                                    MOV     #2,-(SP)
 4904  016510  010600                                                           MOV     SP,R0
 4905  016512  104414                                                           TRAP    C$PNTB
 4906  016514  062706  000006                                                   ADD     #6,SP
 4907  016520                        PRINTB  #FMT1,#ADDRES,MPCSR
 4908  016520  013746  002424                                                    MOV     MPCSR,-(SP)
 4909  016524  012746  037002                                                    MOV     #ADDRES,-(SP)
 4910  016530  012746  012530                                                    MOV     #FMT1,-(SP)
 4911  016534  012746  000003                                                    MOV     #3,-(SP)
 4912  016540  010600                                                           MOV     SP,R0
 4913  016542  104414                                                           TRAP    C$PNTB
 4914  016544  062706  000010                                                   ADD     #10,SP
 4915  016550                        PRINTB  #FMT2
 4916  016550  012746  012540                                                    MOV     #FMT2,-(SP)
 4917  016554  012746  000001                                                    MOV     #1,-(SP)
 4918  016560  010600                                                           MOV     SP,R0
 4919  016562  104414                                                           TRAP    C$PNTB
 4920  016564  062706  000004                                                   ADD     #4,SP
 4921  016570                        PRINTB  #FMT7,#DH1,REGNUM
 4922  016570  013746  002352                                                    MOV     REGNUM,-(SP)
 4923  016574  012746  015137                                                    MOV     #DH1,-(SP)
 4924  016600  012746  012722                                                    MOV     #FMT7,-(SP)
 4925  016604  012746  000003                                                    MOV     #3,-(SP)
 4926  016610  010600                                                           MOV     SP,R0
 4927  016612  104414                                                           TRAP    C$PNTB
 4928  016614  062706  000010                                                   ADD     #10,SP
 4929  016620                        PRINTX  #FMT4,#DH2,#DH3
 4930  016620  012746  015210                                                    MOV     #DH3,-(SP)
 4931  016624  012746  015161                                                    MOV     #DH2,-(SP)
 4932  016630  012746  012624                                                    MOV     #FMT4,-(SP)
 4933  016634  012746  000003                                                    MOV     #3,-(SP)
 4934  016640  010600                                                           MOV     SP,R0
 4935  016642  104415                                                           TRAP    C$PNTX
 4936  016644  062706  000010                                                   ADD     #10,SP
 4937  016650                        PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
 4938  016650  0' 746  002262                                                    MOV     LUR13,-(SP)
```

```
4939  016654  013746  002260                                    MOV     LUR12,-(SP)
4940  016660  013746  002256                                    MOV     LUR11,-(SP)
4941  016664  013746  002254                                    MOV     LUR10,-(SP)
4942  016670  012746  012637                                    MOV     #FMT5,-(SP)
4943  016674  012746  000005                                    MOV     #5,-(SP)
4944  016700  010600                                            MOV     SP,R0
4945  016702  104415                                            TRAP    C$PNTX
4946  016704  062706  000014                                    ADD     #14,SP
4947  016710                  PRINTX  #FMT9,#DH4
4948  016710  012746  015246                                    MOV     #DH4,-(SP)
4949  016714  012746  012766                                    MOV     #FMT9,-(SP)
4950  016720  012746  000002                                    MOV     #2,-(SP)
4951  016724  010600                                            MOV     SP,R0
4952  016726  104415                                            TRAP    C$PNTX
4953  016730  062706  000006                                    ADD     #6,SP
4954  016734                  PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
4955  016734  013746  002272                                    MOV     LUR17,-(SP)
4956  016740  013746  002270                                    MOV     LUR16,-(SP)
4957  016744  013746  002266                                    MOV     LUR15,-(SP)
4958  016750  013746  002264                                    MOV     LUR14,-(SP)
4959  016754  012746  012667                                    MOV     #FMT6,-(SP)
4960  016760  012746  000005                                    MOV     #5,-(SP)
4961  016764  010600                                            MOV     SP,R0
4962  016766  104415                                            TRAP    C$PNTX
4963  016770  062706  000014                                    ADD     #14,SP
4964  016774                  PRINTX  #FMT4,#DH7,#DH8
4965  016774  012746  015350                                    MOV     #DH8,-(SP)
4966  017000  012746  015316                                    MOV     #DH7,-(SP)
4967  017004  012746  012624                                    MOV     #FMT4,-(SP)
4968  017010  012746  000003                                    MOV     #3,-(SP)
4969  017014  010600                                            MOV     SP,R0
4970  017016  104415                                            TRAP    C$PNTX
4971  017020  062706  000010                                    ADD     #10,SP
4972  017024                  PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
4973  017024  013746  002302                                    MOV     AX1.16,-(SP)
4974  017030  013746  002300                                    MOV     AX1.15,-(SP)
4975  017034  013746  002276                                    MOV     AX0.16,-(SP)
4976  017040  013746  002274                                    MOV     AX0.15,-(SP)
4977  017044  012746  012637                                    MOV     #FMT5,-(SP)
4978  017050  012746  000005                                    MOV     #5,-(SP)
4979  017054  010600                                            MOV     SP,R0
4980  017056  104415                                            TRAP    C$PNTX
4981  017060  062706  000014                                    ADD     #14,SP
4982  017064                  PRINTX  #FMT9,#DH9
4983  017064  012746  015407                                    MOV     #DH9,-(SP)
4984  017070  012746  012766                                    MOV     #FMT9,-(SP)
4985  017074  012746  000002                                    MOV     #2,-(SP)
4986  017100  010600                                            MOV     SP,R0
4987  017102  104415                                            TRAP    C$PNTX
4988  017104  062706  000006                                    ADD     #6,SP
4989  017110                  PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
4990  017110  013746  002312                                    MOV     AX3.16,-(SP)
4991  017114  013746  002310                                    MOV     AX3.15,-(SP)
4992  017120  013746  002306                                    MOV     AX2.16,-(SP)
4993  017124  013746  002304                                    MOV     AX2.15,-(SP)
4994  017130  012746  012667                                    MOV     #FMT6,-(SP)
```

```
 4995  017134  012746  000005                                                    MOV     #5,-(SP)
 4996  017140  010600                                                            MOV     SP,R0
 4997  017142  104415                                                            TRAP    C$PNTX
 4998  017144  062706  000014                                                    ADD     #14,SP
 4999  017150                        ENDMSG
 5000  017150                                                          L10005:
 5001  017150  104423                                                            TRAP    C$MSG
 5002
 5003
 5004
 5005
 5006
 5007  017152                        BGNMSG  ERR5
 5008  017152                                                          ERR5::
 5009  017152                        PRINTB  #FMT1,#ADDRES,MPCSR
 5010  017152  013746  002424                                                    MOV     MPCSR,-(SP)
 5011  017156  012746  037002                                                    MOV     #ADDRES,-(SP)
 5012  017162  012746  012530                                                    MOV     #FMT1,-(SP)
 5013  017166  012746  000003                                                    MOV     #3,-(SP)
 5014  017172  010600                                                            MOV     SP,R0
 5015  017174  104414                                                            TRAP    C$PNTB
 5016  017176  062706  000010                                                    ADD     #10,SP
 5017  017202                        PRINTB  #FMT11,REGNUM,LOADAT
 5018  017202  013746  002362                                                    MOV     LOADAT,-(SP)
 5019  017206  013746  002352                                                    MOV     REGNUM,-(SP)
 5020  017212  012746  013024                                                    MOV     #FMT11,-(SP)
 5021  017216  012746  000003                                                    MOV     #3,-(SP)
 5022  017222  010600                                                            MOV     SP,R0
 5023  017224  104414                                                            TRAP    C$PNTB
 5024  017226  062706  000010                                                    ADD     #10,SP
 5025  017232
 5026  017232  012746  012540                                                    MOV     #FMT2,-(SP)
 5027  017236  012746  000001                                                    MOV     #1,-(SP)
 5028  017242  010600                                                            MOV     SP,R0
 5029  017244  104414                                                            TRAP    C$PNTB
 5030  017246  062706  000004                                                    ADD     #4,SP
 5031  017252
 5032  017252  013746  002502                                                    MOV     TMP0,-(SP)
 5033  017256  013746  002504                                                    MOV     TMP1,-(SP)
 5034  017262  012746  012732                                                    MOV     #FMT8,-(SP)
 5035  017266  012746  000003                                                    MOV     #3,-(SP)
 5036  017272  010600                                                            MOV     SP,R0
 5037  017274  104414                                                            TRAP    C$PNTB
 5038  017276  062706  000010                                                    ADD     #10,SP
 5039  017302
 5040  017302  013746  002360                                                    MOV     BADDAT,-(SP)
 5041  017306  013746  002356                                                    MOV     GOODAT,-(SP)
 5042  017312  012746  012562                                                    MOV     #FMT3,-(SP)
 5043  017316  012746  000003                                                    MOV     #3,-(SP)
 5044  017322  010600                                                            MOV     SP,R0
 5045  017324  104414                                                            TRAP    C$PNTB
 5046  017326  062706  000010                                                    ADD     #10,SP
 5047  017332
 5048  017332  012746  015210                                                    MOV     #DH3,-(SP)
 5049  017336  012746  015161                                                    MOV     #DH2,-(SP)
 5050  017342  012746  012624                                                    MOV     #FMT4,-(SP)
```

```
5051  017346  012746  000003                          MOV     #3,-(SP)
5052  017352  010600                                  MOV     SP,R0
5053  017354  104415                                  TRAP    C$PNTX
5054  017356  062706  000010                          ADD     #10,SP
5055  017362                  PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
5056  017362  013746  002262                          MOV     LUR13,-(SP)
5057  017366  013746  002260                          MOV     LUR12,-(SP)
5058  017372  013746  002256                          MOV     LUR11,-(SP)
5059  017376  013746  002254                          MOV     LUR10,-(SP)
5060  017402  012746  012637                          MOV     #FMT5,-(SP)
5061  017406  012746  000005                          MOV     #5,-(SP)
5062  017412  010600                                  MOV     SP,R0
5063  017414  104415                                  TRAP    C$PNTX
5064  017416  062706  000014                          ADD     #14,SP
5065  017422                  PRINTX  #FMT9,#DH4
5066  017422  012746  015246                          MOV     #DH4,-(SP)
5067  017426  012746  012766                          MOV     #FMT9,-(SP)
5068  017432  012746  000002                          MOV     #2,-(SP)
5069  017436  010600                                  MOV     SP,R0
5070  017440  104415                                  TRAP    C$PNTX
5071  017442  062706  000006                          ADD     #6,SP
5072  017446                  PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
5073  017446  013746  002272                          MOV     LUR17,-(SP)
5074  017452  013746  002270                          MOV     LUR16,-(SP)
5075  017456  013746  002266                          MOV     LUR15,-(SP)
5076  017462  013746  002264                          MOV     LUR14,-(SP)
5077  017466  012746  012667                          MOV     #FMT6,-(SP)
5078  017472  012746  000005                          MOV     #5,-(SP)
5079  017476  010600                                  MOV     SP,R0
5080  017500  104415                                  TRAP    C$PNTX
5081  017502  062706  000014                          ADD     #14,SP
5082  017506                  PRINTX  #FMT4,#DH7,#DH8
5083  017506  012746  015350                          MOV     #DH8,-(SP)
5084  017512  012746  015316                          MOV     #DH7,-(SP)
5085  017516  012746  012624                          MOV     #FMT4,-(SP)
5086  017522  012746  000003                          MOV     #3,-(SP)
5087  017526  010600                                  MOV     SP,R0
5088  017530  104415                                  TRAP    C$PNTX
5089  017532  062706  000010                          ADD     #10,SP
5090  017536                  PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
5091  017536  013746  002302                          MOV     AX1.16,-(SP)
5092  017542  013746  002300                          MOV     AX1.15,-(SP)
5093  017546  013746  002276                          MOV     AX0.16,-(SP)
5094  017552  013746  002274                          MOV     AX0.15,-(SP)
5095  017556  012746  012637                          MOV     #FMT5,-(SP)
5096  017562  012746  000005                          MOV     #5,-(SP)
5097  017566  010600                                  MOV     SP,R0
5098  017570  104415                                  TRAP    C$PNTX
5099  017572  062706  000014                          ADD     #14,SP
5100  017576                  PRINTX  #FMT9,#DH9
5101  017576  012746  015407                          MOV     #DH9,-(SP)
5102  017602  012746  012766                          MOV     #FMT9,-(SP)
5103  017606  012746  000002                          MOV     #2,-(SP)
5104  017612  010600                                  MOV     SP,R0
5105  017614  104415                                  TRAP    C$PNTX
5106  017616  062706  000006                          ADD     #6,SP
```

```
5107  017622                                  PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
5108  017622  013746  002312                                      MOV     AX3.16,-(SP)
5109  017626  013746  002310                                      MOV     AX3.15,-(SP)
5110  017632  013746  002306                                      MOV     AX2.16,-(SP)
5111  017636  013746  002304                                      MOV     AX2.15,-(SP)
5112  017642  012746  012667                                      MOV     #FMT6,-(SP)
5113  017646  012746  000005                                      MOV     #5,-(SP)
5114  017652  010600                                              MOV     SP,R0
5115  017654  104415                                              TRAP    C$PNTX
5116  017656  062706  000014                                      ADD     #14,SP
5117  017662                                  ENDMSG
5118  017662                                                  L10006:
5119  017662  104423                                              TRAP    C$MSG
5120
5121
5122
5123
5124
5125  017664                                  BGNMSG  ERR6
5126  017664                                                  ERR6::
5127  017[illegible]                          PRINTB  #FMT10,SUBRPC
5128  017664  013746  002324                                      MOV     SUBRPC,-(SP)
5129  017670  012746  012773                                      MOV     #FMT10,-(SP)
5130  017674  012746  000002                                      MOV     #2,-(SP)
5131  017700  010600                                              MOV     SP,R0
5132  017702  104414                                              TRAP    C$PNTB
5133  017704  062706  000006                                      ADD     #6,SP
5134  017710                                  PRINTB  #FMT1,#ADDRES,MPCSR
5135  017710  013746  002424                                      MOV     MPCSR,-(SP)
5136  017714  012746  037002                                      MOV     #ADDRES,-(SP)
5137  017720  012746  012530                                      MOV     #FMT1,-(SP)
5138  017724  012746  000003                                      MOV     #3,-(SP)
5139  017730  010600                                              MOV     SP,R0
5140  017732  104414                                              TRAP    C$PNTB
5141  017734  062706  000010                                      ADD     #10,SP
5142  017740                                  PRINTB  #FMT2
5143  017740  012746  012540                                      MOV     #FMT2,-(SP)
5144  017744  012746  000001                                      MOV     #1,-(SP)
5145  017750  010600                                              MOV     SP,R0
5146  017752  104414                                              TRAP    C$PNTB
5147  017754  062706  000004                                      ADD     #4,SP
5148  017760                                  PRINTB  #FMT8,TMP1,TMP0
5149  017760  013746  002502                                      MOV     TMP0,-(SP)
5150  017764  013746  002504                                      MOV     TMP1,-(SP)
5151  017770  012746  012732                                      MOV     #FMT8,-(SP)
5152  017774  012746  000003                                      MOV     #3,-(SP)
5153  020000  010600                                              MOV     SP,R0
5154  020002  104414                                              TRAP    C$PNTB
5155  020004  062706  000010                                      ADD     #10,SP
5156  020010                                  PRINTX  #FMT4,#DH2,#DH3
5157  020010  012746  015210                                      MOV     #DH3,-(SP)
5158  020014  012746  015161                                      MOV     #DH2,-(SP)
5159  020020  012746  012624                                      MOV     #FMT4,-(SP)
5160  020024  012746  000003                                      MOV     #3,-(SP)
5161  020030  010600                                              MOV     SP,R0
5162  020032  104415                                              TRAP    C$PNTX
```

```
5163  020034  062706  000010                  ADD     #10,SP
5164  020040
5165  020040  013746  002262          PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
5166  020044  013746  002260                  MOV     LUR13,-(SP)
5167  020050  013746  002256                  MOV     LUR12,-(SP)
5168  020054  013746  002254                  MOV     LUR11,-(SP)
5169  020060  012746  012637                  MOV     LUR10,-(SP)
5170  020064  012746  000005                  MOV     #FMT5,-(SP)
5171  020070  010600                          MOV     #5,-(SP)
5172  020072  104415                          MOV     SP,R0
5173  020074  062706  000014                  TRAP    C$PNTX
5174  020100                                  ADD     #14,SP
5175  020100  012746  015246          PRINTX  #FMT9,#DH4
5176  020104  012746  012766                  MOV     #DH4,-(SP)
5177  020110  012746  000002                  MOV     #FMT9,-(SP)
5178  020114  010600                          MOV     #2,-(SP)
5179  020116  104415                          MOV     SP,R0
5180  020120  062706  000006                  TRAP    C$PNTX
5181  020124                                  ADD     #6,SP
5182  020124  013746  002272          PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
5183  020130  013746  002270                  MOV     LUR17,-(SP)
5184  020134  013746  002266                  MOV     LUR16,-(SP)
5185  020140  013746  002264                  MOV     LUR15,-(SP)
5186  020144  012746  012667                  MOV     LUR14,-(SP)
5187  020150  012746  000005                  MOV     #FMT6,-(SP)
5188  020154  010600                          MOV     #5,-(SP)
5189  020156  104415                          MOV     SP,R0
5190  020160  062706  000014                  TRAP    C$PNTX
5191  020164                                  ADD     #14,SP
5192  020164  012746  015350          PRINTX  #FMT4,#DH7,#DH8
5193  020170  012746  015316                  MOV     #DH8,-(SP)
5194  020174  012746  012624                  MOV     #DH7,-(SP)
5195  020200  012746  000003                  MOV     #FMT4,-(SP)
5196  020204  010600                          MOV     #3,-(SP)
5197  020206  104415                          MOV     SP,R0
5198  020210  062706  000010                  TRAP    C$PNTX
5199  020214                                  ADD     #10,SP
5200  020214  013746  002302          PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
5201  020220  013746  002300                  MOV     AX1 16,-(SP)
5202  020224  013746  002276                  MOV     AX1.15,-(SP)
5203  020230  013746  002274                  MOV     AX0.16,-(SP)
5204  020234  012746  012637                  MOV     AX0.15,-(SP)
5205  020240  012746  000005                  MOV     #FMT5,-(SP)
5206  020244  010600                          MOV     #5,-(SP)
5207  020246  104415                          MOV     SP,R0
5208  020250  062706  000014                  TRAP    C$PNTX
5209  020254                                  ADD     #14,SP
5210  020254  012746  015407          PRINTX  #FMT9,#DH9
5211  020260  012746  012766                  MOV     #DH9,-(SP)
5212  020264  012746  000002                  MOV     #FMT9,-(SP)
5213  020270  010600                          MOV     #2,-(SP)
5214  020272  104415                          MOV     SP,R0
5215  020274  062706  000006                  TRAP    C$PNTX
5216  020300                                  ADD     #6,SP
5217  020300  013746  002312          PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
5218  020304  013746  002310                  MOV     AX3.16,-(SP)
                                              MOV     AX3.15,-(SP)
```

```
5219  020310  013746  002306                                                    MOV     AX2.16,-(SP)
5220  020314  013746  002304                                                    MOV     AX2.15,-(SP)
5221  020320  012746  012667                                                    MOV     #FMT6,-(SP)
5222  020324  012746  000005                                                    MOV     #5,-(SP)
5223  020330  010600                                                            MOV     SP,R0
5224  020332  104415                                                            TRAP    C$PNTX
5225  020334  062706  000014                                                    ADD     #14,SP
5226  020340                      ENDMSG
5227  020340                                                          L10007:
5228  020340  104423                                                            TRAP    C$MSG
5229
5230
5231
5232
5233
5234  020342                      BGNMSG  ERR7
5235  020342                                                          ERR7::
5236  020342                      PRINTB  #FMT1,#ADDRES,MPCSR
5237  020342  013746  002424                                                    MOV     MPCSR,-(SP)
5238  020346  012746  037002                                                    MOV     #ADDRES,-(SP)
5239  020352  012746  012530                                                    MOV     #FMT1,-(SP)
5240  020356  012746  000003                                                    MOV     #3,-(SP)
5241  020362  010600                                                            MOV     SP,R0
5242  020364  104414                                                            TRAP    C$PNTB
5243  020366  062706  000010                                                    ADD     #10,SP
5244  020372                      PRINTB  #FMT2
5245  020372  012746  012540                                                    MOV     #FMT2,-(SP)
5246  020376  012746  000001                                                    MOV     #1,-(SP)
5247  020402  010600                                                            MOV     SP,R0
5248  020404  104414                                                            TRAP    C$PNTB
5249  020406  062706  000004                                                    ADD     #4,SP
5250  020412                      PRINTB  #FMT7,#DH1,REGNUM
5251  020412  013746  002352                                                    MOV     REGNUM,-(SP)
5252  020416  012746  015137                                                    MOV     #DH1,-(SP)
5253  020422  012746  012722                                                    MOV     #FMT7,-(SP)
5254  020426  012746  000003                                                    MOV     #3,-(SP)
5255  020432  010600                                                            MOV     SP,R0
5256  020434  104414                                                            TRAP    C$PNTB
5257  020436  062706  000010                                                    ADD     #10,SP
5258  020442                      PRINTX  #FMT4,#DH2,#DH3
5259  020442  012746  015210                                                    MOV     #DH3,-(SP)
5260  020446  012746  015161                                                    MOV     #DH2,-(SP)
5261  020452  012746  012624                                                    MOV     #FMT4,-(SP)
5262  020456  012746  000003                                                    MOV     #3,-(SP)
5263  020462  010600                                                            MOV     SP,R0
5264  020464  104415                                                            TRAP    C$PNTX
5265  020466  062706  000010                                                    ADD     #10,SP
5266  020472                      PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
5267  020472  013746  002262                                                    MOV     LUR13,-(SP)
5268  020476  013746  002260                                                    MOV     LUR12,-(SP)
5269  020502  013746  002256                                                    MOV     LUR11,-(SP)
5270  020506  013746  002254                                                    MOV     LUR10,-(SP)
5271  020512  012746  012637                                                    MOV     #FMT5,-(SP)
5272  020516  012746  000005                                                    MOV     #5,-(SP)
5273  020522  010600                                                            MOV     SP,R0
5274  020524  104415                                                            TRAP    C$PNTX
```

```
5275  020526  062706  000014                  PRINTX  #FMT9,#DH4
5276  020532                                          ADD     #14,SP
5277  020532  012746  015246                          MOV     #DH4,-(SP)
5278  020536  012746  012766                          MOV     #FMT9,-(SP)
5279  020542  012746  000002                          MOV     #2,-(SP)
5280  020546  010600                                  MOV     SP,R0
5281  020550  104415                                  TRAP    C$PNTX
5282  020552  062706  000006                  PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
5283  020556                                          ADD     #6,SP
5284  020556  013746  002272                          MOV     LUR17,-(SP)
5285  020562  013746  002270                          MOV     LUR16,-(SP)
5286  020566  013746  002266                          MOV     LUR15,-(SP)
5287  020572  013746  002264                          MOV     LUR14,-(SP)
5288  020576  012746  012667                          MOV     #FMT6,-(SP)
5289  020602  012746  000005                          MOV     #5,-(SP)
5290  020606  010600                                  MOV     SP,R0
5291  020610  104415                                  TRAP    C$PNTX
5292  020612  062706  000014                  PRINTX  #FMT4,#DH7,#DH8
5293  020616                                          ADD     #14,SP
5294  020616  012746  015350                          MOV     #DH8,-(SP)
5295  020622  012746  015316                          MOV     #DH7,-(SP)
5296  020626  012746  012624                          MOV     #FMT4,-(SP)
5297  020632  012746  000003                          MOV     #3,-(SP)
5298  020636  010600                                  MOV     SP,R0
5299  020640  104415                                  TRAP    C$PNTX
5300  020642  062706  000010                  PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
5301  020646                                          ADD     #10,SP
5302  020646  013746  002302                          MOV     AX1.16,-(SP)
5303  020652  013746  002300                          MOV     AX1.15,-(SP)
5304  020656  013746  002276                          MOV     AX0.16,-(SP)
5305  020662  013746  002274                          MOV     AX0.15,-(SP)
5306  020666  012746  012637                          MOV     #FMT5,-(SP)
5307  020672  012746  000005                          MOV     #5,-(SP)
5308  020676  010600                                  MOV     SP,R0
5309  020700  104415                                  TRAP    C$PNTX
5310  020702  062706  000014                  PRINTX  #FMT9,#DH9
5311  020706                                          ADD     #14,SP
5312  020706  012746  015407                          MOV     #DH9,-(SP)
5313  020712  012746  012766                          MOV     #FMT9,-(SP)
5314  020716  012746  000002                          MOV     #2,-(SP)
5315  020722  010600                                  MOV     SP,R0
5316  020724  104415                                  TRAP    C$PNTX
5317  020726  062706  000006                  PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
5318  020732                                          ADD     #6,SP
5319  020732  013746  002312                          MOV     AX3.16,-(SP)
5320  020736  013746  002310                          MOV     AX3.15,-(SP)
5321  020742  013746  002306                          MOV     AX2.16,-(SP)
5322  020746  013746  002304                          MOV     AX2.15,-(SP)
5323  020752  012746  012667                          MOV     #FMT6,-(SP)
5324  020756  012746  000005                          MOV     #5,-(SP)
5325  020762  010600                                  MOV     SP,R0
5326  020764  104415                                  TRAP    C$PNTX
5327  020766  062706  000014                          ADD     #14,SP
5328  020772                  ENDMSG
5329  020772                                  L10010:
5330  020772  104423                                  TRAP    C$MSG
```

```
5331
5332
5333
5334
5335
5336  020774                          BGNMSG  ERR8
5337  020774                                                  ERR8::
5338  020774                          PRINTB  #FMT10,SUBRPC
5339  020774  013746  002324                                          MOV     SUBRPC,-(SP)
5340  021000  012746  012773                                          MOV     #FMT10,-(SP)
5341  021004  012746  000002                                          MOV     #2,-(SP)
5342  021010  010600                                                  MOV     SP,R0
5343  021012  104414                                                  TRAP    C$PNTB
5344  021014  062706  000006                                          ADD     #6,SP
5345  021020                          PRINTB  #FMT1,#ADDRES,MPCSR
5346  021020  013746  002424                                          MOV     MPCSR,-(SP)
5347  021024  012746  037002                                          MOV     #ADDRES,-(SP)
5348  021030  012746  012530                                          MOV     #FMT1,-(SP)
5349  021034  012746  000003                                          MOV     #3,-(SP)
5350  021040  010600                                                  MOV     SP,R0
5351  021042  104414                                                  TRAP    C$PNTB
5352  021044  062706  000010                                          ADD     #10,SP
5353  021050                          PRINTB  #FMT2
5354  021050  012746  012540                                          MOV     #FMT2,-(SP)
5355  021054  012746  000001                                          MOV     #1,-(SP)
5356  021060  010600                                                  MOV     SP,R0
5357  021062  104414                                                  TRAP    C$PNTB
5358  021064  062706  000004                                          ADD     #4,SP
5359  021070                          PRINTB  #FMT7,#DH1,REGNUM
5360  021070  013746  002352                                          MOV     REGNUM,-(SP)
5361  021074  012746  015137                                          MOV     #DH1,-(SP)
5362  021100  012746  012722                                          MOV     #FMT7,-(SP)
5363  021104  012746  000003                                          MOV     #3,-(SP)
5364  021110  010600                                                  MOV     SP,R0
5365  021112  104414                                                  TRAP    C$PNTB
5366  021114  062706  000010                                          ADD     #10,SP
5367  021120                          PRINTB  #FMT3,GOODAT,BADDAT
5368  021120  013746  002360                                          MOV     BADDAT,-(SP)
5369  021124  013746  002356                                          MOV     GOODAT,-(SP)
5370  021130  012746  012562                                          MOV     #FMT3,-(SP)
5371  021134  012746  000003                                          MOV     #3,-(SP)
5372  021140  010600                                                  MOV     SP,R0
5373  021142  104414                                                  TRAP    C$PNTB
5374  021144  062706  000010                                          ADD     #10,SP
5375  021150                          PRINTX  #FMT4,#DH2,#DH3
5376  021150  012746  015210                                          MOV     #DH3,-(SP)
5377  021154  012746  015161                                          MOV     #DH2,-(SP)
5378  021160  012746  012624                                          MOV     #FMT4,-(SP)
5379  021164  012746  000003                                          MOV     #3,-(SP)
5380  021170  010600                                                  MOV     SP,R0
5381  021172  104415                                                  TRAP    C$PNTX
5382  021174  062706  000010                                          ADD     #10,SP
5383  021200                          PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
5384  021200  013746  002262                                          MOV     LUR13,-(SP)
5385  021204  013746  002260                                          MOV     LUR12,-(SP)
5386  021210  013746  002256                                          MOV     LUR11,-(SP)
```

```
5387  021214  013746  002254                                          MOV     LUR10,-(SP)
5388  021220  012746  012637                                          MOV     #FMT5,-(SP)
5389  021224  012746  000005                                          MOV     #5,-(SP)
5390  021230  010600                                                  MOV     SP,R0
5391  021232  104415                                                  TRAP    C$PNTX
5392  021234  062706  000014                                          ADD     #14,SP
5393  021240                  PRINTX  #FMT9,#DH4
5394  021240  012746  015246                                          MOV     #DH4,-(SP)
5395  021244  012746  012766                                          MOV     #FMT9,-(SP)
5396  021250  012746  000002                                          MOV     #2,-(SP)
5397  021254  010600                                                  MOV     SP,R0
5398  021256  104415                                                  TRAP    C$PNTX
5399  021260  062706  000006                                          ADD     #6,SP
5400  021264                  PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
5401  021264  013746  002272                                          MOV     LUR17,-(SP)
5402  021270  013746  002270                                          MOV     LUR16,-(SP)
5403  021274  013746  002266                                          MOV     LUR15,-(SP)
5404  021300  013746  002264                                          MOV     LUR14,-(SP)
5405  021304  012746  012667                                          MOV     #FMT6,-(SP)
5406  021310  012746  000005                                          MOV     #5,-(SP)
5407  021314  010600                                                  MOV     SP,R0
5408  021316  104415                                                  TRAP    C$PNTX
5409  021320  062706  000014                                          ADD     #14,SP
5410  021324                  PRINTX  #FMT4,#DH7,#DH8
5411  021324  012746  015350                                          MOV     #DH8,-(SP)
5412  021330  012746  015316                                          MOV     #DH7,-(SP)
5413  021334  012746  012624                                          MOV     #FMT4,-(SP)
5414  021340  012746  000003                                          MOV     #3,-(SP)
5415  021344  010600                                                  MOV     SP,R0
5416  021346  104415                                                  TRAP    C$PNTX
5417  021350  062706  000010                                          ADD     #10,SP
5418  021354                  PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
5419  021354  013746  002302                                          MOV     AX1.16,-(SP)
5420  021360  013746  002300                                          MOV     AX1.15,-(SP)
5421  021364  013746  002276                                          MOV     AX0.16,-(SP)
5422  021370  013746  002274                                          MOV     AX0.15,-(SP)
5423  021374  012746  012637                                          MOV     #FMT5,-(SP)
5424  021400  012746  000005                                          MOV     #5,-(SP)
5425  021404  010600                                                  MOV     SP,R0
5426  021406  104415                                                  TRAP    C$PNTX
5427  021410  062706  000014                                          ADD     #14,SP
5428  021414                  PRINTX  #FMT9,#DH9
5429  021414  012746  015407                                          MOV     #DH9,-(SP)
5430  021420  012746  012766                                          MOV     #FMT9,-(SP)
5431  021424  012746  000002                                          MOV     #2,-(SP)
5432  021430  010600                                                  MOV     SP,R0
5433  021432  104415                                                  TRAP    C$PNTX
5434  021434  062706  000006                                          ADD     #6,SP
5435  021440                  PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
5436  021440  013746  002312                                          MOV     AX3.16,-(SP)
5437  021444  013746  002310                                          MOV     AX3.15,-(SP)
5438  021450  013746  002306                                          MOV     AX2.16,-(SP)
5439  021454  013746  002304                                          MOV     AX2.15,-(SP)
5440  021460  012746  012667                                          MOV     #FMT6,-(SP)
5441  021464  012746  000005                                          MOV     #5,-(SP)
5442  021470  010600                                                  MOV     SP,R0
```

```
5443  021472  104415                                          TRAP    C$PNTX
5444  021474  062706  000014                                  ADD     #14,SP
5445  021500                      ENDMSG
5446  021500                                          L10011:
5447  021500  104423                                          TRAP    C$MSG
5448
5449
5450
5451
5452
5453  021502                      BGNMSG  ERR10
5454  021502                                          ERR10::
5455  021502                      PRINTB  #FMT1,#ADDRES,MPCSR
5456  021502  013746  002424                                  MOV     MPCSR,-(SP)
5457  021506  012746  037002                                  MOV     #ADDRES,-(SP)
5458  021512  012746  012530                                  MOV     #FMT1,-(SP)
5459  021516  012746  000003                                  MOV     #3,-(SP)
5460  021522  010600                                          MOV     SP,R0
5461  021524  104414                                          TRAP    C$PNTB
5462  021526  062706  000010                                  ADD     #10,SP
5463  021532                      PRINTB  #FMT2
5464  021532  012746  012540                                  MOV     #FMT2,-(SP)
5465  021536  012746  000001                                  MOV     #1,-(SP)
5466  021542  010600                                          MOV     SP,R0
5467  021544  104414                                          TRAP    C$PNTB
5468  021546  062706  000004                                  ADD     #4,SP
5469  021552                      PRINTB  #FMT8,TMP1,TMP0
5470  021552  013746  002502                                  MOV     TMP0,-(SP)
5471  021556  013746  002504                                  MOV     TMP1,-(SP)
5472  021562  012746  012732                                  MOV     #FMT8,-(SP)
5473  021566  012746  000003                                  MOV     #3,-(SP)
5474  021572  010600                                          MOV     SP,R0
5475  021574  104414                                          TRAP    C$PNTB
5476  021576  062706  000010                                  ADD     #10,SP
5477  021602                      PRINTX  #FMT4,#DH2,#DH3
5478  021602  012746  015210                                  MOV     #DH3,-(SP)
5479  021606  012746  015161                                  MOV     #DH2,-(SP)
5480  021612  012746  012624                                  MOV     #FMT4,-(SP)
5481  021616  012746  000003                                  MOV     #3,-(SP)
5482  021622  010600                                          MOV     SP,R0
5483  021624  104415                                          TRAP    C$PNTX
5484  021626  062706  000010                                  ADD     #10,SP
5485  021632                      PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
5486  021632  013746  002262                                  MOV     LUR13,-(SP)
5487  021636  013746  002260                                  MOV     LUR12,-(SP)
5488  021642  013746  002256                                  MOV     LUR11,-(SP)
5489  021646  013746  002254                                  MOV     LUR10,-(SP)
5490  021652  012746  012637                                  MOV     #FMT5,-(SP)
5491  021656  012746  000005                                  MOV     #5,-(SP)
5492  021662  010600                                          MOV     SP,R0
5493  021664  104415                                          TRAP    C$PNTX
5494  021666  062706  000014                                  ADD     #14,SP
5495  021672                      PRINTX  #FMT9,#DH4
5496  021672  012746  015246                                  MOV     #DH4,-(SP)
5497  021676  012746  012766                                  MOV     #FMT9,-(SP)
5498  021702  012746  000002                                  MOV     #2,-(SP)
```

```
5499  021706  010600
5500  021710  104415
5501  021712  062706  000006
5502  021716
5503  021716  013746  002272
5504  021722  013746  002270
5505  021726  013746  002266
5506  021732  013746  002264
5507  021736  012746  012667
5508  021742  012746  000005
5509  021746  010600
5510  021750  104415
5511  021752  062706  000014
5512  021756
5513  021756  012746  015350
5514  021762  012746  015316
5515  021766  012746  012624
5516  021772  012746  000003
5517  021776  010600
5518  022000  104415
5519  022002  062706  000010
5520  022006
5521  022006  013746  002302
5522  022012  013746  002300
5523  022016  013746  002276
5524  022022  013746  002274
5525  022026  012746  012637
5526  022032  012746  000005
5527  022036  010600
5528  022040  104415
5529  022042  062706  000014
5530  022046
5531  022046  012746  015407
5532  022052  012746  012766
5533  022056  012746  000002
5534  022062  010600
5535  022064  104415
5536  022066  062706  000006
5537  022072
5538  022072  013746  002312
5539  022076  013746  002310
5540  022102  013746  002306
5541  022106  013746  002304
5542  022112  012746  012667
5543  022116  012746  000005
5544  022122  010600
5545  022124  104415
5546  022126  062706  000014
5547  022132
5548  022132
5549  022132  104423
5550
5551
5552
5553
5554
```

```
                PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17

                PRINTX  #FMT4,#DH7,#DH8

                PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16

                PRINTX  #FMT9,#DH9

                PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16

                ENDMSG
```

```
                MOV     SP,R0
                TRAP    C$PNTX
                ADD     #6,SP
                MOV     LUR17,-(SP)
                MOV     LUR16,-(SP)
                MOV     LUR15,-(SP)
                MOV     LUR14,-(SP)
                MOV     #FMT6,-(SP)
                MOV     #5,-(SP)
                MOV     SP,R0
                TRAP    C$PNTX
                ADD     #14,SP
                MOV     #DH8,-(SP)
                MOV     #DH7,-(SP)
                MOV     #FMT4,-(SP)
                MOV     #3,-(SP)
                MOV     SP,R0
                TRAP    C$PNTX
                ADD     #10,SP
                MOV     AX1.16,-(SP)
                MOV     AX1.15,-(SP)
                MOV     AX0.16,-(SP)
                MOV     AX0.15,-(SP)
                MOV     #FMT5,-(SP)
                MOV     #5,-(SP)
                MOV     SP,R0
                TRAP    C$PNTX
                ADD     #14,SP
                MOV     #DH9,-(SP)
                MOV     #FMT9,-(SP)
                MOV     #2,-(SP)
                MOV     SP,R0
                TRAP    C$PNTX
                ADD     #6,SP
                MOV     AX3.16,-(SP)
                MOV     AX3.15,-(SP)
                MOV     AX2.16,-(SP)
                MOV     AX2.15,-(SP)
                MOV     #FMT6,-(SP)
                MOV     #5,-(SP)
                MOV     SP,R0
                TRAP    C$PNTX
                ADD     #14,SP
L10012:
                TRAP    C$MSG
```

CZDMSF.P11 30-SEP-81 15:40 REPORT CODING SECTION

```
5555                            .SBTTL  REPORT CODING SECTION
5556
5557
5558                            ;//////////////////////////////////////////////////////////////
5559                            ;/ THE REPORT CODING SECTION CONTAINS THE
5560                            ;/ 'PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
5561                            ;//////////////////////////////////////////////////////////////
5562
5563  022134                            BGNRPT
5564  022134                                                          L$RPT::
5565
5566  022134                            ENDRPT
5567  022134                                                          L10013:
5568  022134  104425                                                         TRAP    C$RPT
5569
5570                            .EVEN
5571
5572
5573
5574
5575
```

```
CZDMSF.P11    30-SEP-81 15:40          LOAD DEVICE PROTECTION TABLE

                                       .SBTTL  LOAD DEVICE PROTECTION TABLE

                                       ;////////////////////////////////////////////////////////////////////////////////
                                       ;/ THIS TABLE IDENTIFIES THE LOAD DEVICE TO THE SUPERVISOR, SO THAT IT CAN BE
                                       ;/ PROTECTED FROM TESTING, IF DESIRED.
                                       ;////////////////////////////////////////////////////////////////////////////////

  5583  022136                                 BGNPROT
  5584  022136                                                                L$PROT::
  5585  022136  177777                        .WORD   -1      ;DON'T CHK CSR ADRS
  5586  022140  177777                        .WORD   -1      ;DON'T CHK MASSBUS UNIT NO.
  5587  022142  177777                        .WORD   -1      ;DON'T CHK DRIVE NO.
  5588  022144                                 ENDPROT
```

```
5594                                    .SBTTL  INITIALIZE SECTION
5595
5596                                    ;////////////////////////////////////////////////////////////////////
5597                                    ;/ THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
5598                                    ;/ AT THE BEGINNING OF THE TEST SEQUENCE ON THE NEXT UNIT.
5599                                    ;////////////////////////////////////////////////////////////////////
5600
5601  022144                                    BGNINIT
5602  022144                                                                              L$INIT::
5603
5604  022144  010637  002320                    MOV     SP,PSTACK       ;SAVE BASE-LEVEL STACK POINTER
5605  022150  005037  002324                    CLR     SUBRPC          ;CLEAR SUBR CALL PC
5606  022154  005037  002404                    CLR     DISILO          ;CLEAR CURRENT STATE OF DISSI
5607  022160  005037  002406                    CLR     CHPTYP          ;CLEAR USYRT CHIP TYPE INDICATOR
5608  022164  005037  002376                    CLR     ERROR1          ;CLEAR ERROR FLAGS
5609  022170  005037  002412                    CLR     SAVLEN          ;CLEAR CHAR LENGTH FROM SETUP
5610  022174  005737  002364                    TST     FRSTIM          ;SEE IF FIRST TIME THROUGH AFTER LOAD
5611  022200  001007                            BNE     6$              ;BR IF NOT
5612  022202  013737  000004  002372            MOV     @#4,SAVE4       ;SAVE ERROR TRAP VECTOR
5613  022210  013737  000006  002374            MOV     @#6,SAVE6
5614  022216  000406                            BR      9$
5615  022220  013737  002372  000004  6$:       MOV     SAVE4,@#4       ;RESTORE ERROR TRAP VECTOR
5616  022226  013737  002374  000006            MOV     SAVE6,@#6
5617  022234  012737  000001  002364  9$:       MOV     #1,FRSTIM       ;MARK FLAG FOR NEXT TIME THROUGH
5618                                    ;SEE IF PROGRAM JUST STARTED, BR IF YES
5619  022242                                    READEF  #EF.START
5620  022242  012700  000040                                                            MOV     #EF.START,R0
5621  022246  104447                                                                    TRAP    C$REFG
5622  022250                                    BCOMPLETE       STARST
5623  022250  103417                                                                    BCS     STARST
5624                                    ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
5625  022252                                    READEF  #EF.RESTART
5626  022252  012700  000037                                                            MOV     #EF.RESTART,R0
5627  022256  104447                                                                    TRAP    C$REFG
5628  022260                                    BCOMPLETE       STARST
5629  022260  103413                                                                    BCS     STARST
5630                                    ;SEE IF THIS IS A NEW PASS, BR IF YES
5631  022262                                    READEF  #EF.NEW
5632  022262  012700  000035                                                            MOV     #EF.NEW,R0
5633  022266  104447                                                                    TRAP    C$REFG
5634  022270                                    BCOMPLETE       NEWST
5635  022270  103413                                                                    BCS     NEWST
5636                                    ;SEE IF PROGRAM WAS JUST CONTINUED
5637  022272                                    READEF  #EF.CONTINUE
5638  022272  012700  000036                                                            MOV     #EF.CONTINUE,R0
5639  022276  104447                                                                    TRAP    C$REFG
5640  022300                                    BNCOMPLETE      7$      ;BYPASS JUMP IF COMPLETE FLAG NOT SET
5641  022300  103002                                                                    BCC     7$
5642  022302  000137  022776                    JMP     ENDIT           ;CAUSE 'BCOMPLETE ENDIT'' OUT OF BRANCH RANGE
5643  022306  000416                    7$:     BR      GETPRM
5644  022310                            STARST:
5645  022310  005037  002370                    CLR     STARES          ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
5646                                    ;CLEAR DEVICE MAP
5647  022314  005037  002414                    CLR     DEVMAP
5648  022320                            NEWST:
5649  022320  012737  177777  002316            MOV     #-1,LOGDEV      ;RESET LOGICAL DEVICE TO -1
```

```
5650  022326  005237  002366                  INC     FRSPAS          ;INCREMENT NO. OF PASSES AFTER LOAD
5651  022332  005237  002370                  INC     STARES          ;INCREMENT NO. OF PASSES SINCE STA OR RES
5652  022336  012737  000001  002416          MOV     #BIT0,DEVPTR    ;INIT DEVICE MAP BIT POINTER
5653                                    ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
5654                                    ;   CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
5655  022344                            GETPRM:
5656  022344  005237  002316                  INC     LOGDEV          ;INCREMENT LOGICAL DEVICE NUMBER
5657  022350  023737  002316  002012          CMP     LOGDEV,L$UNIT   ;SEE IF MAXIMUM UNIT NO. EXCEEDED
5658  022356  002360                          BGE     NEWST           ;BR IF YES
5659  022360                                  GPHARD  LOGDEV,R1       ;GET P-TABLE POINTER INTO R1
5660  022360  013700  002316                                                  MOV     LOGDEV,R0
5661  022364  104442                                                          TRAP    C$GPHRD
5662  022366  010001                                                          MOV     R0,R1
5663  022370                                  BCOMPLETE       10$     ;BR IF DEVICE AVAILABLE
5664  022370  103403                                                          BCS     10$
5665  022372  006337  002416                  ASL     DEVPTR          ;SHIFT DEVICE MAP BIT POINTER
5666  022376  000762                          BR      GETPRM          ;SKIP THIS DEVICE
5667  022400  053737  002416  002414  10$:    BIS     DEVPTR,DEVMAP   ;SET BIT FOR THIS DEVICE IN DEVICE MAP
5668  022406  006337  002416                  ASL     DEVPTR          ;SHIFT DEVICE MAP BIT POINTER
5669  022412  011137  002424                  MOV     (R1),MPCSR      ;STORE POINTER TO MICROPROCESSOR CSR'S
5670  022416  011137  002426                  MOV     (R1),BSEL1
5671  022422  005237  002426                  INC     BSEL1           ;GET POINTER TO BSEL1 (MAINTENANCE REGISTER)
5672  022426  011137  002430                  MOV     (R1),SEL4
5673  022432  062737  000004  002430          ADD     #4,SEL4         ;GET POINTER TO SEL4
5674  022440  012137  002432                  MOV     (R1)+,SEL6
5675  022444  062737  000006  002432          ADD     #6,SEL6         ;STORE POINTER TO SEL6
5676  022452  012137  002434                  MOV     (R1)+,LUSWI1    ;GET LU SWITCH PACK #1
5677  022456  012137  002436                  MOV     (R1)+,LUSWI2    ;GET LU SWITCH PACK #2
5678  022462  012137  002440                  MOV     (R1)+,LUSWI3    ;GET LU SWITCH PACK #3
5679  022466  012137  002442                  MOV     (R1)+,TSTCON    ;GET TEST CONNECTOR INDICATOR
5680  022472  013737  002442  002444          MOV     TSTCON,LPBCON   ;PUT TEST CON INDICTR INTO LOOP BACK INDCTR
5681  022500  001417                          BEQ     12$             ;BR IF H3254.5 CONNECTORS
5682  022502  162737  000003  002442          SUB     #3,TSTCON       ;PUT THE FOLLOWING NUMBERS IN LPBCON:
5683  022510  003003                          BGT     11$             ;
5684  022512  012737  000001  002442          MOV     #CBLPBK,TSTCON          ;NON CABLE LOOP BACK    => 0
5685  022520  023727  002444  000004  11$:    CMP     LPBCON,#INTMDM          ;INT MODEM LOOP BACK    => 0
5686  022526  001404                          BEQ     12$             ;H325 CONNECTOR         => 1
5687  022530  023727  002442  000001          CMP     TSTCON,#CBLPBK          ;H3250 CONNECTOR        => 2
5688  022536  001402                          BEQ     13$             ;H3251 CONNECTOR        => 3
5689  022540  005037  002444          12$:    CLR     LPBCON          ;
5690  022544  105037  002450          13$:    CLRB    MLWBYT          ;PREPARE TO SET INITIAL REG 13 WRIBYT VALUE
5691  022550  023727  002442  000002          CMP     TSTCON,#2       ;TEST FOR MODEM LOCAL LOOPBACK
5692  022556  001003                          BNE     30$             ;BR IF NOT MODEM LOCAL LOOPBACK
5693  022560  112737  000110  002450          MOVB    #DTR!MAINT1,MLWBYT      ;KEEP MAINT1 ACTIVE IF LOCAL LOOPBACK
5694  022566  023727  002442  000003  30$:    CMP     TSTCON,#3       ;TEST FOR MODEM REMOTE LOOPBACK
5695  022574  001003                          BNE     32$             ;BR IF NOT MODEM REMOTE LOOPBACK
5696  022576  112737  000104  002450          MOVB    #DTR!MAINT2,MLWBYT      ;KEEP MAINT2 ACTIVE IF REMOTE LOOPBACK
5697                                    ;CAUSE 3 SEC WAIT TO ALLOW MODEM TO SETTLE DOWN FOR LOCAL OR REMOTE LOOPBACK
5698  022604  032737  000002  002442  32$:    BIT     #2,TSTCON       ;TEST FOR MODEM LOOPBACK
5699  022612  001422                          BEQ     34$             ;BR IF MODEM LOOPBACK NOT BEING USED
5700  022614  023727  002370  000001          CMP     STARES,#1       ;TEST FOR FIRST PASS AFTER STA OR RES
5701  022622  001016                          BNE     34$             ;BR IF NOT FIRST PASS
5702  022624                                  PRINTF  #WTMSG          ;PRINT "PLEASE WAIT" MESSAGE
5703  022624  012746  023236                                                  MOV     #WTMSG,-(SP)
5704  022630  012746  000001                                                  MOV     #1,-(SP)
5705  022634  010600                                                          MOV     SP,R0
```

```
5706  022636  104417                                                                    TRAP    C$PNTF
5707  022640  062706  000004                                                            ADD     #4,SP
5708  022644  012702  144444                MOV     #144444,R2      ;INITIALIZE COUNTER FOR 3 SEC WAIT
5709  022650  004737  004634        36$:    JSR     PC,WAIT50       ;DELAY 50 MICRO-SEC
5710  022654  005202                        INC     R2              ;INC WAIT COUNTER
5711  022656  001374                        BNE     36$             ;BR IF TIME NOT UP
5712  022660  011137  002446        34$:    MOV     (R1),BDRATE     ;GET BAUD RATE
5713                                ;SEE IF MANUAL INTERVENTION DESIRED BETWEEN UNITS FOR INSTALLATION OR REMOVAL
5714                                ;   OF TEST CONNECTORS, BR IF NOT
5715  022664  005737  002244                TST     MIFLAG
5716  022670  001442                        BEQ     22$
5717                                ;SEE IF MANUAL INTERVENTION ALLOWED BY SUPERVISOR
5718  022672                                MANUAL
5719  022672  104450                                                                    TRAP    C$MANI
5720                                ;BR IF ALLOWED
5721  022674                                BCOMPLETE       18$
5722  022674  103412                                                                    BCS     18$
5723                                ;PRINT MSG THAT OPERATOR INTERVENTION IS NOT ALLOWED
5724  022676                                PRINTF  #FMT16
5725  022676  012746  023000                                                            MOV     #FMT16,-(SP)
5726  022702  012746  000001                                                            MOV     #1,-(SP)
5727  022706  010600                                                                    MOV     SP,R0
5728  022710  104417                                                                    TRAP    C$PNTF
5729  022712  062706  000004                                                            ADD     #4,SP
5730  022716                        16$:    BREAK                   ;HANG UNTIL ^C TYPED
5731  022716  104422                                                                    TRAP    C$BRK
5732  022720  000776                        BR      16$
5733  022722                        18$:
5734                                ;TYPE ''INSTALL TEST CONNECTOR(S) ON UNIT AT ADRS XXXXXX''
5735  022722                                PRINTF  #FMT17,MPCSR
5736  022722  013746  002424                                                            MOV     MPCSR,-(SP)
5737  022726  012746  023117                                                            MOV     #FMT17,-(SP)
5738  022732  012746  000002                                                            MOV     #2,-(SP)
5739  022736  010600                                                                    MOV     SP,R0
5740  022740  104417                                                                    TRAP    C$PNTF
5741  022742  062706  000006                                                            ADD     #6,SP
5742  022746  005037  002466                        CLR     REG2
5743  022752                        20$:
5744                                ;ASK OPERATOR TO ''TYPE <Y> <CR> WHEN READY TO PROCEED''
5745  022752                                GMANIL  TYPEY,REG2,1,NO
5746  022752  104443                                                                    TRAP    C$GMAN
5747  022754  000404                                                                    BR      10000$
5748  022756  002466                                                                    .WORD   REG2
5749  022760  000120                                                                    .WORD   T$CODE
5750  022762  023206                                                                    .WORD   TYPEY
5751  022764  000001                                                                    .WORD   1
5752  022766                                                                10000$:
5753  022766  023727  002466  000001            CMP     REG2,#1
5754  022774  001366                        BNE     20$
5755  022776                        22$:
5756  022776                        ENDIT:
5757  022776                                ENDINIT
5758  022776                                                                L10015:
5759  022776  104411                                                                    TRAP    C$INIT
5760
5761  023000  047045  040445  040515  FMT16:  .ASCII  /%N%AMANUAL INTERVENTION NOT ALLOWED!%N/
```

```
5762  023006  052516  046101  044440
5763  023014  052116  051105  042526
5764  023022  052116  047511  020116
5765  023030  047516  020124  046101
5766  023036  047514  042527  020504
5767  023044  047045
5768  023046  040445  054524  042520          .ASCIZ  /%ATYPE CONTROL-C (^C) <CR> TO PRCCEED:%N/
5769  023054  041440  047117  051124
5770  023062  046117  041455  024040
5771  023070  041536  020051  041474
5772  023076  037122  052040  020117
5773  023104  051120  041517  042505
5774  023112  035104  047045     000
5775  023117     045  022516  044501  FMT17:  .ASCIZ  /%N%AINSTALL TEST CONNECTOR(S) ON UNIT AT ADRS :  %06%N/
5776  023124  051516  040524  046114
5777  023132  052040  051505  020124
5778  023140  047503  047116  041505
5779  023146  047524  024122  024523
5780  023154  047440  020116  047125
5781  023162  052111  040440  020124
5782  023170  042101  051522  035040
5783  023176  020040  047445  022466
5784  023204  000116
5785  023206  054524  042520  036040  TYPEY:  .ASCIZ  /TYPE <Y><CR> WHEN READY/
5786  023214  037131  041474  037122
5787  023222  053440  042510  020116
5788  023230  042522  042101  000131
5789  023236  047045  047045  040445  WTMSG:  .ASCIZ  /%N%N%A*** PLEASE WAIT ***/
5790  023244  025052  020052  046120
5791  023252  040505  042523  053440
5792  023260  044501  020124  025052
5793  023266  000052
5794                                          .EVEN
5795
5796
5797
5798
5799
```

CZDMSF.P11 30-SEP-81 15:40 AUTO DROP UNIT SECTION

```
5800                                    .SBTTL  AUTO DROP UNIT SECTION
5801
5802                                    ;////////////////////////////////////////////////////////////////////////////
5803                                    ;/ THE AUTO DROP CODING DETERMINES WHETHER OR NOT THE DEVICE WHOSE P-TABLE
5804                                    ;/ WAS JUST OBTAINED IS READY FOR TESTING, AND IT IS DROPPED IF NOT READY.
5805                                    ;////////////////////////////////////////////////////////////////////////////
5806
5807  023270                                    BGNAUTO
5808  023270                                                                                  L$AUTO::
5809                                    ;ESTABLISH PRIORITY = 7
5810  023270                                    SETPRI  #PRI07
5811  023270  012700  000340                                                                 MOV     #PRI07,R0
5812  023274  104441                                                                         TRAP    C$SPRI
5813  023276  012737  023320  000004            MOV     #6$,@#4         ;SET UP NON-EXISTENT MEMORY ERROR TRAP VECTOR
5814  023304  012737  000340  000006            MOV     #PRI07,@#6
5815  023312  005777  157106                    TST     @#PCSR          ;ADDRESS SEL0
5816  023316  000405                            BR      9$              ;TAKE THIS BRANCH IF DEVICE RESPONDS
5817                                    ;COME HERE IF DEVICE CSR IS NON-EXISTENT
5818  023320  062706  000004            6$:     ADD     #4,SP           ;CLEAN UP THE STACK POINTER
5819  023324                                    DODU    LOGDEV          ;DROP THIS UNIT FROM TESTING
5820  023324  013700  002316                                                                 MOV     LOGDEV,R0
5821  023330  104451                                                                         TRAP    C$DODU
5822  023332  013737  002372  000004  9$:      MOV     SAVE4,@#4       ;RESTORE ERROR TRAP VECTOR
5823  023340  013737  002374  000006            MOV     SAVE6,@#6
5824  023346                                    ENDAUTO
5825  023346                                                                                  L10016:
5826  023346  104461                                                                         TRAP    C$AUTO
5827
5828
5829
5830
5831
```

```
CZDMSF.P11    30-SEP-81 15:40          CLEANUP CODING SECTION

  5832                                 .SBTTL  CLEANUP CODING SECTION
  5833
  5834                                 ;////////////////////////////////////////////////////////////////////////
  5835                                 ;/ THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
  5836                                 ;/ AT THE END OF THE TEST SEQUENCE ON A PARTICULAR UNIT.
  5837                                 ;////////////////////////////////////////////////////////////////////////
  5838
  5839  023350                               BGNCLN
  5840  023350                                                                 L$CLEAN::
  5841
  5842
  5843  023350                               ENDCLN
  5844  023350                                                                 L10017:
  5845  023350  104412                                                                TRAP    C$CLEAN
  5846
  5847
  5848
  5849
  5850
```

```
5851                                    .SBTTL  DROP UNIT SECTION
5852
5853                                    ;////////////////////////////////////////////////////////////////////////////
5854                                    ;/ THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
5855                                    ;/ TO NO LONGER BE TESTED.
5856                                    ;////////////////////////////////////////////////////////////////////////////
5857
5858  023352                                    BGNDU
5859  023352                                                                    L$DU::
5860                                    ;ISSUE UNIBUS RESET TO CLEAN UP
5861  023352                                    BRESET
5862  023352  104433                                                                    TRAP    C$RESET
5863                                    ;PRINT 'UNIT XX DROPPED''
5864  023354                                    PRINTF  #FMT27,LOGDEV
5865  023354  013746  002316                                                           MOV     LOGDEV,-(SP)
5866  023360  012746  023402                                                           MOV     #FMT27,-(SP)
5867  023364  012746  000002                                                           MOV     #2,-(SP)
5868  023370  010600                                                                    MOV     SP,R0
5869  023372  104417                                                                    TRAP    C$PNTF
5870  023374  062706  000006                                                           ADD     #6,SP
5871  023400                                    ENDDU
5872  023400                                                                    L10020:
5873  023400  104453                                                                    TRAP    C$DU
5874
5875  023402  047045  040445  047125  FMT27:  .ASCIZ  /%N%AUNIT %D2%A DROPPED%N/
5876  023410  052111  022440  031104
5877  023416  040445  042040  047522
5878  023424  050120  042105  047045
5879  023432     000
5880          023434                    .EVEN
5881
5882
5883
5884
5885
```

```
                                .SBTTL  ADD UNIT SECTION

                                ;////////////////////////////////////////////////////////////////////
                                ;/ THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ;/  TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING.  IF
                                ;/  'EF.AUNIT' IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
                                ;////////////////////////////////////////////////////////////////////

                                        BGNAU
                                                                        L$AU::
                                        ENDAU
                                                                        L10021:
023434  104452                                                                  TRAP    C$AU
```

```
                                        .SBTTL  HARDWARE TESTS





                                        ;;*********************************************************************
                                        .SBTTL       TEST 1 - BIT STUFFING TEST
                                        ;*
                                        ;* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
                                        ;* INITIATED IN BIT MODE . TWO LEADING FLAGS ARE SENT,
                                        ;* FOLLOWED BY ALL SIXTEEN CHARS IN DATA PATTERN S. THIS PATTERN
                                        ;* CONSISTS OF CHARACTERS WHICH REQUIRE NO BIT STUFFING AND CHARACTERS
                                        ;* WHICH REQUIRE BIT STUFFING INDIVIDUALLY AND IN COMBINATION WITH
                                        ;* ADJACENT CHARACTERS. ALL 16 CHARACTERS ARE READ AND COMPARED
                                        ;* BY THE RECEIVER.
                                        ;*   PATTERN S = 000,017,036,074,170,360,037,076,174,370,077,176,374,
                                        ;*               177,376,377
                                        ;;*************************************************************
 023436                                 BGNTST
 023436                                                                                 T1::
 023436  012737  023560  002334                 MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
 023444  004737  005226                         JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
 023450  000000                                 000
 023452  000300                                 CRC2!CRC1               ;BIT MODE, NO ERR DETECTION
 023454  004737  010562                         JSR     PC,LDBYTS       ;LOAD PAT S INTO TX SILO
 023460  002626                                 PATS
 023462  000020                                 16.
 023464  012737  001000  002400                 MOV     #TXEOM,TXWORD
 023472  004737  004662                         JSR     PC,LDTXSI       ;LOAD 2 EOM'S INTO TX SILO
 023476  004737  004662                         JSR     PC,LDTXSI
 023502  004737  004742                         JSR     PC,STPLU        ;CLK MORE THAN ENTIRE MSG
 023506  000300                                 192.
 023510  012701  002626                         MOV     #PATS,R1        ;INIT PAT S POINTER
 023514  112137  023524                 6$:     MOVB    (R1)+,8$
 023520  004737  007302                         JSR     PC,CKDATA       ;CHK A RCV'D CHAR
 023524  000000                         8$:     .WORD   0
 023526  000000                                 0
 023530  020127  002645                         CMP     R1,#PATS+15.    ;SEE IF 15 CHARS CHECKED YET
 023534  103767                                 BLO     6$              ;BR IF NOT YET
 023536  111137  023554                         MOVB    (R1),12$
 023542  052737  001000  023554                 BIS     #RXEBL,12$      ;GET SET TO CHK EBLK = 1
 023550  004737  007302                         JSR     PC,CKDATA       ;CHK LAST CHAR AND EBLK = 1
 023554  000000                         12$:    .WORD   0
 023556  000000                                 0
 023560  004737  003262                 24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
 023564                                 ENDTST
 023564                                                                                 L10022:
 023564  104401                                                                                 TRAP    C$ETST





                                        ;;*************************************************************
                                        .SBTTL       TEST 2 - RCV OVERRUN ERROR SET AND CLEAR TEST
```

```
                                        ;*
                                        ;* IN THIS TEST, A RCV OVERRUN ERROR IS FORCED IN EACH OF 2 SUBTESTS.
                                        ;* IN THE FIRST, A MESSAGE IS INITIATED, 64 001 CHARS ARE SENT, AND THE
                                        ;* RECEIVER IS NOT SERVICED IN RESPONSE TO THE USYRT RCV FLAG, WHICH CAUSES RCV
                                        ;* OVERRUN TO SET. THEN, A CHECK IS MADE TO INSURE THAT OVRR IS NOT
                                        ;* CLEARED BY THE LINE UNIT READING THE USYRT STATUS.
                                        ;* THEN, IC IS SET TO CLEAR THE ERROR, A NEW 001 CHAR IS CLOCKED INTO THE
                                        ;* RECVR, AND THE CLEARING OF OVRR IS VERIFIED.
                                        ;*
                                        ;* IN THE SECOND SUBTEST, RCV OVRUN IS FORCED AGAIN, A MASTER CLEAR
                                        ;* IS ISSUED TO CLEAR THE ERROR, A NEW 001 CHAR IS CLOCKED INTO THE RECVR,
                                        ;* AND THE CLEARING OF OVRR IS VERIFIED.
                                        ;;*********************************************************************
023566                                  BGNTST
023566                                                                  T2::
023566  012737  024216  002334                 MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
                                        ;-------------------------------------------------------------------
                                        ; CAUSE OVRR, SET IC TO CLEAR IT
                                        ;-------------------------------------------------------------------
023574  004737  005226                          JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
023600  000226                                  SYNCH
023602  000311                                  CRC2!CRC1!STRIP!DDCMP   ;DDCMP, NO ERR DET
023604  004737  010734                          JSR     PC,LODSIL       ;LOAD 64 001 CHARS INTO TX SILO
023610  000001                                  001
023612  000100                                  64.
023614  004737  006766                          JSR     PC,RCV1ST       ;CLOCK UNTIL FIRST DATA CHAR RCV'D
023620  000030                                  24.
023622  004737  004742                          JSR     PC,STPLU        ;CLOCK UNTIL 59 MORE RCV'D
023626  000730                                  472.
023630  004737  010734                          JSR     PC,LODSIL       ;LOAD 6 MORE INTO TX SILO
023634  000001                                  001
023636  000006                                  6
023640  004737  004742                          JSR     PC,STPLU        ;CLK 8 MORE TIMES TO FORCE UNDERRUN
023644  000100                                  64.
023646  012701  000100                          MOV     #64.,R1         ;READ AND CHK 64 CHARS FROM RCV SILO
023652  004737  007302          6$:             JSR     PC,CKDATA
023656  000001                                  001
023660  000000                                  0
023662  005301                                  DEC     R1
023664  001372                                  BNE     6$
023666  004737  007302                          JSR     PC,CKDATA       ;READ CHAR, CHK OVRR = 1
023672  004001                                  4001
023674  000010                                  8.
023676  004737  007302                          JSR     PC,CKDATA       ;READ CHAR, CHK OVRR STILL = 1
023702  004001                                  4001
023704  000010                                  8.
023706  012737  000012  002352                  MOV     #12,REGNUM      ;SET REG NO. = 12
023714  012737  000200  002340                  MOV     #IC,WRIBYT
023722  004737  003436                          JSR     PC,WRITLU       ;SET IC TO CLEAR RCVR
023726  012737  000000  002340                  MOV     #0,WRIBYT
023734  004737  003436                          JSR     PC,WRITLU       ;CLEAR IC TO ALLOW RECEIVER TO FUNCTION
023740  004737  010734                          JSR     PC,LODSIL       ;LOAD 2 SOM'S
023744  000626                                  TXSOM!SYNCH
023746  000002                                  2.
023750  004737  010734                          JSR     PC,LODSIL       ;LOAD 2 001 CHARS
023754  000001                                  001
```

```
6017  023756  000002                        2.
6018  023760  004737  006766                JSR     PC,RCV1ST       ;CLOCK UNTIL FIRST DATA CHAR RCV'D
6019  023764  000036                        30.
6020  023766  004737  006706                JSR     PC,RDRXSI       ;READ RCV SILO
6021  023772  132737  000010  002403        BITB    #OVRR,RXWORD+1  ;CHK FOR OVRR CLEARED
6022  024000  001407                        BEQ     8$              ;BR IF OVRR CLEARED
6023  024002  004737  004214                JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
6024                              ;REPORT OVRR NOT CLEARED
6025  024006                                ERRDF   41,EM41,ERR7
6026  024006  104455                                                                TRAP    C$ERDF
6027  024010  000051                                                                .WORD   41
6028  024012  014217                                                                .WORD   EM41
6029  024014  020342                                                                .WORD   ERR7
6030  024016  000477                        BR      24$
6031  024020                              8$:
6032                                      ;-------------------------------------------------------------------
6033                                      ; CAUSE OVRR, SET MST CLR TO CLEAR IT
6034                                      ;-------------------------------------------------------------------
6035  024020  004737  005226                JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
6036  024024  000226                        SYNCH
6037  024026  000311                        CRC2!CRC1!STRIP!DDCMP   ;DDCMP, NO ERR DET
6038  024030  004737  010734                JSR     PC,LODSIL       ;LOAD 64 001 CHARS INTO TX SILO
6039  024034  000001                        001
6040  024036  000100                        64.
6041  024040  004737  006766                JSR     PC,RCV1ST       ;CLOCK UNTIL FIRST DATA CHAR RCV'D
6042  024044  000030                        24.
6043  024046  004737  004742                JSR     PC,STPLU        ;CLOCK UNTIL 59 MORE RCV'D
6044  024052  000730                        472.
6045  024054  004737  010734                JSR     PC,LODSIL       ;LOAD 6 MORE INTO TX SILO
6046  024060  000001                        001
6047  024062  000006                        6.
6048  024064  004737  004742                JSR     PC,STPLU        ;CLK 8 MORE TIMES TO FORCE UNDERRUN
6049  024070  000100                        64.
6050  024072  012701  000100                MOV     #64.,R1         ;READ AND CHK 64 CHARS FROM RCV SILO
6051  024076  004737  007302          9$:   JSR     PC,CKDATA
6052  024102  000001                        001
6053  024104  000000                        0
6054  024106  005301                        DEC     R1
6055  024110  001372                        BNE     9$
6056  024112  004737  007302                JSR     PC,CKDATA       ;READ CHAR, CHK OVRR = 1
6057  024116  004001                        4001
6058  024120  000010                        8.
6059  024122  004737  007302                JSR     PC,CKDATA       ;READ CHAR, CHK OVRR STILL = 1
6060  024126  004001                        4001
6061  024130  000010                        8.
6062  024132  012737  000012  002352        MOV     #12,REGNUM      ;SET REG NO. = 12
6063  024140  004737  005226                JSR     PC,!NITRN       ;MST CLR, LOAD 2 SOM'S
6064  024144  000226                        SYNCH
6065  024146  000311                        CRC2!CRC1!STRIP!DDCMP
6066  024150  004737  010734                JSR     PC,LODSIL       ;LOAD 8 001 CHARS INTO TX SILO
6067  024154  000001                        001
6068  024156  000010                        8.
6069  024160  004737  006766                JSR     PC,RCV1ST       ;CLOCK UNTIL FIRST DATA CHR RCV'D
6070  024164  000030                        24.
6071  024166  004737  006706                JSR     PC,RDRXSI       ;READ RCV SILO
6072  024172  132737  000010  002403        BITB    #OVRR,RXWORD+1  ;CHK FOR OVRR CLEARED
```

```
024200  001406                                  BEQ     24$              ;BR IF OVRR CLEARED
024202  004737  004214                          JSR     PC,GETALL        ;GET REGS FOR PRINTOUT
                                        ;REPORT OVRR NOT CLEARED
024206                                          ERRDF   41,EM41,ERR7
024206  104455                                                                                  TRAP    C$ERDF
024210  000051                                                                                  .WORD   41
024212  014217                                                                                  .WORD   EM41
024214  020342                                                                                  .WORD   ERR7
024216  004737  003262                  24$:    JSR     PC,MSTCLR        ;ISSUE CLEAN UP MST CLR
024222                                  ENDTST
024222                                                                                  L10023:
024222  104401                                                                                  TRAP    C$ETST





                                        ;;*********************************************************************
                                        .SBTTL      TEST 3 - ABORT SEQUENCE TEST
                                        ;*
                                        ;* SET BIT MODE, CRC, AND ENABLE THE DEVICE FOR
                                        ;* TRANSMIT AND RECEIVE. SEND 2 FLAGS AND 4 DATA CHARS (001).
                                        ;* AS THE FIRST DATA CHAR IS BEING TRANSMITTED,
                                        ;* SET THE ABORT BIT (REG 11).
                                        ;* ON THE RECEIVER SIDE, CHECK FOR RECEPTION OF THE FIRST DATA CHAR
                                        ;* AND THEN THE SETTING OF RAB AND REOM A CHAR TIME LATER.
                                        ;* ALSO, CHECK FOR IACT = 0. THEN, CHECK THAT RAB
                                        ;* IS CLEARED BY READING THE USYRT STATUS, TRANSMITTING A NEW MSG,
                                        ;* RECEIVING THE FIRST CHAR (003) AND CHECKING FOR RAB CLEARED.
                                        ;*
                                        ;* REPEAT THE ABOVE SEQUENCE, SET IC, TRANSMIT A NEW MSG,
                                        ;* AND CHECK THAT THIS CLEARS RAB.
                                        ;*
                                        ;;*********************************************************************
024224                                  BGNTST
024224                                                                                  T3::
024224  012737  024450  002334                  MOV     #8$,RETADR       ;SET TEST EXIT ADRS FOR ERRORS
                                        ;-----------------------------------------------------------------------
                                        ; CAUSE ABORT, START NEW MSG TO CLEAR IT
                                        ;-----------------------------------------------------------------------
024232  004737  005226                          JSR     PC,INITRN        ;MST CLR, LOAD 2 SOM'S
024236  000000                                  000
024240  000000                                  000                      ;BIT MODE, CRC
024242  004737  010504                          JSR     PC,LODMSG        ;LOAD MSG INTO TX SILO
024246  002716                                  MSG3
024250  000014                                  12.
024252  004737  006766                          JSR     PC,RCV1ST        ;CLK AND RCV FIRST DATA CHAR
024256  000060                                  48.
024260  004737  007302                          JSR     PC,CKDATA        ;CHK CHR = 001, CLK ABORT CHAR
024264  000001                                  001
024266  000010                                  8.
024270  004737  007302                          JSR     PC,CKDATA        ;CHK FOR RAB, EBLK, AND 001 CHAR
024274  003001                                  RXABT!RXEBL!001
024276  000000                                  0
024300  004737  006246                          JSR     PC,IACTIV        ;CHK FOR IACT = 0
024304  000000                                  0
```

```
6129  024306  004737  006766                  JSR     PC,RCV1ST       ;CLK AND RCV NEW MSG
6130  024312  000060                          48.
6131  024314  004737  007302                  JSR     PC,CKDATA       ;CHK CHAR = 003
6132  024320  000003                          003
6133  024322  000000                          0
6134                                  ;-----------------------------------------------------------------
6135                                  ; CAUSE ABORT, SET IC TO CLEAR IT
6136                                  ;-----------------------------------------------------------------
6137  024324  004737  005226                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
6138  024330  000000                          000
6139  024332  000000                          000                     ;BIT MODE, CRC
6140  024334  004737  010504                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
6141  024340  002716                          MSG3
6142  024342  000014                          12.
6143  024344  004737  006766                  JSR     PC,RCV1ST       ;CLK AND RCV FIRST DATA CHAR
6144  024350  000060                          48.
6145  024352  004737  007302                  JSR     PC,CKDATA       ;CHK CHR = 001, CLK ABORT CHAR
6146  024356  000001                          001
6147  024360  000010                          8.
6148  024362  004737  007302                  JSR     PC,CKDATA       ;CHK FOR RAB, EBLK, AND 001 CHAR
6149  024366  003001                          RXABT!RXEBL!001
6150  024370  000000                          0
6151  024372  012737  000012  002352          MOV     #12,REGNUM      ;SET REG NO. = 12
6152  024400  012737  000200  002340          MOV     #IC,WRIBYT
6153  024406  004737  003436                  JSR     PC,WRITLU       ;SET IC TO CLEAR RCVR
6154  024412  004737  006766                  JSR     PC,RCV1ST       ;CLOCK AND RCV NEW MSG
6155  024416  000060                          48.
6156  024420  004737  006706                  JSR     PC,RDRXSI       ;READ RCV SILO
6157  024424  132737  000004  002403          BITB    #RAB,RXWORD+1   ;CHK FOR RAB CLEARED
6158  024432  001406                          BEQ     8$              ;BR IF RAB CLEARED
6159  024434  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
6160                                  ;REPORT RAB NOT CLEARED
6161  024440                                  ERRDF   39,EM39,ERR7
6162  024440  104455                                                                  TRAP    C$ERDF
6163  024442  000047                                                                  .WORD   39
6164  024444  014167                                                                  .WORD   EM39
6165  024446  020342                                                                  .WORD   ERR7
6166  024450  004737  003262          8$:     JSR     PC,MSTCLR       ;ISSUE MST CLR TO CLEAN UP
6167  024454                          ENDTST
6168  024454                                                                  L10024:
6169  024454  104401                                                                  TRAP    C$ETST
6170
6171
6172
6173
6174
6175                                  ;;*****************************************************************************
6176                                  .SBTTL      TEST 4 - ABORT AND IDLE FLAGS TEST
6177                                  ;*
6178                                  ;* TRANSMIT THE SAME ABORT SEQUENCE AS IN THE PREVIOUS TEST, BUT
6179                                  ;* WITH THE IDLE BIT SET. CHECK THAT FLAGS ARE SENT AND RECEIVED
6180                                  ;* (NOT ABORT CHARACTERS) BY VERIFYING THAT RAB DOES
6181                                  ;* NOT SET, AND THAT THE MESSAGE TERMINATES WITH EBLK = 1.
6182                                  ;;*****************************************************************************
6183  024456                          BGNTST
6184  024456                                                                  , T4::
```

```
6185  024456  012737  024532  002334          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
6186  024464  004737  005226                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
6187  024470  000000                          000
6188  024472  000040                          IDLE                    ;BIT MODE, NO ERROR DET
6189  024474  004737  010504                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
6190  024500  002716                          MSG3
6191  024502  000005                          5
6192  024504  004737  006766                  JSR     PC,RCV1ST       ;CLK AND RCV FIRST DATA CHAR
6193  024510  000060                          48.
6194  024512  004737  007302                  JSR     PC,CKDATA       ;CHK CHR = 001, CLK FLAG CHAR
6195  024516  000001                          001
6196  024520  000010                          8.
6197  024522  004737  007302                  JSR     PC,CKDATA       ;CHK RAB = 0, EBLK = 1
6198  024526  001001                          RXEBL!001
6199  024530  000000                          0
6200  024532  004737  003262          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
6201  024536                          ENDTST
6202  024536                                                          L10025:
6203  024536  104401                                                          TRAP    C$ETST
6204
6205
6206
6207
6208
6209                                  ;;**********************************************************************
6210                                  .SBTTL      TEST 5 - TRANSMITTER UNDERRUN ERROR, IDLE ABORT CHARS, BIT MODE
6211                                  ;*
6212                                  ;* A MESSAGE IS INITIATED IN BIT MODE, 4 001 CHARS ARE SENT, AND THE TRANSMITTER
6213                                  ;* IS NOT SERVICED IN RESPONSE TO THE LAST TX FLAG, WHICH CAUSES TX
6214                                  ;* UNDERRUN ERROR TO SET. ON THE RECEIVER SIDE, CHECK THAT THE DATA
6215                                  ;* CHAR IS RECEIVED, AND THAT 8 CYCLES LATER THE RAB BIT SETS, AND
6216                                  ;* THE DEVICE IDLES ABORT CHARACTERS.
6217                                  ;;**********************************************************************
6218  024540                          BGNTST
6219  024540                                                          T5::
6220  024540  012737  024622  002334          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
6221  024546  004737  005226                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
6222  024552  000000                          000
6223  024554  000000                          000
6224  024556  012737  000100  002404          MOV     #TXEN,DISILO    ;SET TX ENB TO KEEP RTS HIGH
6225  024564  004737  010734                  JSR     PC,LODSIL       ;LOAD 4 001 CHARS INTO TX SILO
6226  024570  000001                          001
6227  024572  000004                          4
6228  024574  004737  006766                  JSR     PC,RCV1ST       ;CLK AND RCV FIRST CHAR
6229  024600  000060                          48.
6230  024602  004737  007302                  JSR     PC,CKDATA       ;CHK DATA = 001, CLOCK ABORT CHAR
6231  024606  000001                          001
6232  024610  000011                          9.
6233  024612  004737  007302                  JSR     PC,CKDATA       ;CHK FOR RAB, EBLK, AND 001 CHAR
6234  024616  003001                          RXABT!RXEBL!001
6235  024620  000000                          0
6236  024622  004737  003262          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
6237  024626                          ENDTST
6238  024626                                                          L10026:
6239  024626  104401                                                          TRAP    C$ETST
6240
```

```
                        ;;*****************************************************************************
                        .SBTTL      TEST 6 - RECEIVER DISABLE TEST
                        ;*
                        ;* TRANSMIT AND RECEIVE ARE ENABLED IN BIT MODE, AND 2 FLAGS
                        ;* ARE SENT, FOLLOWED BY 5 252 DATA CHARS. AFTER THE SECOND DATA CHAR HAS BEGUN
                        ;* TO BE RECEIVED, IC IS SET.
                        ;* THEN, THE PROGRAM CHECKS THAT A USYRT RCV FLAG IS NOT GENERATED, AND
                        ;* THE RECEIVER DATA PATH STOPS OPERATING IN THE MIDDLE OF THE CHAR.
                        ;;*****************************************************************************
024630                  BGNTST
024630                                                                          T6::
024630  012737  024726  002334          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
024636  004737  005226                   JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
024642  000000                           000
024644  000000                           000                     ;BIT MODE, CRC
024646  004737  010734                   JSR     PC,LODSIL       ;LOAD 5 252 CHARS
024652  000252                           252
024654  000005                           5
024656  004737  006766                   JSR     PC,RCV1ST       ;CLK AND RCV FIRST DATA CHAR
024662  000060                           48.
024664  004737  004742                   JSR     PC,STPLU        ;CLK TO MIDDLE OF 2ND CHAR
024670  000004                           4
024672  012737  000012  002352          MOV     #12,REGNUM      ;SET REG NO. = 12
024700  012737  000200  002340          MOV     #IC,WRIBYT
024706  004737  003436                   JSR     PC,WRITLU       ;SET IC IN REG 12
024712  004737  006246                   JSR     PC,IACTIV       ;CHK IACT = 0
024716  000000                           0
024720  004737  005762                   JSR     PC,ISIRDY       ;CHK ICIR = 1, IRDY = 0
024724  000001                           1
024726  004737  003262          24$:     JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
024732                  ENDTST
024732                                                                  L10027:
024732  104401                                                                  TRAP    C$ETST


                        ;;*****************************************************************************
                        .SBTTL      TEST 7 - ASSEMBLED BIT COUNT TEST
                        ;*
                        ;* THE FOLLOWING SEQUENCE IS PERFORMED 8 TIMES, EACH TIME USING A
                        ;* DIFFERENT TX CHAR LENGTH (FROM 2 TO 8 BITS) AND A RCV CHAR LENGTH = 8
                        ;* BITS :
                        ;* A MESSAGE IS INITIATED IN BIT MODE, NO CRC.
                        ;* 2 FLAGS ARE SENT, FOLLOWED BY 3 000 DATA CHARACTERS AND A
                        ;* TERMINATING FLAG. AFTER THE RECEIVER HAS RECEIVED THE MESSAGE, AX0-16
                        ;* IS READ TO RETRIEVE THE ASSEMBLED BIT COUNT. THIS COUNT IS CHECKED TO INSURE
                        ;* THAT IT IS CORRECT FOR THE TX CHAR LENGTH USED IN THAT TRANSMISSION.
                        ;;*****************************************************************************
024734                  BGNTST
024734                                                                          T7::
```

```
6297  024734  012737  025472  002334          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
6298  024742  004737  005226                  JSR     PC,INITRN       ;FIND OUT WHICH USYRT CHIP
6299  024746  000000                          0
6300  024750  000000                          0
6301  024752  012701  000100                  MOV     #TXLEN1,R1      ;SET INITIAL TX LENGTH TO 2 BITS
6302  024756  004737  003262          6$:     JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
6303  024762  004737  010312                  JSR     PC,SETUP        ;PROGRAM THE USYRT
6304  024766  000000                          000
6305  024770  000300                          CRC2!CRC1
6306  024772  000000                          000
6307  024774  000000                          000
6308  024776  012737  000014  002352          MOV     #14,REGNUM      ;SET REG NO. = 14
6309  025004  012737  000140  002340          MOV     #TXEN!DISSI,WRIBYT
6310  025012  004737  003436                  JSR     PC,WRITLU       ;SET TXEN AND DISSI IN REG 14
6311  025016  012737  000140  002404          MOV     #TXEN!DISSI,DISILO      ;SET DISABLE SILO FLAG
6312  025024  012737  000012  002352          MOV     #12,REGNUM      ;SET LU REG NO. = 12
6313  025032  112737  000040  002340          MOVB    #LULP,WRIBYT
6314  025040  004737  003436                  JSR     PC,WRITLU       ;SET LULP IN REG 12
6315  025044  012737  000002  002354          MOV     #2,AXNUM        ;SET AX BYTE NO. = 2
6316  025052  105037  002346                  CLRB    WAX15
6317  025056  112737  000001  002350          MOVB    #TSOM,WAX16
6318  025064  004737  004000                  JSR     PC,WRITAX       ;LOAD SOM CHAR
6319  025070  005004                          CLR     R4              ;INIT COUNTER
6320  025072  012737  000011  002352          MOV     #11,REGNUM      ;SET REG NO. = 11
6321  025100  004737  004742          7$:     JSR     PC,STPLU        ;CLOCK LU FOR A CYCLE
6322  025104  000001                          1
6323  025106  004737  003360                  JSR     PC,READLU       ;READ REG 11
6324  025112  132737  000100  002336          BITB    #OACT,REDBYT    ;SEE IF OACT SET YET
6325  025120  001014                          BNE     10$             ;BR IF OACT SET
6326  025122  005204                          INC     R4              ;INCR COUNTER
6327  025124  020427  000004                  CMP     R4,#4           ;SEE IF COUNT TOO BIG
6328  025130  002763                          BLT     7$              ;BR IF NOT
6329  025132  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
6330                          ;REPORT OACT NOT SET
6331  025136                                  ERRDF   11,EM11,ERR7
6332  025136  104455                                                                  TRAP    C$ERDF
6333  025140  000013                                                                  .WORD   11
6334  025142  013623                                                                  .WORD   EM11
6335  025144  020342                                                                  .WORD   ERR7
6336  025146  000137  025472                  JMP     24$
6337  025152  004737  004000          10$:    JSR     PC,WRITAX       ;LOAD ANOTHER SOM CHAR
6338  025156  004737  004742                  JSR     PC,STPLU        ;CLK FIRST FLAG
6339  025162  000010                          8.
6340  025164  105037  002350                  CLRB    WAX16
6341  025170  004737  004000                  JSR     PC,WRITAX       ;LOAD FIRST 000 CHAR
6342  025174  004737  004742                  JSR     PC,STPLU        ;CLK SECOND FLAG
6343  025200  000010                          8.
6344  025202  004737  004000                  JSR     PC,WRITAX       ;LOAD SECOND 000 CHAR
6345  025206  004737  004742                  JSR     PC,STPLU        ;CLK FIRST 000 CHAR
6346  025212  000010                          8.
6347  025214  012737  000006  002354          MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX 3
6348  025222  010137  002350                  MOV     R1,WAX16        ;GET TX CHAR LENGTH
6349  025226  004737  004000                  JSR     PC,WRITAX       ;SET TX CHAR LENGTH
6350  025232  012737  000002  002354          MOV     #2,AXNUM        ;SET AX BYTE NO. = 2
6351  025240  105037  002350                  CLRB    WAX16
6352  025244  005737  002406                  TST     CHPTYP          ;SEE IF SIG USYRT
```

```
6353  025250  001403                          BEQ     5$                ;BR IF YES
6354  025252  112737  000002  002350          MOVB    #TEOM,WAX16       ;SET TEOM WITH LAST DATA CHAR
6355  025260  004737  004000          5$:     JSR     PC,WRITAX         ;LOAD 3RD 000 CHAR
6356  025264  004737  004742                  JSR     PC,STPLU          ;CLK 2ND 000 CHAR
6357  025270  000010                          8.
6358  025272  005737  002406                  TST     CHPTYP            ;SEE IF SIG USYRT
6359  025276  001005                          BNE     16$               ;BR IF NOT
6360  025300  112737  000002  002350          MOVB    #TEOM,WAX16
6361  025306  004737  004000                  JSR     PC,WRITAX         ;LOAD AN EOM CHAR
6362  025312  012737  000001  002354  16$:    MOV     #1,AXNUM          ;SET AX BYTE NO. = 1
6363  025320  005003                          CLR     R3
6364  025322  004737  003612          12$:    JSR     PC,READAX         ;READ AX0
6365  025326  132737  000002  002344          BITB    #REOM,RAX16       ;CHK FOR REOM = 1
6366  025334  001016                          BNE     14$               ;BR IF YES
6367  025336  004737  004742                  JSR     PC,STPLU          ;CLOCK LU FOR A CYCLE
6368  025342  000001                          1
6369  025344  005203                          INC     R3                ;INCR COUNT
6370  025346  020327  000023                  CMP     R3,#19.           ;SEE IF COUNT TOO BIG
6371  025352  002763                          BLT     12$               ;BR IF NOT
6372  025354  004737  004214                  JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
6373                                  ;REPORT REOM NOT SET
6374  025360                                  ERRDF   31,EM31,ERR10
6375  025360  104455                                                                    TRAP    C$ERDF
6376  025362  000037                                                                    .WORD   31
6377  025364  014042                                                                    .WORD   EM31
6378  025366  021502                                                                    .WORD   ERR10
6379  025370  000440                          BR      24$
6380  025372  013702  002344          14$:    MOV     RAX16,R2          ;GET AX0-16 CONTENTS
6381  025376  042702  000217                  BIC     #217,R2           ;MASK OFF ALL BUT ASSEMB BIT COUNT
6382  025402  006102                          ROL     R2
6383  025404  120201                          CMPB    R2,R1             ;CHK FOR CORRECT ASSEMB BIT COUNT
6384  025406  001421                          BEQ     9$                ;BR IF MATCH
6385  025410  010137  002356                  MOV     R1,GOODAT         ;SET EXPECTED DATA
6386  025414  006237  002356                  ASR     GOODAT
6387  025420  152737  000002  002356          BISB    #REOM,GOODAT
6388  025426  013737  002344  002360          MOV     RAX16,BADDAT      ;SET ACTUAL DATA
6389  025434  004737  004214                  JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
6390                                  ;REPORT ASSEMB BIT COUNT INCORRECT
6391  025440                                  ERRDF   47,EM47,ERR3
6392  025440  104455                                                                    TRAP    C$ERDF
6393  025442  000057                                                                    .WORD   47
6394  025444  014422                                                                    .WORD   EM47
6395  025446  016012                                                                    .WORD   ERR3
6396  025450  000410                          BR      24$
6397  025452  005701          9$:     TST     R1                ;SEE IF ALL DONE YET
6398  025454  001406                          BEQ     24$               ;BR IF YES
6399  025456  062701  000040                  ADD     #TXLEN0,R1        ;INCR TX LENGTH
6400  025462  042701  000400                  BIC     #400,R1           ;MASK OFF OVERFLOW IF 8 BITS
6401  025466  000137  024756                  JMP     6$                ;PROCEED
6402  025472  005037  002404          24$:    CLR     DISILO
6403  025476  004737  003262                  JSR     PC,MSTCLR         ;ISSUE MASTER CLR TO CLEAN UP
6404  025502                          ENDTST
6405  025502                                                                  L10030:
6406  025502  104401                                                                    TRAP    C$ETST
6407
6408
```

```
6409
6410
6411
6412                    ;;*************************************************************************
6413                    .SBTTL      TEST 8 - SECONDARY STATION ADDRESS BIT TEST
6414                    ;*
6415                    ;* FIRST, A MASTER CLEAR IS ISSUED. THEN, THE LINE UNIT IS PLACED IN
6416                    ;* BIT MODE, AND THE SECA BIT (REG 17) IS SET.
6417                    ;* 2 FLAGS ARE SENT, FOLLOWED BY 252, 000, AND A TERMINATING FLAG.
6418                    ;* THEN, THE RECEIVER IS CHECKED TO MAKE SURE THAT NO DATA CHARS ARE
6419                    ;* RECEIVED.
6420                    ;*
6421                    ;* NEXT, THE SECONDARY STATION ADDRESS BITS IN AX2-15 ARE LOADED
6422                    ;* WITH THE FIRST WORD OF DATA PATTERN T . 2 FLAGS ARE SENT,
6423                    ;* FOLLOWED BY THE FIRST WORD OF DATA PATTERN T, A 000 CHAR,
6424                    ;* AND A TERMINATING FLAG.
6425                    ;* THEN, THE RCV'D DATA IS CHECKED TO MAKE SURE THAT THE SEC STATION
6426                    ;* ADDRESS IS RCV'D AS THE FIRST DATA CHAR, FOLLOWED BY 000.
6427                    ;*
6428                    ;* THEN, THE SUBTEST IS REPEATED FOR EACH OF THE REMAINING WORDS OF
6429                    ;* DATA PATTERN T.
6430                    ;*   PATTERN T = 000,125,252,176,177
6431                    ;*************************************************************************
6432  025504           BGNTST
6433  025504                                                                      T8::
6434                    ;-------------------------------------------------------------------
6435                    ; SEND MSG WITH INVALID SEC STA ADRS
6436                    ;-------------------------------------------------------------------
6437  025504                    BGNSUB
6438  025504                                                                      T8.1:
6439  025504  104402                                                                   TRAP    C$BSUB
6440  025506  012737  025604  002334          MOV     #3$,RETADR      ;SET SUBTEST EXIT ADRS FOR ERRORS
6441  025514  004737  005226                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
6442  025520  000000                          000                     ;SEC ADRS = 000
6443  025522  000020                          SECA                    ;BIT MODE, CRC, SEC ADRS MODE
6444  025524  004737  010734                  JSR     PC,LODSIL       ;LOAD 252 INTO TX SILO
6445  025530  000252                          252
6446  025532  000001                          1
6447  025534  004737  010734                  JSR     PC,LODSIL       ;LOAD 000 DATA INTO TX SILO
6448  025540  000000                          000
6449  025542  000001                          1
6450  025544  004737  010734                  JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
6451  025550  001000                          TXEOM
6452  025552  000002                          2
6453  025554  004737  004742                  JSR     PC,STPLU        ;TRANSMIT THE MSG
6454  025560  000060                          48.
6455  025562  004737  006246                  JSR     PC,IACTIV       ;CHK IACT = 0
6456  025566  000000                          0
6457  025570  004737  004742                  JSR     PC,STPLU        ;CLOCK 8 MORE CYCLES
6458  025574  000010                          8.
6459  025576  004737  006246                  JSR     PC,IACTIV       ;CHK IACT = 0
6460  025602  000000                          0
6461  025604           3$:
6462  025604                    ENDSUB
6463  025604                                                                      L10032:
6464  025604  104403                                                                    TRAP    C$ESUB
```

```
                                        ;-----------------------------------------------------------------
                                        ; SEND MSG'S WITH VALID SEC ADRS'S FROM PAT T
                                        ;-----------------------------------------------------------------
025606  012701  002646                          MOV     #PATT,R1        ;INIT DATA PATTERN POINTER
025612                                  A11:
025612                                          BGNSUB
025612                                                                          T8.2:
025612  104402                                                                          TRAP    C$BSUB
025614  012737  025724  002334                  MOV     #24$,RETADR     ;SET SUBTEST EXIT ADRS FOR ERRORS
025622  111137  025642                          MOVB    (R1),5$         ;SET SEC ADRS
025626  111137  025652                          MOVB    (R1),6$         ;SET FIRST DATA CHAR
025632  111137  025710                          MOVB    (R1),9$         ;SET EXPECTED DATA CHAR
025636  004737  005226                          JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
025642  000000                          5$:     .WORD   0
025644  000020                                  SECA                    ;BIT MODE, CRC, SEC ADRS MODE
025646  004737  010734                          JSR     PC,LODSIL       ;LOAD 1ST DATA CHAR INTO TX SILO
025652  000000                          6$:     .WORD   0
025654  000001                                  1
025656  004737  010734                          JSR     PC,LODSIL       ;LOAD A 000 CHAR INTO TX SILO
025662  000000                                  000
025664  000001                                  1
025666  004737  010734                          JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
025672  001000                                  TXEOM
025674  000002                                  2
025676  004737  006766                          JSR     PC,RCV1ST       ;CLOCK AND RCV FIRST DATA CHAR
025702  000060                                  48.
025704  004737  007302                          JSR     PC,CKDATA       ;CHK FOR CORRECT RCV'D SEC STA ADRS
025710  000000                          9$:     .WORD   0
025712  000011                                  9.
025714  004737  007302                          JSR     PC,CKDATA       ;READ AND CHK 000 CHAR, EBLK=1, BCC=0
025720  101000                                  CRCCHK!RXEBL!000
025722  000000                                  0
025724                                  24$:
025724                                          ENDSUB
025724                                                                          L10033:
025724  104403                                                                          TRAP    C$ESUB
025726  005201                                  INC     R1              ;INCR PATTERN POINTER
025730  020127  002653                          CMP     R1,#ENDPAT      ;SEE IF ALL DONE YET
025734  103726                                  BLO     A11             ;BR IF NO
025736  004737  003262                          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
025742                                  ENDTST
025742                                                                          L10031:
025742  104401                                                                          TRAP    C$ETST





                                        ;;*****************************************************************
                                        .SBTTL      TEST 9 - RDALL (ALL PARTIES ADDRESS) BIT TEST
                                        ;*
                                        ;* FIRST, A MASTER CLEAR IS ISSUED. THEN, THE LINE UNIT IS PLACED IN
                                        ;* BIT MODE, AND THE SECA BIT IS SET.
                                        ;* 2 FLAGS ARE SENT, FOLLOWED BY 377, 125, AND A TERMINATING FLAG.
                                        ;* THEN, THE RECEIVER IS CHECKED TO MAKE SURE THAT NO DATA CHARS ARE
                                        ;* RECEIVED.
```

```
                                ;* NEXT, THE RDALL BIT IN REG 17 IS SET TO 1. 2 FLAGS
                                ;* ARE SENT, FOLLOWED BY 377, 125, AND A TERMINATING FLAG.
                                ;* THEN, THE REC'D DATA IS CHECKED TO MAKE SURE THAT 377
                                ;* IS REC'D AS THE FIRST DATA CHAR, FOLLOWED BY 125.
                                ;;*************************************************************************
025744                          BGNTST
025744                                                                          T9::
                                ;-------------------------------------------------------------
                                ; SET SEC ADR = 000, SEND ADR = 377, WITH RDALL = 0
                                ;-------------------------------------------------------------
                                ;
                                        BGNSUB
025744                                                                          T9.1:
025744  104402                                                                          TRAP    C$BSUB
025746  012737  026044  002334          MOV     #3$,RETADR      ;SET SUBTEST EXIT ADRS FOR ERRORS
025754  004737  005226                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
025760  000000                          000                     ;SEC ADRS = 000
025762  000020                          SECA                    ;BIT MODE, CRC, SEC ADRS MODE
025764  004737  010734                  JSR     PC,LODSIL       ;LOAD 377 INTO TX SILO
025770  000377                          377
025772  000001                          1
025774  004737  010734                  JSR     PC,LODSIL       ;LOAD 125 DATA INTO TX SILO
026000  000125                          125
026002  000001                          1
026004  004737  010734                  JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
026010  001000                          TXEOM
026012  000002                          2
026014  004737  004742                  JSR     PC,STPLU        ;TRANSMIT THE MSG
026020  000060                          48.
026022  004737  006246                  JSR     PC,IACTIV       ;CHK IACT = 0
026026  000000                          0
026030  004737  004742                  JSR     PC,STPLU        ;CLOCK 8 MORE CYCLES
026034  000010                          8.
026036  004737  006246                  JSR     PC,IACTIV       ;CHK IACT = 0
026042  000000                          0
026044                          3$:
026044                                  ENDSUB
026044                                                                          L10035:
026044  104403                                                                          TRAP    C$ESUB
                                ;-------------------------------------------------------------
                                ; SET SEC ADR = 000, SEND ADR = 377, WITH RDALL = 1
                                ;-------------------------------------------------------------
                                ;
026046                                  BGNSUB
026046                                                                          T9.2:
026046  104402                                                                          TRAP    C$BSUB
026050  012737  026144  002334          MOV     #24$,RETADR     ;SET SUBTEST EXIT ADRS FOR ERRORS
026056  004737  005226                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
026062  000000                          000                     ;SEC ADRS = 000
026064  000024                          SECA!RDALL              ;BIT MODE, CRC, SEC ADRS MODE, RDALL
026066  004737  010734                  JSR     PC,LODSIL       ;LOAD 1ST DATA CHAR INTO TX SILO
026072  000377                          377
026074  000001                          1
026076  004737  010734                  JSR     PC,LODSIL       ;LOAD A 125 CHAR INTO TX SILO
026102  000125                          125
026104  000001                          1
026106  004737  010734                  JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
026112  001000                          TXEOM
```

```
6577  026114  000002                          2
6578  026116  004737  006766                JSR     PC,RCV1ST       ;CLOCK AND RCV FIRST DATA CHAR
6579  026122  000060                          48.
6580  026124  004737  007302                JSR     PC,CKDATA       ;CHK FOR 377 CHAR RCV'D
6581  026130  000377                          377
6582  026132  000010                          8.
6583  026134  004737  007302                JSR     PC,CKDATA       ;READ AND CHK 125 CHAR, EBLK=1, BCC=0
6584  026140  101125                          CRCCHK!RXEBL!125
6585  026142  000000                          0
6586  026144                          24$:
6587  026144                                  ENDSUB
6588  026144                                                                  L10036:
6589  026144  104403                                                                  TRAP    C$ESUB
6590  026146                          ENDTST
6591  026146                                                                  L10034:
6592  026146  104401                                                                  TRAP    C$ETST
6593
6594
6595
6596
6597
6598                                  ;;*************************************************************************
6599                                  .SBTTL      TEST 10 - INSERT ERROR (IERR) BIT TEST - CHAR MODE, NO CRC
6600                                  ;*
6601                                  ;* THE LINE UNIT IS PLACED IN DDCMP MODE WITH NO ERROR DETECTION, AND 2
6602                                  ;* SYNCHS, A 000 CHAR, A 377 CHAR, AND 2 SYNCHS ARE LOADED INTO THE
6603                                  ;* TRANSMITTER SILO. THEN, THE LU IS CLOCKED UNTIL THE 2ND BIT OF THE 000
6604                                  ;* CHAR IS ABOUT TO BE SENT AND THE IERR BIT IS SET FOR A CLOCK TIME AND
6605                                  ;* THEN CLEARED. IN THE SAME WAY, IERR IS SET PRIOR TO THE SENDING OF THE 4TH
6606                                  ;* AND 5TH BITS OF THE 000 CHAR. IT IS ALSO SET FOR THE SENDING OF THE FIRST
6607                                  ;* 4 BITS OF THE 377 CHAR. THE PROGRAM READS THE FIRST RCV'D CHAR FROM AX0
6608                                  ;* AND CHECKS IT TO BE 032, AND READS THE 2ND CHAR AND CHECKS IT TO BE 377.
6609                                  ;* THEN, A MASTER CLEAR IS DONE TO IDLE THE DEVICE.
6610                                  ;;*************************************************************************
6611  026150                          BGNTST
6612  026150                                                                  T10::
6613  026150  012737  026276  002334          MOV     #15$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
6614  026156  004737  005226                   JSR     PC,INITRN       ;LOAD 2 SOM'S, CLOCK THEM INTO USYRT
6615  026162  000226                          SYNCH
6616  026164  000011                          STRIP!DDCMP
6617  026166  004737  010504                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
6618  026172  002706                          MSG2+4
6619  026174  000004                          4
6620  026176  004737  004742                  JSR     PC,STPLU        ;CLOCK LU UNTIL 2ND BIT OF 000 CHAR
6621  026202  100021                          CHPCHK!17.
6622  026204  004737  007200                  JSR     PC,STPERR       ;SET IERR 1 CYCLE
6623  026210  000011                          STRIP!DDCMP
6624  026212  000001                          1
6625  026214  004737  004742                  JSR     PC,STPLU        ;CLOCK LU UNTIL 4TH BIT OF 000 CHAR
6626  026220  000001                          1
6627  026222  004737  007200                  JSR     PC,STPERR       ;SET IERR FOR 2 CYCLES
6628  026226  000011                          STRIP!DDCMP
6629  026230  000002                          2
6630  026232  004737  004742                  JSR     PC,STPLU        ;CLOCK LU UNTIL 1ST BIT OF 377 CHAR
6631  026236  000003                          3
6632  026240  004737  007200                  JSR     PC,STPERR       ;SET IERR FOR 4 CYCLES
```

```
026244  000011                                  STRIP!DDCMP
026246  000004                                  4
026250  004737  006766                          JSR     PC,RCV1ST       ;CLOCK AND RCV 1ST CHAR
026254  000014                                  12.
026256  004737  007302                          JSR     PC,CKDATA       ;READ AND COMPARE 1ST CHAR TO 032
026262  000032                                  032
026264  000010                                  8.
026266  004737  007302                          JSR     PC,CKDATA       ;READ AND COMPARE 2ND CHAR TO 377
026272  000377                                  377                     ;377 CHAR
026274  000000                                  0
026276  004737  003262                  15$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
026302                                  ENDTST
026302                                                                          L10037:
026302  104401                                                                          TRAP    C$ETST





                                        ;;*************************************************************************
                                        .SBTTL      TEST 11 - SWITCH PACK PRINTOUT AND TEST
                                        ;*
                                        ;* - READ AND PRINT SWITCHES IN REG 11 (E134 SW10,9 , E121 SW9,10) :
                                        ;* THE PROGRAM READS REG 11 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
                                        ;* (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
                                        ;* THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). BITS 1,2
                                        ;* ARE E121 SW10,9 , AND BITS 3,5 ARE E134 SW9,10.
                                        ;*
                                        ;* - READ AND PRINT SWITCHES IN REG 15 (E134 SW8-1) :
                                        ;* THE PROGRAM READS REG 15 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
                                        ;* (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
                                        ;* THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). BITS 0-7 ARE E134 SW1-8.
                                        ;*
                                        ;* - READ AND PRINT SWITCHES IN REG 16 (E121 SW8-1) :
                                        ;* THE PROGRAM READS REG 16 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
                                        ;* (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
                                        ;* THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). BITS 0-7 ARE E121 SW1-8.
                                        ;;*************************************************************************
026304                                  BGNTST
026304                                                                          T11::
                                        ;------------------------------------------------------------------------
                                        ; READ AND PRINT SWITCHES IN REG 11, IF DESIRED
                                        ;------------------------------------------------------------------------
026304                                          BGNSUB
026304                                                                          T11.1:
026304  104402                                                                          TRAP    C$BSUB
026306  004737  003262                          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
026312  012737  000011  002352                  MOV     #11,REGNUM      ;SET LU REG NO. = 11
026320  004737  003360                          JSR     PC,READLU       ;READ LU REG 11
026324  142737  000321  002336                  BICB    #321,REDBYT     ;MASK OFF NON-SWITCH BITS
026332  023727  002366  000001                  CMP     FRSPAS,#1       ;SEE IF IN FIRST PASS AFTER LOAD
026340  001403                                  BEQ     3$              ;BR IF YES
026342  005737  002246                          TST     PRNFLG          ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
026346  001424                                  BEQ     4$              ;BR IF NOT
026350                                  3$:
                                        ;PRINT DEVICE ADDRESS
```

```
                                        PRINTF  #FMT18,MPCSR
                                                                                MOV     MPCSR,-(SP)
                                                                                MOV     #FMT18,-(SP)
                                                                                MOV     #2,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTF
                                                                                ADD     #6,SP
                                ;PRINT REG 11 SWITCHES
                                        PRINTF  #FMT12,REDBYT
                                                                                MOV     REDBYT,-(SP)
                                                                                MOV     #FMT12,-(SP)
                                                                                MOV     #2,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTF
                                                                                ADD     #6,SP
                                4$:     TST     SWIFLG          ;SEE IF TEST IS ALLOWED
                                        BEQ     6$              ;BR IF NOT
                                        CMPB    REDBYT,LUSWI1   ;COMPARE SWITCHES TO EXPECTED
                                        BEQ     6$              ;BR IF MATCH
                                        MOV     LUSWI1,GOODAT   ;SET EXPECTED DATA
                                        MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
                                        JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                ;REPORT REG 11 SWITCHES INCORRECT
                                        ERRDF   43,EM43,ERR2
                                                                                TRAP    C$ERDF
                                                                                .WORD   43
                                                                                .WORD   EM43
                                                                                .WORD   ERR2
                                6$:
                                        ENDSUB
                                                                        L10041:
                                                                                TRAP    C$ESUB
                                ;-------------------------------------------------------------------------
                                ; READ AND PRINT SWITCHES IN REG 15, IF DESIRED
                                ;-------------------------------------------------------------------------
                                        BGNSUB
                                                                                T11.2:
                                                                                TRAP    C$BSUB
                                        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
                                        MOV     #15,REGNUM      ;SET LU REG NO. = 15
                                        JSR     PC,READLU       ;READ LU REG 15
                                        CMP     FRSPAS,#1       ;SEE IF IN FIRST PASS AFTER LOAD
                                        BEQ     3$              ;BR IF YES
                                        TST     PRNFLG          ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
                                        BEQ     4$              ;BR IF NOT
                                3$:
                                ;PRINT REG 15 SWITCHES
                                        PRINTF  #FMT13,REDBYT
                                                                                MOV     REDBYT,-(SP)
                                                                                MOV     #FMT13,-(SP)
                                                                                MOV     #2,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTF
                                                                                ADD     #6,SP
                                4$:     TST     SWIFLG          ;SEE IF TEST IS ALLOWED
                                        BEQ     6$              ;BR IF NOT
```

```
                        CMPB    REDBYT,LUSWI2   ;COMPARE SWITCHES TO EXPECTED
                        BEQ     6$              ;BR IF MATCH
                        MOV     LUSWI2,GOODAT   ;SET EXPECTED DATA
                        MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
                        JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                ;REPORT REG 15 SWITCHES INCORRECT
                        ERRDF   44,EM44,ERR2
                                                                        TRAP    C$ERDF
                                                                        .WORD   44
                                                                        .WORD   EM44
                                                                        .WORD   ERR2
                6$:
                        ENDSUB
                                                                L10042:
                                                                        TRAP    C$ESUB
                ;-----------------------------------------------------------------------
                ; READ AND PRINT SWITCHES IN REG 16, IF DESIRED
                ;-----------------------------------------------------------------------
                        BGNSUB
                                                                T11.3:
                                                                        TRAP    C$BSUB
                        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
                        MOV     #16,REGNUM      ;SET LU REG NO. = 16
                        JSR     PC,READLU       ;READ LU REG 16
                        CMP     FRSPAS,#1       ;SEE IF IN FIRST PASS AFTER LOAD
                        BEQ     3$              ;BR IF YES
                        TST     PRNFLG          ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
                        BEQ     4$              ;BR IF NOT
                3$:
                ;PRINT REG 16 SWITCHES
                        PRINTF  #FMT14,REDBYT
                                                                        MOV     REDBYT,-(SP)
                                                                        MOV     #FMT14,-(SP)
                                                                        MOV     #2,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTF
                                                                        ADD     #6,SP
                4$:     TST     SWIFLG          ;SEE IF TEST IS ALLOWED
                        BEQ     6$              ;BR IF NOT
                        CMPB    REDBYT,LUSWI3   ;COMPARE SWITCHES TO EXPECTED
                        BEQ     6$              ;BR IF MATCH
                        MOV     LUSWI3,GOODAT   ;SET EXPECTED DATA
                        MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
                        JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                ;REPORT REG 16 SWITCHES INCORRECT
                        ERRDF   45,EM45,ERR2
                                                                        TRAP    C$ERDF
                                                                        .WORD   45
                                                                        .WORD   EM45
                                                                        .WORD   ERR2
                6$:
                        ENDSUB
                                                                L10043:
                                                                        TRAP    C$ESUB
                ENDTST
                                                                L10040:
```

```
 6801  026754  104401                                                                    TRAP    C$ETST
 6802
 6803
 6804
 6805
 6806
 6807                          ;;*************************************************************************************
 6808                          .SBTTL      TEST 12 - REG AX3-15 PRINTOUT
 6809                          ;*
 6810                          ;* IN THIS TEST, REG AX3-15 IS READ AND THE CONTENTS PRINTED OUT IF DESIRED BY
 6811                          ;* THE OPERATOR, AS INDICATED IN THE SOFTWARE P-TABLE. THE DEFAULT IS TO NOT
 6812                          ;* PRINT THE REG.
 6813                          ;;*************************************************************************************
 6814  026756                  BGNTST
 6815  026756                                                                                  T12::
 6816  026756  004737  003262          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 6817  026762  142777  000010  153436  BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
 6818  026770  012737  000006  002354  MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX3-15
 6819  026776  004737  003612          JSR     PC,READAX       ;READ AX3-15,AX3-16
 6820  027002  023727  002366  000001  CMP     FRSPAS,#1       ;SEE IF FIRST PASS AFTER LOAD
 6821  027010  001403                  BEQ     3$              ;BR IF YES
 6822  027012  005737  002246          TST     PRNFLG          ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
 6823  027016  001424                  BEQ     4$              ;BR IF NOT
 6824  027020                  3$:
 6825                          ;PRINT DEVICE ADDRESS
 6826  027020                          PRINTF  #FMT18,MPCSR
 6827  027020  013746  002424                                                          MOV     MPCSR,-(SP)
 6828  027024  012746  013325                                                          MOV     #FMT18,-(SP)
 6829  027030  012746  000002                                                          MOV     #2,-(SP)
 6830  027034  010600                                                                  MOV     SP,R0
 6831  027036  104417                                                                  TRAP    C$PNTF
 6832  027040  062706  000006                                                          ADD     #6,SP
 6833                          ;PRINT AX3-15
 6834  027044                          PRINTF  #FMT15,RAX15
 6835  027044  013746  002342                                                          MOV     RAX15,-(SP)
 6836  027050  012746  013256                                                          MOV     #FMT15,-(SP)
 6837  027054  012746  000002                                                          MOV     #2,-(SP)
 6838  027060  010600                                                                  MOV     SP,R0
 6839  027062  104417                                                                  TRAP    C$PNTF
 6840  027064  062706  000006                                                          ADD     #6,SP
 6841  027070                  4$:
 6842  027070                  ENDTST
 6843  027070                                                                  L10044:
 6844  027070  104401                                                                  TRAP    C$ETST
 6845
 6846
 6847
 6848
 6849
 6850                          ;;*************************************************************************************
 6851                          .SBTTL      TEST 13 - CRC GENERATION TEST
 6852                          ;*
 6853                          ;* - CRC-16, CHAR MODE:
 6854                          ;* THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE WITH CRC-16 SELECTED -
 6855                          ;* 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOOP AND STEPLU
 6856                          ;* TO CLOCK THE DATA.    AT THE END OF THE MESSAGE THE
```

```
                                ;* PROGRAM CHECKS FOR BCC = 1 (IN REG 12) INDICATING NO ERROR.
                                ;*
                                ;* - CRC-CCITT - 1'S PRESET:
                                ;* THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED. AT
                                ;* THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.
                                ;*
                                ;* - CRC-CCITT - 0'S PRESET:
                                ;* THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED. AT
                                ;* THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.
                                ;*****************************************************************************
027072                          BGNTST
027072                                                                  T13::
                                ;-------------------------------------------------------------------------
                                ; CRC 16, CHAR MODE
                                ;-------------------------------------------------------------------------
027072  012737  027422  002334          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
027100  004737  003262                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
027104  004737  010312                  JSR     PC,SETUP        ;PROGRAM THE USYRT
027110  000226                          SYNCH
027112  000011                          STRIP!DDCMP
027114  000000                          000
027116  000000                          000
027120  004737  010504                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
027124  002654                          MSG1
027126  000011                          9.
027130  004737  004742                  JSR     PC,STPLU        ;CLOCK THE MSG
027134  000136                          94.
027136  004737  007302                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000
027142  000000                          000
027144  000000                          0
027146  004737  007302                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 125
027152  000125                          125
027154  000000                          0
027156  004737  007302                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 252
027162  000252                          252
027164  000000                          0
027166  004737  007302                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 377
027172  000377                          377
027174  000000                          0
027176  004737  007302                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000, CHK BCC = 1
027202  100400                          CRCCHK!400
027204  000000                          0

                                ;-------------------------------------------------------------------------
                                ; CRC-CCITT-1'S PRESET, BIT MODE
                                ;-------------------------------------------------------------------------
027206  004737  003262                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
027212  004737  010312                  JSR     PC,SETUP        ;PROGRAM THE USYRT
027216  000000                          000
027220  000000                          000
027222  000000                          000
027224  000000                          000
027226  004737  010504                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
027232  002654                          MSG1
027234  000011                          9.
027236  004737  004742                  JSR     PC,STPLU        ;CLOCK THE MSG
```

```
                                        102.
027242  000146                          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000
027244  004737  007302                  000
027250  000000                          0
027252  000000                          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 125
027254  004737  007302                  125
027260  000125                          0
027262  000000                          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 252
027264  004737  007302                  252
027270  000252                          0
027272  000000                          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 377
027274  004737  007302                  377
027300  000377                          0
027302  000000                          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000, CHK BCC = 0
027304  004737  007302                  CRCCHK!1000
027310  101000                          0
027312  000000
                                ;-----------------------------------------------------------
                                ; CRC-CCITT-0'S PRESET, BIT MODE
                                ;-----------------------------------------------------------
027314  004737  003262                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
027320  004737  010312                  JSR     PC,SETUP        ;PROGRAM THE USYRT
027324  000000                          000
027326  000100                          CRC1
027330  000000                          000
027332  000000                          000
027334  004737  010504                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
027340  002654                          MSG1
027342  000011                          9.
027344  004737  004742                  JSR     PC,STPLU        ;CLOCK THE MSG
027350  000146                          102.
027352  004737  007302                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000
027356  000000                          000
027360  000000                          0
027362  004737  007302                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 125
027366  000125                          125
027370  000000                          0
027372  004737  007302                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 252
027376  000252                          252
027400  000000                          0
027402  004737  007302                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 377
027406  000377                          377
027410  000000                          0
027412  004737  007302                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000, CHK BCC = 0
027416  101000                          CRCCHK!1000
027420  000000                          0

027422  004737  003262          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
027426                          ENDTST
027426                                                          L10045:
027426  104401                                                          TRAP    C$ETST
```

```
                                ;;*************************************************************************
                                .SBTTL      TEST 14 - CRC ERROR DETECTION TEST
                                ;*
                                ;* - CRC-16, CHAR MODE :
                                ;* THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE, WITH CRC-16 SELECTED -
                                ;* 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOOP AND STEPLU
                                ;* TO CLOCK THE DATA. JUST BEFORE THE FIRST BIT OF THE LAST 000 CHAR IS SENT,
                                ;* THE IERR BIT IS SET IN REG 17 TO CAUSE A 1 TO BE SENT, INTRODUCING A DATA
                                ;* ERROR. AT THE END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 0, INDICATING
                                ;* AN ERROR.
                                ;*
                                ;* - CRC-CCITT - 1'S PRESET :
                                ;* THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED. AT THE
                                ;* END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
                                ;*
                                ;* - CRC-CCITT - 0'S PRESET :
                                ;* THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED. AT THE
                                ;* END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
                                ;*************************************************************************
027430                          BGNTST
027430                                                                  T14::
                                ;-----------------------------------------
                                ; CRC 16, CHAR MODE
                                ;-----------------------------------------
027430  012737  030032  002334          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
027436  004737  005226                  JSR     PC,INITRN       ;LOAD 2 SOM'S, CLOCK THEM INTO THE USYRT
027442  000226                          SYNCH
027444  000011                          STRIP!DDCMP
027446  004737  005610                  JSR     PC,TXCHAR       ;LOAD 000 CHAR, TX 1ST SYNCH
027452  000000                          000
027454  100010                          CHPCHK!8.
027456  004737  010504                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
027462  002662                          MSG1+6
027464  000006                          6
027466  004737  004742                  JSR     PC,STPLU        ;CLOCK LINE UNIT UNTIL 1ST BIT OF 000 CHAR
027472  000010                          8.
027474  004737  007200                  JSR     PC,STPERR       ;MAKE 1ST BIT = 1 INSTEAD OF 0
027500  000011                          STRIP!DDCMP
027502  000001                          1
027504  004737  00474.                  JSR     PC,STPLU        ;CLOCK REST OF MESSAGE
027510  000122                          82.
027512  004737  007302                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 001 (INTENDED ERROR)
027516  000001                          001
027520  000000                          0
027522  004737  007302                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 125
027526  000125                          125
027530  000000                          0
027532  004737  007302                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 252
027536  000252                          252
027540  000000                          0
027542  004737  007302                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 377
027546  000377                          377
027550  000000                          0
027552  004737  007302                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000, CHK BCC = 0
027556  100000                          CRCCHK!000
```

```
7025  027560  000000                    0
7026
7027                                ;----------------------------------------
7028                                ; CRC-CCITT-1'S PRESET, BIT MODE
7029                                ;----------------------------------------
7030  027562  004737  005226           JSR     PC,INITRN       ;LOAD 2 SOM'S, CLOCK THEM INTO THE USYRT
7031  027566  000000                    000
7032  027570  000000                    000
7033  027572  004737  005610           JSR     PC,TXCHAR       ;LOAD 000 CHAR, TX 1ST FLAG
7034  027576  000000                    000
7035  027600  100010                    CHPCHK!8.
7036  027602  004737  010504           JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
7037  027606  002662                    MSG1+6
7038  027610  000006                    6
7039  027612  004737  004742           JSR     PC,STPLU        ;CLOCK LINE UNIT UNTIL 1ST BIT OF 000 CHAR
7040  027616  000010                    8.
7041  027620  004737  007200           JSR     PC,STPERR       ;MAKE 1ST BIT = 1 INSTEAD OF 0
7042  027624  000000                    000
7043  027626  000001                    1
7044  027630  004737  004742           JSR     PC,STPLU        ;CLOCK REST OF MESSAGE
7045  027634  000122                    82.
7046  027636  004737  007302           JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 001 (INTENDED ERROR)
7047  027642  000001                    001
7048  027644  000000                    0
7049  027646  004737  007302           JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 125
7050  027652  000125                    125
7051  027654  000000                    0
7052  027656  004737  007302           JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 252
7053  027662  000252                    252
7054  027664  000000                    0
7055  027666  004737  007302           JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 377
7056  027672  000377                    377
7057  027674  000000                    0
7058  027676  004737  007302           JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000, CHK BCC = 1
7059  027702  101400                    CRCCHK!1400
7060  027704  000000                    0
7061
7062                                ;----------------------------------------
7063                                ; CRC-CCITT-0'S PRESET, BIT MODE
7064                                ;----------------------------------------
7065  027706  004737  005226           JSR     PC,INITRN       ;LOAD 2 SOM'S, CLOCK THEM INTO THE USYRT
7066  027712  000000                    000
7067  027714  000100                    CRC1
7068  027716  004737  005610           JSR     PC,TXCHAR       ;LOAD 000 CHAR, TX 1ST FLAG
7069  027722  000000                    000
7070  027724  100010                    CHPCHK!8.
7071  027726  004737  010504           JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
7072  027732  002662                    MSG1+6
7073  027734  000006                    6
7074  027736  004737  004742           JSR     PC,STPLU        ;CLOCK LINE UNIT UNTIL 1ST BIT OF 000 CHAR
7075  027742  000010                    8.
7076  027744  004737  007200           JSR     PC,STPERR       ;MAKE 1ST BIT = 1 INSTEAD OF 0
7077  027750  000100                    CRC1
7078  027752  000001                    1
7079  027754  004737  004742           JSR     PC,STPLU        ;CLOCK REST OF MESSAGE
7080  027760  000122                    82.
```

```
 7081  027762  004737  007302          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 001 (INTENDED ERROR)
 7082  027766  000001                  001
 7083  027770  000000                  0
 7084  027772  004737  007302          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 125
 7085  027776  000125                  125
 7086  030000  000000                  0
 7087  030002  004737  007302          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 252
 7088  030006  000252                  252
 7089  030010  000000                  0
 7090  030012  004737  007302          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 377
 7091  030016  000377                  377
 7092  030020  000000                  0
 7093  030022  004737  007302          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000, CHK BCC = 1
 7094  030026  101400                  CRCCHK!1400
 7095  030030  000000                  0
 7096
 7097  030032  004737  003262  24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
 7098  030036                  ENDTST
 7099  030036                                                          L10046:
 7100  030036  104401                                                          TRAP    C$ETST
 7101
 7102
 7103
 7104
 7105
 7106                          ;;**************************************************************************************
 7107                          .SBTTL      TEST 15 - VRC PARITY GENERATION TEST
 7108                          ;*
 7109                          ;* SUBTEST 1 - TEST OF CORRECT ODD VRC PARITY GENERATION :
 7110                          ;* THE LINE UNIT IS PLACED IN CHAR MODE, WITH ODD VRC AND 7-BIT CHARS SELECTED.
 7111                          ;* THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)
 7112                          ;* OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.
 7113                          ;* FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 1 AND FOR THE
 7114                          ;* LAST 4 CHARS IT SHOULD = 0.
 7115                          ;*
 7116                          ;* SUBTEST 2 - TEST OF CORRECT EVEN VRC PARITY GENERATION :
 7117                          ;* THE LINE UNIT IS PLACED IN CHAR MODE, WITH EVEN VRC AND 7-BIT CHARS SELECTED.
 7118                          ;* THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)
 7119                          ;* OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.
 7120                          ;* FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 0 AND FOR THE
 7121                          ;* LAST 4 CHARS IT SHOULD = 1.
 7122                          ;*
 7123                          ;*    DATA PATTERN Q = 000,120,125,137,040,052,057,177
 7124                          ;**************************************************************************************
 7125  030040                  BGNTST
 7126  030040                                                          T15::
 7127                          ;-------------------------------------------------------------
 7128                          ; TEST ODD VRC GENERATION
 7129                          ;-------------------------------------------------------------
 7130  030040  012737  000006  002354          MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX3
 7131  030046  012737  000017  002352          MOV     #17,REGNUM      ;SET REG NO. = 17
 7132  030054                  BGNSUB
 7133  030054                                                          T15.1:
 7134  030054  104402                                                          TRAP    C$BSUB
 7135  030056  012737  030260  002334          MOV     #8$,RETADR      ;SET SUBTEST EXIT ADDRESS FOR ERRORS
 7136  030064  004737  005226                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
```

```
7137  030070  000026                          026
7138  030072  000111                          CRC1!STRIP!DDCMP        ;CHAR MODE, ODD VRC
7139  030074  004737  010562                  JSR     PC,LDBYTS       ;LOAD DATA INTO TX SILO
7140  030100  002577                          PATQ
7141  030102  000010                          8.
7142  030104  004737  010734                  JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
7143  030110  001000                          TXEOM
7144  030112  000002                          2
7145  030114  005037  002346                  CLR     WAX15
7146  030120  012737  000347  002350          MOV     #TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0,WAX16
7147  030126  004737  004000                  JSR     PC,WRITAX       ;SET TX AND RCV LENGTHS = 7
7148  030132  004737  004742                  JSR     PC,STPLU        ;CLOCK FIRST SYNCH
7149  030136  000010                          8.
7150  030140  004737  004742                  JSR     PC,STPLU        ;CLOCK 2ND SYNCH
7151  030144  000010                          8.
7152  030146  005001                          CLR     R1              ;INIT CHAR COUNT
7153  030150  004737  004742          2$:     JSR     PC,STPLU        ;CLOCK A CHAR
7154  030154  000010                          8.
7155  030156  004737  003360                  JSR     PC,READLU       ;READ REG 17
7156  030162  005201                          INC     R1              ;INCR CHAR COUNT
7157  030164  020127  000004                  CMP     R1,#4           ;SEE IF 4 CHARS CLKD YET
7158  030170  003014                          BGT     4$              ;BR IF YES
7159  030172  132737  000040  002336          BITB    #TXDATA,REDBYT  ;SEE IF PARITY BIT IS SET
7160  030200  001024                          BNE     6$              ;BR IF YES
7161  030202  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7162                                  ;REPORT ODD VRC PARITY BIT NOT SET
7163  030206                                  ERRDF   48,EM48,ERR7
7164  030206  104455                                                                  TRAP    C$ERDF
7165  030210  000060                                                                  .WORD   48
7166  030212  014455                                                                  .WORD   EM48
7167  030214  020342                                                                  .WORD   ERR7
7168  030216                                  ESCAPE  SUB
7169  030216  104410                                                                  TRAP    C$ESCAPE
7170  030220  000040                                                                  .WORD   L10050-.
7171  030222  132737  000040  002336  4$:     BITB    #TXDATA,REDBYT  ;SEE IF PARITY BIT IS CLEARED
7172  030230  001410                          BEQ     6$              ;BR IF YES
7173  030232  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7174                                  ;REPORT ODD VRC PARITY BIT NOT CLEARED
7175  030236                                  ERRDF   49,EM49,ERR7
7176  030236  104455                                                                  TRAP    C$ERDF
7177  030240  000061                                                                  .WORD   49
7178  030242  014510                                                                  .WORD   EM49
7179  030244  020342                                                                  .WORD   ERR7
7180  030246                                  ESCAPE  SUB
7181  030246  104410                                                                  TRAP    C$ESCAPE
7182  030250  000010                                                                  .WORD   L10050-.
7183  030252  020127  000010          6$:     CMP     R1,#8.          ;SEE IF ALL CHARS TESTED YET
7184  030256  002734                          BLT     2$              ;BR IF NOT
7185  030260                          8$:
7186  030260                                  ENDSUB
7187  030260                                                                  L10050:
7188  030260  104403                                                                  TRAP    C$ESUB
7189                                  ;-----------------------------------------------------------
7190                                  ; TEST EVEN VRC GENERATION
7191                                  ;-----------------------------------------------------------
7192  030262  012737  000006  002354          MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX3
```

```
7193  030270  012737  000017  002352          MOV     #17,REGNUM      ;SET REG NO. = 17
7194  030276                                  BGNSUB
7195  030276                                                                  T15.2:
7196  030276  104402                                                                  TRAP    C$BSUB
7197  030300  012737  030502  002334          MOV     #18$,RETADR     ;SET SUBTEST EXIT ADRS FOR ERRORS
7198  030306  004737  005226                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
7199  030312  000026                          026
7200  030314  000211                          CRC2!STRIP!DDCMP        ;CHAR MODE, EVEN VRC
7201  030316  004737  010562                  JSR     PC,LDBYTS       ;LOAD DATA INTO TX SILO
7202  030322  002577                          PATQ
7203  030324  000010                          8.
7204  030326  004737  010734                  JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
7205  030332  001000                          TXEOM
7206  030334  000002                          2
7207  030336  005037  002346                  CLR     WAX15
7208  030342  012737  000347  002350          MOV     #TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0,WAX16
7209  030350  004737  004000                  JSR     PC,WRITAX       ;SET TX AND RCV LENGTHS = 7
7210  030354  004737  004742                  JSR     PC,STPLU        ;CLOCK FIRST SYNCH
7211  030360  000010                          8.
7212  030362  004737  004742                  JSR     PC,STPLU        ;CLOCK 2ND SYNCH
7213  030366  000010                          8.
7214  030370  005001                          CLR     R1              ;INIT CHAR COUNT
7215  030372  004737  004742          12$:    JSR     PC,STPLU        ;CLOCK A CHAR
7216  030376  000010                          8.
7217  030400  004737  003360                  JSR     PC,READLU       ;READ REG 17
7218  030404  005201                          INC     R1              ;INCR CHAR COUNT
7219  030406  020127  000004                  CMP     R1,#4           ;SEE IF 4 CHARS CLKD YET
7220  030412  003014                          BGT     14$             ;BR IF YES
7221  030414  132737  000040  002336          BITB    #TXDATA,REDBYT  ;SEE IF PARITY BIT IS CLEARED
7222  030422  001424                          BEQ     16$             ;BR IF YES
7223  030424  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7224                                  ;REPORT EVEN VRC PARITY BIT NOT CLEARED
7225  030430                                  ERRDF   51,EM51,ERR7
7226  030430  104455                                                          TRAP    C$ERDF
7227  030432  000063                                                          .WORD   51
7228  030434  014577                                                          .WORD   EM51
7229  030436  020342                                                          .WORD   ERR7
7230  030440                                  ESCAPE  SUB
7231  030440  104410                                                          TRAP    C$ESCAPE
7232  030442  000040                                                          .WORD   L10051-.
7233  030444  132737  000040  002336  14$:     BITB    #TXDATA,REDBYT  ;SEE IF PARITY BIT IS SET
7234  030452  001010                          BNE     16$             ;BR IF YES
7235  030454  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7236                                  ;REPORT EVEN VRC PARITY BIT NOT SET
7237  030460                                  ERRDF   50,EM50,ERR7
7238  030460  104455                                                          TRAP    C$ERDF
7239  030462  000062                                                          .WORD   50
7240  030464  014543                                                          .WORD   EM50
7241  030466  020342                                                          .WORD   ERR7
7242  030470                                  ESCAPE  SUB
7243  030470  104410                                                          TRAP    C$ESCAPE
7244  030472  000010                                                          .WORD   L10051-.
7245  030474  020127  000010          16$:    CMP     R1,#8.          ;SEE IF ALL CHARS TESTED YET
7246  030500  002734                          BLT     12$             ;BR IF NOT
7247  030502                          18$:
7248  030502                                  ENDSUB
```

```
7249  030502                                                                L10051:
7250  030502  104403                                                                TRAP    C$ESUB
7251  030504  004737  003262          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
7252  030510                  ENDTST
7253  030510                                                                L10047:
7254  030510  104401                                                                TRAP    C$ETST
7255
7256
7257
7258
7259
7260                          ;;*************************************************************************
7261                          .SBTTL      TEST 16 - VRC ERROR DETECTION TEST
7262                          ;*
7263                          ;* SUBTEST 1 - FORCING OF BCC USING ODD VRC
7264                          ;* THE LINE UNIT IS PLACED IN CHAR MODE WITH ODD VRC AND 7-BIT CHARS SELECTED.
7265                          ;* THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER
7266                          ;* 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM
7267                          ;* CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING
7268                          ;* NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE
7269                          ;* RECEIVED (INDICATING AN ERROR).
7270                          ;*
7271                          ;* SUBTEST 2 - FORCING OF BCC USING EVEN VRC
7272                          ;* THE LINE UNIT IS PLACED IN CHAR MODE WITH EVEN VRC AND 7-BIT CHARS SELECTED.
7273                          ;* THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER
7274                          ;* 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM
7275                          ;* CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING
7276                          ;* NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE
7277                          ;* RECEIVED (INDICATING AN ERROR).
7278                          ;*
7279                          ;*   DATA PATTERN R = 000,100,120,124,164,172,176,177,000,100,120,124,164,
7280                          ;*                    172,176.
7281                          ;;*************************************************************************
7282  030512                  BGNTST
7283  030512                                                                T16::
7284                          ;-------------------------------------------------------------------
7285                          ; TEST ODD VRC ERROR DETECTION
7286                          ;-------------------------------------------------------------------
7287  030512  012737  000006  002354          MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX3
7288  030520  012737  000012  002352          MOV     #12,REGNUM      ;SET REG NO.
7289  030526                                  BGNSUB
7290  030526                                                                T16.1:
7291  030526  104402                                                                TRAP    C$BSUB
7292  030530  012737  030746  002334          MOV     #10$,RETADR     ;SET SUBTEST EXIT ADRS FOR ERRORS
7293  030536  004737  005226                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
7294  030542  000026                          026
7295  030544  000111                          CRC1!STRIP!DDCMP        ;CHAR MODE, ODD VRC
7296  030546  004737  010734                  JSR     PC,LODSIL       ;LOAD A THIRD SOM INTO TX SILO
7297  030552  000400                          TXSOM
7298  030554  000001                          1
7299  030556  004737  010562                  JSR     PC,LDBYTS       ;LOAD DATA INTO TX BUFFER
7300  030562  002607                          PATR
7301  030564  000017                          15.
7302  030566  004737  010734                  JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
7303  030572  001000                          TXEOM
7304  030574  000002                          2
```

```
7305  030576  005037  002346                    CLR     WAX15
7306  030602  012737  000347  002350            MOV     #TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0,WAX16
7307  030610  013737  002350  002412            MOV     WAX16,SAVLEN    ;STORE LENGTH 7
7308  030616  004737  004000                    JSR     PC,WRITAX       ;SET TX AND RCV LENGTHS = 7
7309  030622  004737  004742                    JSR     PC,STPLU        ;CLOCK 1ST 8 CHARS, WITH NO ERRORS
7310  030626  000130                            88.
7311  030630  012701  000007                    MOV     #7,R1           ;INIT COUNTER FOR LAST 7 CHARS
7312  030634  004737  007200          3$:       JSR     PC,STPERR       ;ASSERT IERR BIT FOR 1 TIME
7313  030640  000111                            CRC1!STRIP!DDCMP
7314  030642  000001                            1
7315  030644  004737  004742                    JSR     PC,STPLU        ;CLOCK REST OF CHAR
7316  030650  000007                            7
7317  030652  005301                            DEC     R1              ;DECR COUNTER
7318  030654  001367                            BNE     3$              ;BR IF NOT DONE TRANSMITTING YET
7319  030656  004737  004742                    JSR     PC,STPLU        ;CLOCK 2 TERMINATING SYNCHS
7320  030662  000020                            16.
7321  030664  012701  000010                    MOV     #8.,R1          ;INIT COUNTER FOR ERROR-FREE CHARS
7322  030670  012703  002607                    MOV     #PATR,R3        ;INIT DATA PATTERN POINTER
7323  030674  112337  030704          5$:       MOVB    (R3)+,6$        ;GET AN EXPECTED DATA CHAR
7324  030700  004737  007302                    JSR     PC,CKDATA       ;GO CHECK CHAR, CHK BCC=0
7325  030704  100000                6$:       BCCCHK!000
7326  030706  000000                            0
7327  030710  005301                            DEC     R1              ;DECR COUNTER
7328  030712  001370                            BNE     5$              ;BR IF NOT DONE YET
7329  030714  012701  000007                    MOV     #7,R1           ;INIT COUNTER FOR ERROR CHARS
7330  030720  112337  030736          8$:       MOVB    (R3)+,9$        ;GET EXPECTED DATA CHAR
7331  030724  052737  000001  030736            BIS     #BIT0,9$        ;EXPECT ERROR BIT 0 SET
7332  030732  004737  007302                    JSR     PC,CKDATA       ;CHECK DATA, CHK BCC=1
7333  030736  100400                9$:       BCCCHK!RXBCC!000
7334  030740  000000                            0
7335  030742  005301                            DEC     R1              ;DECR COUNTER
7336  030744  001365                            BNE     8$              ;BR IF NOT DONE YET
7337  030746                        10$:
7338  030746                                    ENDSUB
7339  030746                                                                    L10053:
7340  030746  104403                                                                   TRAP    C$ESUB
7341                                ;-------------------------------------------------------------
7342                                ; TEST EVEN VRC ERROR DETECTION
7343                                ;-------------------------------------------------------------
7344  030750  012737  000006  002354            MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX3
7345  030756  012737  000012  002352            MOV     #12,REGNUM      ;SET REG NO.
7346  030764                                    BGNSUB
7347  030764                                                                    T16.2:
7348  030764  104402                                                                   TRAP    C$BSUB
7349  030766  012737  031204  002334            MOV     #30$,RETADR     ;SET SUBTEST EXIT ADRS FOR ERRORS
7350  030774  004737  005226                    JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
7351  031000  000026                            026
7352  031002  000211                            CRC2!STRIP!DDCMP        ;CHAR MODE, EVEN VRC
7353  031004  004737  010734                    JSR     PC,LODSIL       ;LOAD A THIRD SOM INTO TX SILO
7354  031010  000400                            TXSOM
7355  031012  000001                            1
7356  031014  004737  010562                    JSR     PC,LDBYTS       ;LOAD DATA INTO TX BUFFER
7357  031020  002607                            PATR
7358  031022  000017                            15.
7359  031024  004737  010734                    JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
7360  031030  001000                            TXEOM
```

```
7361  031032  000002                          2
7362  031034  005037  002346                  CLR     WAX15
7363  031040  012737  000347  002350          MOV     #TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0,WAX16
7364  031046  013737  002350  002412          MOV     WAX16,SAVLEN    ;STORE LENGTH 7
7365  031054  004737  004000                  JSR     PC,WRITAX       ;SET TX AND RCV LENGTHS = 7
7366  031060  004737  004742                  JSR     PC,STPLU        ;CLOCK 1ST 8 CHARS, WITH NO ERRORS
7367  031064  000130                          88.
7368  031066  012701  000007                  MOV     #7,R1           ;INIT COUNTER FOR LAST 7 CHARS
7369  031072  004737  007200          23$:    JSR     PC,STPERR       ;ASSERT IERR BIT FOR 1 TIME
7370  031076  000211                          CRC2!STRIP!DDCMP
7371  031100  000001                          1
7372  031102  004737  004742                  JSR     PC,STPLU        ;CLOCK REST OF CHAR
7373  031106  000007                          7
7374  031110  005301                          DEC     R1              ;DECR COUNTER
7375  031112  001367                          BNE     23$             ;BR IF NOT DONE TRANSMITTING YET
7376  031114  004737  004742                  JSR     PC,STPLU        ;CLOCK 2 TERMINATING SYNCHS
7377  031120  000020                          16.
7378  031122  012701  000010                  MOV     #8.,R1          ;INIT COUNTER FOR ERROR-FREE CHARS
7379  031126  012703  002607                  MOV     #PATR,R3        ;INIT DATA PATTERN POINTER
7380  031132  112337  031142          25$:    MOVB    (R3)+,26$       ;GET EXPECTED DATA CHAR
7381  031136  004737  007302                  JSR     PC,CKDATA       ;CHK DATA, CHECK BCC=0
7382  031142  100000                  26$:    BCCCHK!000
7383  031144  000000                          0
7384  031146  005301                          DEC     R1              ;DECR COUNTER
7385  031150  001370                          BNE     25$             ;BR IF NOT DONE YET
7386  031152  012701  000007                  MOV     #7,R1           ;INIT COUNTER FOR ERROR CHARS
7387  031156  112337  031174          28$:    MOVB    (R3)+,29$       ;GET EXPECTED DATA CHAR
7388  031162  052737  000001  031174          BIS     #BIT0,29$       ;SET EXPECTED ERROR BIT 0
7389  031170  004737  007302                  JSR     PC,CKDATA       ;CHK DATA, CHK BCC=1
7390  031174  100400                  29$:    BCCCHK!RXBCC!000
7391  031176  000000                          0
7392  031200  005301                          DEC     R1              ;DECR COUNTER
7393  031202  001365                          BNE     28$             ;BR IF NOT DONE YET
7394  031204                          30$:
7395  031204                                  ENDSUB
7396  031204                                                                  L10054:
7397  031204  104403                                                                  TRAP    C$ESUB
7398  031206  004737  003262                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
7399  031212                          ENDTST
7400  031212                                                                  L10052:
7401  031212  104401                                                                  TRAP    C$ETST
7402
7403
7404
7405
7406
7407                                    ;;*****************************************************************************
7408                                    .SBTTL      TEST 17 - INTEGRAL MODEM INTERFACE TEST - CHAR MODE, CRC
7409                                    ;*
7410                                    ;* THE INTEGRAL MODEM IS SELECTED BY THE PROGRAM IN AX3-15, AND A
7411                                    ;* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
7412                                    ;* ON THE LINE UNIT OR AT THE CABLE. THE MESSAGE CONSISTS OF
7413                                    ;* 5 SYNCHS, 000,125,252,377,000, AND  1 SYNCH. IF THE P-TABLE FOR THE CURRENT
7414                                    ;* UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
7415                                    ;* SKIPPED FOR THAT UNIT.
7416                                    ;;*****************************************************************************
```

```
031214                                  BGNTST
031214                                                                                          T17::
031214  012737  000021  002422                  MOV     #17.,TSTNUM       ;SET TEST NO.
031222  012737  031474  002334                  MOV     #24$,RETADR       ;SET TEST EXIT ADDRESS FOR ERRORS
031230  004737  003262                          JSR     PC,MSTCLR         ;ISSUE MASTER CLEAR
031234  004737  011004                          JSR     PC,CKLPBK         ;CHECK LOOPBACK -
031240  000010                                  INTGRL                    ;   SEE IF TEST SHOULD BE RUN
031242  012737  000323  031260                  MOV     #I422!XYZ!V35!OP!TEST,6$ ;SET UP TO SELECT INTEGRAL MODEM
031250  004737  010312                          JSR     PC,SETUP          ;PROGRAM THE USYRT
031254  000226                                  SYNCH
031256  000011                                  STRIP!DDCMP
031260  000000                          6$:     .WORD   0
031262  000000                                  000
031264  004737  010644                          JSR     PC,LDMSG1         ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
031270  142777  000010  151130                  BICB    #LULOOP,@SEL1     ;CLEAR LULOOP, CLOCK MSG
031276  012737  000012  002352                  MOV     #12,REGNUM        ;SET LU REG NO. = 12
031304  012703  002746                          MOV     #RCVBUF,R3        ;GET POINTER TO RCV MSG BUF
031310  013702  002252                  9$:     MOV     TCOUNT,R2             ;INIT TIMER
031314  004737  003360                  10$:    JSR     PC,READLU         ;READ REG 12
031320  132737  000020  002336                  BITB    #IRDY,REDBYT      ;SEE IF IRDY IS SET YET
031326  001011                                  BNE     12$               ;BR IF YES
031330  005202                                  INC     R2                ;INCREMENT TIMER
031332  001370                                  BNE     10$               ;BR IF NO TIME-OUT YET
031334  004737  004214                          JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
                                        ;REPORT IRDY NOT SET
031340                                          ERRDF   17,EM17,ERR7
031340  104455                                                                                  TRAP    C$ERDF
031342  000021                                                                                  .WORD   17
031344  013655                                                                                  .WORD   EM17
031346  020342                                                                                  .WORD   ERR7
031350  000451                                  BR      24$               ;ESCAPE TO END OF TEST
031352  012337  031362                  12$:    MOV     (R3)+,16$
031356  004737  007302                          JSR     PC,CKDATA         ;COMPARE RCV'D DATA CHAR TO EXPECTED
031362  000000                          16$:    0
031364  000000                                  0
031366  020327  002764                          CMP     R3,#RCVBUF+14.    ;SEE IF ALL CHARS CHECKED YET
031372  103746                                  BLO     9$                ;BR IF NOT YET
031374  004737  004634                          JSR     PC,WAIT50         ;STALL FOR 50 MICRO-SEC
031400  004737  005040                          JSR     PC,OACTIV         ;CHECK OACT = 0
031404  000000                                  0
031406  004737  006246                          JSR     PC,IACTIV         ;CHECK IACT STILL = 1
031412  000001                                  1
031414  012737  000013  002352                  MOV     #13,REGNUM        ;SET REG NO. = 13
031422  004737  003360                          JSR     PC,READLU         ;READ REG 13
031426  042737  000232  002336                  BIC     #RING!HDX!MODR!STBY,REDBYT      ;CLR UNUSED BITS
031434  023727  002336  000000                  CMP     REDBYT,#0         ;CHECK REG 13 FOR 000 (MODEM SIGNALS SHOULD BE CLEARED)
031442  001414                                  BEQ     24$               ;BR IF CLEARED
031444  012737  000000  002356                  MOV     #0,GOODAT         ;SET EXPECTED DATA = 0
031452  013737  002336  002360                  MOV     REDBYT,BADDAT     ;SET ACTUAL DATA
031460  004737  004214                          JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
                                        ;REPORT REG MISCOMPARE
031464                                          ERRDF   3,EM3,ERR2
031464  104455                                                                                  TRAP    C$ERDF
031466  000003                                                                                  .WORD   3
031470  013520                                                                                  .WORD   EM3
031472  015504                                                                                  .WORD   ERR2
```

```
 7473  031474                          24$:
 7474  031474                          ENDTST
 7475  031474                                                              L10055:
 7476  031474  104401                                                              TRAP    C$ETST
 7477
 7478
 7479
 7480
 7481
 7482                          ;;*****************************************************************************
 7483                          .SBTTL      TEST 18 - V.35 MODEM INTERFACE TEST - CHAR MODE, CRC
 7484                          ;*
 7485                          ;* THE V.35 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A
 7486                          ;* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
 7487                          ;* ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
 7488                          ;* OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
 7489                          ;* 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
 7490                          ;* UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
 7491                          ;* SKIPPED FOR THAT UNIT.
 7492                          ;;*****************************************************************************
 7493  031476                          BGNTST
 7494  031476                                                              T18::
 7495  031476  012737  000022  002422          MOV     #18.,TSTNUM     ;SET TEST NO.
 7496  031504  012737  031746  002334          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
 7497  031512  004737  003262                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 7498  031516  004737  011004                  JSR     PC,CKLPBK       ;CHECK LOOPBACK -
 7499  031522  000020                          V35                     ;   SEE IF TEST SHOULD BE RUN
 7500  031524  012737  000313  031542          MOV     #I422!XYZ!INTGRL!OP!TEST,6$ ;SET UP TO SELECT V35
 7501  031532  004737  010312                  JSR     PC,SETUP        ;PROGRAM THE USYRT
 7502  031536  000226                          SYNCH
 7503  031540  000011                          STRIP!DDCMP
 7504  031542  000000                  6$:     .WORD   0
 7505  031544  000000                          000
 7506  031546  142777  000010  150652          BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
 7507  031554  012737  000013  002352          MOV     #13,REGNUM      ;SET LU REG NO. = 13
 7508  031562  004737  003360                  JSR     PC,READLU       ;READ REG 13
 7509  031566  132737  000001  002336          BITB    #CARR,REDBYT    ;CHECK FOR CARRIER FALSELY SET
 7510  031574  001415                          BEQ     8$              ;BR IF NOT SET
 7511  031576  012737  000000  002356          MOV     #000,GOODAT     ;SET EXPECTED DATA
 7512  031604  013737  002336  002360          MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
 7513  031612  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 7514                          ;REPORT CARRIER NOT CLEARED
 7515  031616                          ERRDF   66,EM66,ERR7
 7516  031616  104455                                                                      TRAP    C$ERDF
 7517  031620  000102                                                                      .WORD   66
 7518  031622  015122                                                                      .WORD   EM66
 7519  031624  020342                                                                      .WORD   ERR7
 7520  031626  000447                          BR      24$
 7521  031630  152777  000010  150570  8$:     BISB    #LULOOP,@BSEL1  ;SET LULOOP AGAIN
 7522  031636  004737  010644                  JSR     PC,LDMSG1       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
 7523  031642  142777  000010  150556          BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP, CLOCK MSG
 7524  031650  012737  000012  002352          MOV     #12,REGNUM      ;SET LU REG NO. = 12
 7525  031656  012703  002746                  MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
 7526  031662  013702  002252          9$:     MOV     TCOUNT,R2               ;INIT TIMER
 7527  031666  004737  003360          10$:    JSR     PC,READLU       ;READ REG 12
 7528  031672  132737  000020  002336          BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
```

```
7529  031700  001011                          BNE     12$             ;BR IF YES
7530  031702  005202                          INC     R2              ;INCREMENT TIMER
7531  031704  001370                          BNE     10$             ;BR IF NO TIME-OUT YET
7532  031706  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7533                                  ;REPORT IRDY NOT SET
7534  031712                                  ERRDF   17,EM17,ERR7
7535  031712  104455                                                                  TRAP    C$ERDF
7536  031714  000021                                                                  .WORD   17
7537  031716  013655                                                                  .WORD   EM17
7538  031720  020342                                                                  .WORD   ERR7
7539  031722  000411                          BR      24$             ;ESCAPE TO END OF TEST
7540  031724  012337  031734          12$:    MOV     (R3)+,16$
7541  031730  004737  007302                  JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
7542  031734  000000                  16$:    0
7543  031736  000000                          0
7544  031740  020327  002764                  CMP     R3,#RCVBUF+14.  ;SEE IF ALL CHARS CHECKED YET
7545  031744  103746                          BLO     9$              ;BR IF NOT YET
7546  031746                          24$:
7547  031746                          ENDTST
7548  031746                                                                  L10056:
7549  031746  104401                                                                  TRAP    C$ETST
7550
7551
7552
7553
7554
7555                                  ;;*****************************************************************************
7556                                  .SBTTL      TEST 19 - RS 232C AND RS 423 MODEM INTERFACE TEST - CHAR MODE, CRC
7557                                  ;*
7558                                  ;* THE RS232C & RS423 (XYZ) MODEM INTERFACE IS SELECTED BY THE PROGRAM IN
7559                                  ;* AX3-15, AND A MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURN-
7560                                  ;* AROUND CONNECTOR ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
7561                                  ;* OR A MODEM TEST MODE. THE MESSAGE CONSISTS
7562                                  ;* OF 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE
7563                                  ;* P-TABLE FOR THE CURRENT UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS
7564                                  ;* PROVIDED, THE TEST WILL BE SKIPPED FOR THAT UNIT.
7565                                  ;* TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
7566                                  ;* RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
7567                                  ;;*****************************************************************************
7568  031750                          BGNTST
7569  031750                                                                  T19::
7570  031750  012737  000023  002422          MOV     #19.,TSTNUM     ;SET TEST NO.
7571  031756  012737  032132  002334          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
7572  031764  004737  003262                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
7573  031770  004737  011004                  JSR     PC,CKLPBK       ;CHECK LOOPBACK -
7574  031774  000100                          XYZ                     ;   SEE IF TEST SHOULD BE RUN
7575  031776  012737  000233  032014          MOV     #I422!V35!INTGRL!OP!TEST,6$ ;SET UP TO SELECT XYZ
7576  032004  004737  010312                  JSR     PC,SETUP        ;PROGRAM THE USYRT
7577  032010  000226                          SYNCH
7578  032012  000011                          STRIP!DDCMP
7579  032014  000000                  6$:     .WORD   0
7580  032016  000000                          000
7581  032020  004737  012070                  JSR     PC,SETMTM       ;IF MOD LPBK, SET MAINT BIT, TEST FOR TESTMD SET
7582  032024  004737  010644                  JSR     PC,LDMSG1       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
7583  032030  004737  012412                  JSR     PC,TSTMLB       ;TEST FOR CS SET IF MODEM LOOPBACK
7584  032034  012737  000012  002352          MOV     #12,REGNUM      ;SET LU REG NO. = 12
```

```
 7585  032042  012703  002746              MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
 7586  032046  013702  002252      9$:     MOV     TCOUNT,R2               ;INIT TIMER
 7587  032052  004737  003360      10$:    JSR     PC,REA^LU       ;READ REG 12
 7588  032056  132737  000020  002336      BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
 7589  032064  001011                      BNE     12$             ;BR IF YES
 7590  032066  005202                      INC     R2              ;INCREMENT TIMER
 7591  032070  001370                      BNE     10$             ;BR IF NO TIME-OUT YET
 7592  032072  004737  004214              JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 7593                              ;REPORT IRDY NOT SET
 7594  032076                              ERRDF   17,EM17,ERR7
 7595  032076  104455                                                          TRAP    C$ERDF
 7596  032100  000021                                                         .WORD   17
 7597  032102  013655                                                         .WORD   EM17
 7598  032104  020342                                                         .WORD   ERR7
 7599  032106  000411                      BR      24$             ;ESCAPE TO END OF TEST
 7600  032110  012337  032120      12$:    MOV     (R3)+,16$
 7601  032114  004737  007302              JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
 7602  032120  000000              16$:    0
 7603  032122  000000                      0
 7604  032124  020327  002764              CMP     R3,#RCVBUF+14.  ;SEE IF ALL CHARS CHECKED YET
 7605  032130  103746                      BLO     9$              ;BR IF NOT YET
 7606  032132                      24$:
 7607  032132                      ENDTST
 7608  032132                                                              L10057:
 7609  032132  104401                                                          TRAP    C$ETST
 7610
 7611
 7612
 7613
 7614
 7615                              ;;*****************************************************************************
 7616                              ;;
 7617                              .SBTTL      TEST 20 - RS 422 MODEM INTERFACE TEST - CHAR MODE, CRC
 7618                              ;*
 7619                              ;* THE RS 422 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A
 7620                              ;* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
 7621                              ;* ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
 7622                              ;* OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
 7623                              ;* 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
 7624                              ;* UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
 7625                              ;* SKIPPED FOR THAT UNIT.
 7626                              ;* TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
 7627                              ;* RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
 7628  032134                      ;;*****************************************************************************
 7629  032134                      BGNTST
 7630  032134                                                                  T20::
 7631  032134  012737  000024  002422          MOV     #20.,TSTNUM     ;SET TEST NO.
 7632  032142  012737  032316  002334          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
 7633  032150  004737  003262                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 7634  032154  004737  011004                  JSR     PC,CKLPBK       ;CHECK LOOPBACK -
 7635  032160  000200                          I422                    ;   SEE IF TEST SHOULD BE RUN
 7636  032162  012737  000133  032200          MOV     #XYZ!V35!INTGRL!OP!TEST,6$ ;SET UP TO SELECT 422
 7637  032170  004737  010312                  JSR     PC,SETUP        ;PROGRAM THE USYRT
 7638  032174  000226                          SYNCH
 7639  032176  000011                          STRIP!DDCMP
 7640  032200  000000              6$:         .WORD   0
 7641  032202  000000                          000
```

```
 7641  032204  004737  012070                    JSR     PC,SETMTM       ;IF MOD LPBK, SET MAINT BIT, TEST FOR TESTMD SET
 7642  032210  004737  010644                    JSR     PC,LDMSG1       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
 7643  032214  004737  012412                    JSR     PC,TSTMLB       ;TEST FOR CS SET IF MODEM LOOPBACK
 7644  032220  012737  000012  002352            MOV     #12,REGNUM      ;SET LU REG NO. = 12
 7645  032226  012703  002746                    MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
 7646  032232  013702  002252          9$:       MOV     TCOUNT,R2               ;INIT TIMER
 7647  032236  004737  003360          10$:      JSR     PC,READLU       ;READ REG 12
 7648  032242  132737  000020  002336            BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
 7649  032250  001011                            BNE     12$             ;BR IF YES
 7650  032252  005202                            INC     R2              ;INCREMENT TIMER
 7651  032254  001370                            BNE     10$             ;BR IF NO TIME-OUT YET
 7652  032256  004737  004214                    JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 7653                                  ;REPORT IRDY NOT SET
 7654  032262                                    ERRDF   17,EM17,ERR7
 7655  032262  104455                                                                    TRAP    C$ERDF
 7656  032264  000021                                                                    .WORD   17
 7657  032266  013655                                                                    .WORD   EM17
 7658  032270  020342                                                                    .WORD   ERR7
 7659  032272  000411                            BR      24$             ;ESCAPE TO END OF TEST
 7660  032274  012337  032304          12$:      MOV     (R3)+,16$
 7661  032300  004737  007302                    JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
 7662  032304  000000                  16$:      0
 7663  032306  000000                            0
 7664  032310  020327  002764                    CMP     R3,#RCVBUF+14.  ;SEE IF ALL CHARS CHECKED YET
 7665  032314  103746                            BLO     9$              ;BR IF NOT YET
 7666  032316                          24$:
 7667  032316                          ENDTST
 7668  032316                                                                  L10060:
 7669  032316  104401                                                                    TRAP    C$ETST
 7670
 7671
 7672
 7673
 7674
 7675                                  ;;*************************************************************************
 7676                                  .SBTTL      TEST 21 - HALF-DUPLEX BIT (HALF DUPX) TEST
 7677                                  ;*
 7678                                  ;* THIS TEST VERIFIES THAT SETTING HALF-DUPLEX BIT IN REG 13 DOES NOT INHIBIT
 7679                                  ;* LOADING OF THE USYRT TRANSMITTER FROM THE TRANSMITTER SILO.
 7680                                  ;* A MASTER CLEAR IS ISSUED, DDCMP MODE IS ENTERED, AND THE HALF DUPX
 7681                                  ;* BIT IN REG 13 IS SET. A MESSAGE IS LOADED INTO THE TX SILO
 7682                                  ;* CONSISTING OF 2 SYNCHS, 000,125,252,377,000, AND 2 MORE SYNCHS.
 7683                                  ;* THE LINE UNIT IS THEN CLOCKED EXTENSIVELY, AND THE TX SILO IS CHECKED TO
 7684                                  ;* BE UNLOADED (ALL CHARS SHOULD HAVE BEEN REMOVED) AND THE RECEIVER
 7685                                  ;* IS MONITORED TO INSURE THAT NO RCV FLAGS ARE GENERATED.
 7686                                  ;;*************************************************************************
 7687  032320                          BGNTST
 7688  032320                                                                    T21::
 7689  032320  012737  032416  002334            MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
 7690  032326  012737  000013  002352            MOV     #13,REGNUM      ;SET REG NO. = 13
 7691  032334  004737  005226                    JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
 7692  032340  000226                            SYNCH
 7693  032342  000011                            STRIP!DDCMP
 7694  032344  112737  000020  002340            MOVB    #HDX,WRIBYT
 7695  032352  153737  002450  002340            BISB    MLWBYT,WRIBYT   ;ALLOW FOR MAINT1/2 BITS IF MODEM LOOPBACK
 7696  032360  004737  003436                    JSR     PC,WRITLU       ;SET HDX BIT IN REG 13
```

```
7697  032364  004737  010504              JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
7698  032370  002660                      MSG1+4
7699  032372  000007                      7
7700  032374  004737  004742              JSR     PC,STPLU        ;CLK MORE THAN ENTIRE MSG
7701  032400  000136                      94.
7702  032402  004737  004350              JSR     PC,OSIRDY       ;CHK ORDY = 1, OCOR = 0
7703  032406  000001                      1
7704  032410  004737  005762              JSR     PC,ISIRDY       ;CHK ICIR = 1, IRDY = 0
7705  032414  000001                      1
7706  032416  004737  003262      24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
7707  032422              ENDTST
7708  032422                                                              L10061:
7709  032422  104401                                                            TRAP    CSETST
7710
7711
7712
7713
7714
7715                      ;;*************************************************************************
7716                      .SBTTL      TEST 22 - HALF-DUPLEX RCV DISABLED TEST WITH SILOS DISABLED
7717                      ;*
7718                      ;* THIS TEST SENDS A MESSAGE IN HDX, CHAR MODE, WITH NO ERROR DETECTION, AND
7719                      ;* THE SILOS DISABLED. THE MSG CONSISTS OF 2 SYNCHS AND 2 000 CHARS.
7720                      ;* THE DATA IS SENT WITH LULOOP SET FOR TTL DATA LOOPBACK. THE PROGRAM CHECKS
7721                      ;* THAT THE RECEIVER NEVER BECOMES ACTIVE, BECAUSE THE RCV CLOCK IS INHIBITED
7722                      ;* WHEN THE HDX BIT IS SET.
7723                      ;;*************************************************************************
7724  032424              BGNTST
7725  032424                                                              T22::
7726  032424  012737  032630  002334          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
7727  032432  004737  003262                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
7728  032436  004737  010312                  JSR     PC,SETUP        ;PROGRAM USYRT FOR CHAR MODE, NO CRC
7729  032442  000226                          SYNCH
7730  032444  000301                          CRC2!CRC1!DDCMP
7731  032446  000000                          000
7732  032450  000000                          000
7733  032452  012737  000013  002352          MOV     #13,REGNUM      ;SET LU REG NO. = 13
7734  032460  112737  000020  002340          MOVB    #HDX,WRIBYT     ;SET HDX BIT IN REG 13
7735  032466  153737  002450  002340          BISB    MLWBYT,WRIBYT   ;ALLOW FOR MAINT BITS FOR MODEM LOOPBACK
7736  032474  004737  003436                  JSR     PC,WRITLU
7737  032500  012737  000014  002352          MOV     #14,REGNUM      ;SET LU REG NO. = 14
7738  032506  112737  000140  002340          MOVB    #TXEN!DISSI,WRIBYT ;DISABLE SILOS
7739  032514  004737  003436                  JSR     PC,WRITLU
7740  032520  012737  000140  002404          MOV     #TXEN!DISSI,DISILO
7741  032526  012737  000002  002354          MOV     #2,AXNUM        ;SET AX BYTE NO FOR AX1
7742  032534  112737  000000  002346          MOVB    #000,WAX15      ;SET TSOM IN USYRT
7743  032542  112737  000001  002350          MOVB    #TSOM,WAX16
7744  032550  004737  004000                  JSR     PC,WRITAX
7745  032554  004737  004742                  JSR     PC,STPLU        ;CLOCK FIRST SYNCH OUT
7746  032560  000013                          11.
7747  032562  112737  000000  002346          MOVB    #000,WAX15      ;LOAD FIRST 000 DATA CHAR INTO USYRT
7748  032570  112737  000000  002350          MOVB    #000,WAX16
7749  032576  004737  004000                  JSR     PC,WRITAX
7750  032602  004737  004742                  JSR     PC,STPLU        ;CLOCK SECOND SYNCH
7751  032606  000010                          8.
7752  032610  004737  004000                  JSR     PC,WRITAX       ;LOAD SECOND 000 CHAR
```

```
 7753  032614  004737  004742          JSR     PC,STPLU        ;CLOCK FIRST 000 CHAR OUT
 7754  032620  000013                  11.
 7755  032622  004737  006246          JSR     PC,IACTIV       ;CHK FOR IACT = 0 (RECEIVER NOT ACTIVE)
 7756  032626  000000                  0
 7757  032630  005037  002404  24$:    CLR     DISILO          ;CLEAR DISABLE SILO FLAG
 7758  032634  004737  003262          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
 7759  032640                  ENDTST
 7760  032640                                                          L10062:
 7761  032640  104401                                                          TRAP    C$ETST
 7762
 7763
 7764
 7765
 7766                          ;:*****************************************************************************
 7767                          .SBTTL      TEST 23 - INTERACTION OF MODEM CONTROL BITS
 7768                          ;*
 7769                          ;* THIS TEST WILL BE RUN ONLY IF THE P-TABLE FOR THIS UNIT INDICATES THAT EITHER
 7770                          ;* THE H3254 AND H3255, THE H325, THE H3250, OR THE H3251 TEST CONNECTORS ARE
 7771                          ;* INSTALLED.  OTHERWISE, THE TEST WILL BE SKIPPED FOR THE UNIT.
 7772                          ;* SUBTESTS 2 THRU 6 ARE SKIPPED IF AN H325 OR H3250 TEST CONNECTOR IS INSTALLED.
 7773                          ;* THE FOLLOWING SUBTESTS ARE PERFORMED:
 7774                          ;1 - A MASTER CLEAR IS DONE AND REG 13 IS READ AND CHECKED FOR INITIALIZED
 7775                          ;*   STATE, WITH LULOOP SET TO 1. THEN, LULOOP IS CLEARED AND REG 13 IS READ
 7776                          ;*   AND CHECKED FOR THE PROPER STATE, WITH LULOOP CLEARED.
 7777                          ;*   REG 13 IS THEN LOADED WITH 0'S, AND READ AND CHECKED FOR THE INITIALIZED
 7778                          ;*   STATE.
 7779                          ;*   REG 17 IS THEN READ AND CHECKED FOR INITIALIZED STATE.
 7780                          ;2 - RUN IS SET IN BSEL1, AND REG 13 IS READ AND CHECKED FOR RING SET.
 7781                          ;3 - POLL IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR SIGQ SET.
 7782                          ;4 - SELFR IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR SIGR SET.
 7783                          ;5 - MAINT1 IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR TEST MODE SET.
 7784                          ;6 - SELSBY IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR STBY SET.
 7785                          ;7 - DTR IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR DTR AND MODR SET.
 7786                          ;*   IF USING H325 TEST CONNECTOR, REG 13 IS ALSO CHECKED FOR RING SET.
 7787                          ;8 - BPOLL IS SET IN REG 12, ONLY TO LIGHT THE LED FOR THIS SIGNAL.
 7788                          ;9 - HDX IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR HDX SET.
 7789                          ;10 - A MASTER CLEAR IS DONE, 2 TSOM'S ARE LOADED INTO THE TX SILO, THE LINE
 7790                          ;*   UNIT IS CLOCKED UNTIL THE TRANSMITTER IS ACTIVE, AND REG 13 IS READ AND
 7791                          ;*   CHECKED FOR RTS, CS, CARR SET.
 7792                          ;:*****************************************************************************
 7793  032642                  BGNTST
 7794  032642                                                          T23::
 7795  032642  012737  000027  002422          MOV     #23.,TSTNUM     ;SET TEST NO.
 7796  032650  012737  034264  002334          MOV     #A12,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
 7797  032656  005737  002444                  TST     LPBCON          ;CHECK FOR CABLE LOOPBACK
 7798  032662  001003                          BNE     1$              ;SKIP CKLPBK IF CABLE LOOPBACK
 7799  032664  004737  011004                  JSR     PC,CKLPBK       ;SEE IF H3254,5 INSTALLED - SKIP TEST IF NOT
 7800  032670  100000                          TCCHEK
 7801
 7802                          ;-----------------------------------------------------------------------------
 7803                          ; DO MASTER CLEAR, CHK REGS 13,17 FOR INITIALIZED STATES
 7804                          ;-----------------------------------------------------------------------------
 7805  032672                  1$:                             ;LANDING FOR CABLE LOOPBACK BRANCH
 7806  032672                          BGNSUB
 7807  032672                                                          T23.1:
 7808  032672  104402                                                          TRAP    C$BSUB
```

```
7809  032674  004737  003262                    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
7810  032700  012737  000013  002352            MOV     #13,REGNUM      ;SET REG NO. = 13
7811  032706  004737  003360                    JSR     PC,READLU       ;READ REG 13
7812  032712  023727  002336  000210            CMP     REDBYT,#RING!MODR       ;CHECK REG 13 FOR INIT'D STATE
7813  032720  001416                            BEQ     6$              ;BR IF REG 13 INIT'D
7814  032722  012737  000210  002356            MOV     #RING!MODR,GOODAT       ;SET EXPECTED DATA
7815  032730  013737  002336  002360            MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
7816  032736  004737  004214          3$:       JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7817                                  ;REPORT REG MISCOMPARE
7818  032742                                    ERRDF   3,EM3,ERR2
7819  032742  104455                                                                    TRAP    C$ERDF
7820  032744  000003                                                                    .WORD   3
7821  032746  013520                                                                    .WORD   EM3
7822  032750  015504                                                                    .WORD   ERR2
7823  032752                                    ESCAPE  SUB
7824  032752  104410                                                                    TRAP    C$ESCAPE
7825  032754  000220                                                                    .WORD   L10064-.
7826  032756  142777  000010  147442  6$:       BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
7827  032764  004737  003360                    JSR     PC,READLU       ;READ REG 13
7828  032770  023727  002336  000000            CMP     REDBYT,#0       ;CHECK FOR INITIALIZED STATE
7829  032776  001416                            BEQ     8$              ;BR IF OK
7830  033000  012737  000000  002356            MOV     #0,GOODAT       ;GET EXPECTED DATA
7831  033006  013737  002336  002360            MOV     REDBYT,BADDAT   ;GET ACTUAL DATA
7832  033014  004737  004214                    JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7833                                  ;REPORT REG NOT INITIALIZED BY MASTER CLEAR
7834  033020                                    ERRDF   2,EM2,ERR2
7835  033020  104455                                                                    TRAP    C$ERDF
7836  033022  000002                                                                    .WORD   2
7837  033024  013461                                                                    .WORD   EM2
7838  033026  015504                                                                    .WORD   ERR2
7839  033030                                    ESCAPE  SUB
7840  033030  104410                                                                    TRAP    C$ESCAPE
7841  033032  000142                                                                    .WORD   L10064-.
7842  033034  005037  002340          8$:       CLR     WRIBYT          ;SET DATA = 0 TO BE WRITTEN
7843  033040  004737  003436                    JSR     PC,WRITLU       ;LOAD 0'S INTO REG 13
7844  033044  004737  003360                    JSR     PC,READLU       ;READ REG 13
7845  033050  023727  002336  000000            CMP     REDBYT,#000     ;CHECK FOR REG 13 CLEARED
7846  033056  001407                            BEQ     9$              ;BR IF CLEARED
7847  033060  012737  000000  002356            MOV     #000,GOODAT     ;SET EXPECTED DATA
7848  033066  013737  002336  002360            MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
7849  033074  000720                            BR      3$              ;GO PRINT ERROR
7850  033076  012737  000017  002352  9$:       MOV     #17,REGNUM      ;SET REG NO. = 17
7851  033104  004737  003360                    JSR     PC,READLU       ;READ REG 17
7852  033110  042737  000002  002336            BIC     #MCLK,REDBYT    ;IGNORE MCLK BIT
7853  033116  023727  002444  000001            CMP     LPBCON,#H325    ;TEST FOR H325 TEST CONNECTOR
7854  033124  001405                            BEQ     5$              ;BR IF H325 CONNECTOR IS BEING USED
7855  033126  023727  002444  000002            CMP     LPBCON,#H3250   ;TEST FOR H3250 TEST CONNECTOR
7856  033134  001401                            BEQ     5$              ;BR IF H3250 CONNECTOR IS BEING USED
7857  033136  000403                            BR      7$              ;BR IF NEITHER H325 OR H3250
7858  033140  042737  000300  002336  5$:       BIC     #SIGR!SIGQ,REDBYT       ;MASK OUT SIGR AND SIGQ
7859  033146  123727  002336  000051  7$:       CMPB    REDBYT,#TXDATA!ICIR!DDCMP ;CHK REG 17 FOR INIT'D STATE
7860  033154  001407                            BEQ     10$             ;BR IF REG 17 INITIALIZED
7861  033156  012737  000051  002356            MOV     #TXDATA!ICIR!DDCMP,GOODAT ;SET EXPECTED DATA
7862  033164  013737  002336  002360            MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
7863  033172  000661                            BR      3$              ;GO REPORT ERROR
7864  033174                          10$:
```

```
                                        ENDSUB
                                                                        L10064:
                                                                                TRAP    C$ESUB

                                        CMP     LPBCON,#H325    ;TEST FOR H325 CONNECTOR
                                        BNE     4$              ;BYPASS JUMP (CAUSING JUMP ON ZERO)
                                        JMP     STS237          ;BRANCH TO SUBTEST 7 IF H325 CONNECTOR
                                4$:     CMP     LPBCON,#H3250   ;TEST FOR H3250 CONNECTOR
                                        BNE     5$              ;BYPASS JUMP (CAUSING JUMP ON ZERO)
                                        JMP     STS237          ;BRANCH TO SUBTEST 7 IF H3250 CONNECTOR
                                5$:                             ;LANDING FOR BRANCH AROUND JUMP
                                ;-------------------------------------------------------------------------
                                ; SET RUN IN BSEL1, CHECK FOR RING SET IN REG 13.  SKIPPED IF H325 OR H3250
                                ; TEST CONNECTORS SELECTED.  THIS SUBTEST CHECKS ONLY INTERNAL CIRCUITRY--
                                ; NEITHER THE CABLE, THE DRIVERS, NOR THE RECEIVERS ARE TESTED.
                                ;-------------------------------------------------------------------------
                                        BGNSUB
                                                                        T23.2:
                                                                                TRAP    C$BSUB
                                        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
                                        CLRB    @BSEL1          ;CLEAR LULOOP
                                        MOVB    #RUN,@BSEL1     ;SET RUN BIT IN BSEL1
                                        MOVB    #LULOOP,@BSEL1  ;CLEAR RUN, SET LULOOP
                                        MOV     #13,REGNUM      ;SET REG NO. = 13
                                        JSR     PC,READLU       ;READ REG 13
                                        BITB    #RING,REDBYT    ;SEE IF RING = 1
                                        BNE     9$              ;BR IF RING = 1
                                        JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                ;REPORT RING NOT SET
                                        ERRDF   56,EM56,ERR7
                                                                                TRAP    C$ERDF
                                                                                .WORD   56
                                                                                .WORD   EM56
                                                                                .WORD   ERR7
                                        ESCAPE  SUB
                                                                                TRAP    C$ESCAPE
                                                                                .WORD   L10065-.
                                9$:
                                        ENDSUB
                                                                        L10065:
                                                                                TRAP    C$ESUB

                                ;-------------------------------------------------------------------------
                                ; SET POLL IN REG 13, CHK FOR SIGQ SET IN REG 17.  SKIPPED FOR H325 AND H3250
                                ;-------------------------------------------------------------------------
                                        BGNSUB
                                                                        T23.3:
                                                                                TRAP    C$BSUB
                                        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
                                        MOVB    #POLL,WRIBYT
                                        MOV     #13,REGNUM      ;SET REG NO. = 13
                                        JSR     PC,WRITLU       ;SET POLL IN REG 13
                                        MOV     #17,REGNUM      ;SET REG NO. = 17
                                        JSR     PC,READLU       ;READ REG 17
                                        BITB    #SIGQ,REDBYT    ;SEE IF SIGQ = 1
                                        BNE     6$              ;BR IF SIGQ = 1
```

```
7921  033370  004737  004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7922                                ;REPORT SIGQ NOT SET
7923  033374                        ERRDF   63,EM63,ERR7
7924  033374  104455                                                  TRAP    C$ERDF
7925  033376  000077                                                  .WORD   63
7926  033400  015054                                                  .WORD   EM63
7927  033402  020342                                                  .WORD   ERR7
7928  033404                        6$:
7929  033404                        ENDSUB
7930  033404                                                      L10066:
7931  033404  104403                                                  TRAP    C$ESUB
7932                                ;-------------------------------------------------------------------------
7933                                ; SET SELFR IN REG 13, CHK FOR SIGR SET IN REG 17.  SKIPPED FOR H325 AND H3250
7934                                ;-------------------------------------------------------------------------
7935  033406                        BGNSUB
7936  033406                                                      T23.4:
7937  033406  104402                                                  TRAP    C$BSUB
7938  033410  004737  003262        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
7939  033414  112737  000040  002340  MOVB    #SELFR,WRIBYT
7940  033422  012737  000013  002352  MOV     #13,REGNUM      ;SET REG NO. = 13
7941  033430  004737  003436          JSR     PC,WRITLU       ;SET SELFR IN REG 13
7942  033434  012737  000017  002352  MOV     #17,REGNUM      ;SET REG NO. = 17
7943  033442  004737  003360          JSR     PC,READLU       ;READ REG 17
7944  033446  132737  000200  002336  BITB    #SIGR,REDBYT    ;SEE IF SIGR = 1
7945  033454  001006                  BNE     6$              ;BR IF SIGR = 1
7946  033456  004737  004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7947                                ;REPORT SIGR NOT SET
7948  033462                        ERRDF   64,EM64,ERR7
7949  033462  104455                                                  TRAP    C$ERDF
7950  033464  000100                                                  .WORD   64
7951  033466  015071                                                  .WORD   EM64
7952  033470  020342                                                  .WORD   ERR7
7953  033472                        6$:
7954  033472                        ENDSUB
7955  033472                                                      L10067:
7956  033472  104403                                                  TRAP    C$ESUB
7957                                ;-------------------------------------------------------------------------
7958                                ; SET MAINT1 IN REG 13, CHK FOR TEST MODE SET IN REG 17.  SKIPPED FOR H325 AND H3250
7959                                ;-------------------------------------------------------------------------
7960  033474                        BGNSUB
7961  033474                                                      T23.5:
7962  033474  104402                                                  TRAP    C$BSUB
7963  033476  004737  003262        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
7964  033502  112737  000010  002340  MOVB    #MAINT1,WRIBYT
7965  033510  012737  000013  002352  MOV     #13,REGNUM      ;SET REG NO. = 13
7966  033516  004737  003436          JSR     PC,WRITLU       ;SET MAINT1 IN REG 13
7967  033522  012737  000017  002352  MOV     #17,REGNUM      ;SET REG NO. = 17
7968  033530  142777  000010  146670  BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
7969  033536  004737  003360          JSR     PC,READLU       ;READ REG 17
7970  033542  132737  000004  002336  BITB    #TESTMD,REDBYT  ;SEE IF TESTMD = 1
7971  033550  001006                  BNE     6$              ;BR IF TESTMD = 1
7972  033552  004737  004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7973                                ;REPORT TEST MODE NOT SET BY MAINT1
7974  033556                        ERRDF   52,EM52,ERR7
7975  033556  104455                                                  TRAP    C$ERDF
7976  033560  000064                                                  .WORD   52
```

```
                                                                                .WORD   EM52
                                                                                .WORD   ERR7
                        6$:
                                ENDSUB
                                                                        L10070:
                                                                                TRAP    C$ESUB

                        ;-------------------------------------------------------------------------------
                        ; SET SELSBY IN REG 13, CHK FOR STBY SET IN REG 13.  SKIPPED FOR H325 AND H3250
                        ;-------------------------------------------------------------------------------
                                BGNSUB
                                                                        T23.6:
                                                                                TRAP    C$BSUB
033570  104402
033572  004737  003262          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
033576  112737  000002  002340  MOVB    #SELSBY,WRIBYT
033604  012737  000013  002352  MOV     #13,REGNUM      ;SET REG NO. = 13
033612  004737  003436          JSR     PC,WRITLU       ;SET SELSBY IN REG 13
033616  004737  003360          JSR     PC,READLU       ;READ REG 13
033622  132737  000002  002336  BITB    #STBY,REDBYT    ;SEE IF STBY = 1
033630  001006                  BNE     6$              ;BR IF STBY = 1
033632  004737  004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                        ;REPORT STBY NOT SET
033636                  ERRDF   59,EM59,ERR7
033636  104455                                                          TRAP    C$ERDF
033640  000073                                                          .WORD   59
033642  014773                                                          .WORD   EM59
033644  020342                                                          .WORD   ERR7
                        6$:
                                ENDSUB
                                                                        L10071:
033646  104403                                                                TRAP    C$ESUB

                        ;-------------------------------------------------------------------------------
                        ; SET DTR IN REG 13, CHECK FOR DTR AND MODR SET IN REG 13
                        ; IF H325 TEST CONNECTOR IS SELECTED, REG 13 IS ALSO TESTED FOR RING SET.
                        ;-------------------------------------------------------------------------------

033650                  STS237:                         ;TEST STARTS HERE FOR H325 AND H3250 CONNECTORS

                                BGNSUB
                                                                        T23.7:
                                                                                TRAP    C$BSUB
033650  104402
033652  004737  003262          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
033656  012737  000013  002352  MOV     #13,REGNUM      ;SET REG NO. = 13
033664  112737  000104  002340  MOVB    #DTR!MAINT2,WRIBYT
033672  004737  003436          JSR     PC,WRITLU       ;SET DTR IN REG 13
                                                        ;  (ALSO SET MAINT2 FOR MANUFACT. TEST CONN.)
033676  142777  000010  146522  BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
033704  004737  003360          JSR     PC,READLU       ;READ REG 13
033710  132737  000100  002336  BITB    #DTR,REDBYT     ;SEE IF DTR = 1
033716  001010                  BNE     6$              ;BR IF DTR = 1
033720  004737  004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                        ;REPORT DTR NOT SET
033724                  ERRDF   55,EM55,ERR7
033724  104455                                                          TRAP    C$ERDF
```

```
8033  033726  000067                                                          .WORD   55
8034  033730  014711                                                          .WORD   EM55
8035  033732  020342                                                          .WORD   ERR7
8036  033734                          ESCAPE  SUB
8037  033734  104410                                                          TRAP    C$ESCAPE
8038  033736  000062                                                          .WORD   L10072-.
8039  033740  132737  000010  002336 6$:     BITB    #MODR,REDBYT    ;SEE IF MODR = 1
8040  033746  001006                  BNE     12$             ;BR IF MODR = 1
8041  033750  004737  004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
8042                                  ;REPORT MODR NOT SET
8043  033754                          ERRDF   57,EM57,ERR7
8044  033754  104455                                                          TRAP    C$ERDF
8045  033756  000071                                                          .WORD   57
8046  033760  014742                                                          .WORD   EM57
8047  033762  020342                                                          .WORD   ERR7
8048  033764  023727  002444  000001 12$:     CMP     LPBCON,#H325    ;TEST FOR H325 TEST CONNECTOR
8049  033772  001012                  BNE     13$             ;SKIP THE NEXT BIT CHECK IF NOT H325
8050  033774  132737  000200  002336          BITB    #RING,REDBYT    ;CHECK TO VERIFY RING BIT SET
8051  034002  001006                  BNE     13$             ;BR IF RING SET
8052  034004  004737  004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
8053                                  ;REPORT RING NOT SET
8054  034010                          ERRDF   67,EM56,ERR7
8055  034010  104455                                                          TRAP    C$ERDF
8056  034012  000103                                                          .WORD   67
8057  034014  014725                                                          .WORD   EM56
8058  034016  020342                                                          .WORD   ERR7
8059  034020                  13$:
8060  034020                          ENDSUB
8061  034020                                                          L10072:
8062  034020  104403                                                          TRAP    C$ESUB
8063
8064                          ;-----------------------------------------------------------------
8065                          ; SET BPOLL IN REG 12, TO LIGHT LED ONLY
8066                          ;-----------------------------------------------------------------
8067  034022                          BGNSUB
8068  034022                                                          T23.8:
8069  034022  104402                                                          TRAP    C$BSUB
8070  034024  004737  003262          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
8071  034030  012737  000012  002352  MOV     #12,REGNUM      ;SET LU REG NO. = 12
8072  034036  112737  000100  002340  MOVB    #BPOLL,WRIBYT   ;SET BPOLL IN LU REG 12
8073  034044  004737  003436          JSR     PC,WRITLU
8074  034050                          ENDSUB
8075  034050                                                          L10073:
8076  034050  104403                                                          TRAP    C$ESUB
8077
8078                          ;-----------------------------------------------------------------
8079                          ; SET HDX IN REG 13, CHK FOR HDX SET IN REG 13
8080                          ;-----------------------------------------------------------------
8081  034052                          BGNSUB
8082  034052                                                          T23.9:
8083  034052  104402                                                          TRAP    C$BSUB
8084  034054  004737  003262          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
8085  034060  112737  000020  002340  MOVB    #HDX,WRIBYT
8086  034066  012737  000013  002352  MOV     #13,REGNUM      ;SET REG NO. = 13
8087  034074  004737  003436          JSR     PC,WRITLU       ;SET HDX IN REG 13
8088  034100  004737  003360          JSR     PC,READLU       ;READ REG 13
```

```
CZDMSF.P11    30-SEP-81 15:40              TEST 23 - INTERACTION OF MODEM CONTROL BITS

 8089  034104  132737  000020  002336          BITB    #HDX,REDBYT     ;SEE IF HDX = 1
 8090  034112  001006                          BNE     6$              ;BR IF HDX = 1
 8091  034114  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 8092                                  ;REPORT HDX NOT SET
 8093  034120                                  ERRDF   58,EM58,ERR7
 8094  034120  104455                                                          TRAP    C$ERDF
 8095  034122  000072                                                          .WORD   58
 8096  034124  014757                                                          .WORD   EM58
 8097  034126  020342                                                          .WORD   ERR7
 8098  034130                          6$:
 8099  034130                                  ENDSUB
 8100  034130                                                        L10074:
 8101  034130  104403                                                          TRAP    C$ESUB
 8102
 8103                                  ;-------------------------------------------------------------------------
 8104                                  ; DO MASTER CLEAR, LOAD 2 TSOM'S INTO TX SILO, CLOCK TRANSMITTER UNTIL ACTIVE,
 8105                                  ;    CHECK FOR RTS, CS, CARR SET IN REG 13
 8106                                  ; THIS SUBTEST CHECKS ONLY INTERNAL CIRCUITRY, NEITHER THE CABLE, THE DRIVERS
 8107                                  ;    NOR THE RECEIVERS ARE TESTED.
 8108                                  ;-------------------------------------------------------------------------
 8109  034132                                  BGNSUB
 8110  034132                                                        T23.10:
 8111  034132  104402                                                          TRAP    C$BSUB
 8112  034134  004737  005226                  JSR     PC,INITRN       ;MST CLR, LOAD SOM'S, CLK TRANSMITTER
 8113  034140  000000                          000
 8114  034142  000000                          000
 8115  034144  012737  000013  002352          MOV     #13,REGNUM      ;SET REG NO. = 13
 8116  034152  004737  003360                  JSR     PC,READLU       ;READ REG 13
 8117  034156  132737  000040  002336          BITB    #RTS,REDBYT     ;SEE IF RTS = 1
 8118  034164  001010                          BNE     6$              ;BR IF RTS = 1
 8119  034166  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 8120                                  ;REPORT RTS NOT SET
 8121  034172                                  ERRDF   60,EM60,ERR7
 8122  034172  104455                                                          TRAP    C$ERDF
 8123  034174  000074                                                          .WORD   60
 8124  034176  015010                                                          .WORD   EM60
 8125  034200  020342                                                          .WORD   ERR7
 8126  034202                                  ESCAPE  SUB
 8127  034202  104410                                                          TRAP    C$ESCAPE
 8128  034204  000056                                                          .WORD   L10075-.
 8129  034206  132737  000004  002336  6$:     BITB    #CS,REDBYT      ;SEE IF CS = 1
 8130  034214  001010                          BNE     9$              ;BR IF CS = 1
 8131  034216  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 8132                                  ;REPORT CS NOT SET
 8133  034222                                  ERRDF   61,EM61,ERR7
 8134  034222  104455                                                          TRAP    C$ERDF
 8135  034224  000075                                                          .WORD   61
 8136  034226  015024                                                          .WORD   EM61
 8137  034230  020342                                                          .WORD   ERR7
 8138  034232                                  ESCAPE  SUB
 8139  034232  104410                                                          TRAP    C$ESCAPE
 8140  034234  000026                                                          .WORD   L10075-.
 8141  034236  132737  000001  002336  9$:     BITB    #CARR,REDBYT    ;SEE IF CARR = 1
 8142  034244  001006                          BNE     12$             ;BR IF CARR = 1
 8143  034246  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 8144                                  ;REPORT CARR NOT SET
```

```
                        ERRDF   62,EM62,ERR7
                                                                        TRAP    C$ERDF
                                                                        .WORD   62
                                                                        .WORD   EM62
                                                                        .WORD   ERR7
                12$:
                        ENDSUB
                                                                L10075:
                                                                        TRAP    C$ESUB

                A12:
                        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
                ENDTST
                                                                L10063:
                                                                        TRAP    C$ETST

                ;;*************************************************************************
                .SBTTL      TEST 24 - DATA TEST - BIT MODE, NO ERR DET
                ;*
                ;* A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH ERROR DETECTION
                ;* INHIBITED. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
                ;* AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
                ;* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
                ;* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
                ;* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
                ;* TEST WILL NOT BE RUN.
                ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
                ;*               375,373,367,357,337,277,177
                ;* 8-BIT CHARACTERS ARE USED.
                ;* TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
                ;* RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
                ;;*************************************************************************
                BGNTST
                                                                        T24::
                        MOV     #24.,TSTNUM     ;SET TEST NO.
                        MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
                        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
                        JSR     PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
                        0
                        MOV     MODINT,6$       ;SET MODEM SELECTION
                        JSR     PC,SETUP        ;PROGRAM THE USYRT
                        000
                        CRC2!CRC1               ;BIT MODE, NO ERR DET
                6$:     .WORD   0               ;MODEM SELECTION GOES HERE
                        000
                        JSR     PC,SETMTM       ;IF MOD LPBK, SET MAINT BIT, TEST FOR TESTMD SET
                        JSR     PC,LODATA       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
                        MOV     #RXEBL!177,RCVBUF+78. ; SET LAST DATA CHAR IN BUFFER
                        JSR     PC,TSTMLB       ;TEST FOR CARR SET IF MODEM LOOPBACK
                        MOV     #12,REGNUM      ;SET LU REG NO. = 12
                        MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
                9$:     MOV     TCOUNT,R2               ;INIT TIMER
```

```
8201  034402  004737  003360          10$:    JSR     PC,READLU       ;READ REG 12
8202  034406  132737  000020  002336          BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
8203  034414  001011                          BNE     12$             ;BR IF YES
8204  034416  005202                          INC     R2              ;INCREMENT TIMER
8205  034420  001370                          BNE     10$             ;BR IF NO TIME-OUT YET
8206  034422  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
8207                                  ;REPORT IRDY NOT SET
8208  034426                                  ERRDF   17,EM17,ERR7
8209  034426  104455                                                                  TRAP    C$ERDF
8210  034430  000021                                                                  .WORD   17
8211  034432  013655                                                                  .WORD   EM17
8212  034434  020342                                                                  .WORD   ERR7
8213  034436  000411                          BR      24$             ;ESCAPE TO END OF TEST
8214  034440  012337  034450          12$:    MOV     (R3)+,16$
8215  034444  004737  007302                  JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
8216  034450  000000          16$:    0
8217  034452  000000                  0
8218  034454  020327  003066                  CMP     R3,#RCVBUF+80.  ;SEE IF ALL CHARS CHECKED YET
8219  034460  103746                          BLO     9$              ;BR IF NOT YET
8220  034462                          24$:
8221  034462                          ENDTST
8222  034462                                                                  L10076:
8223  034462  104401                                                          TRAP    C$ETST
8224
8225
8226
8227
8228
8229                                  ;;*****************************************************************
8230                                  .SBTTL      TEST 25 - DATA TEST - CHAR MODE, NO ERR DET
8231                                  ;*
8232                                  ;* A MESSAGE IS INITIATED IN CHAR MODE, WITH ERROR DETECTION
8233                                  ;* INHIBITED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
8234                                  ;* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
8235                                  ;* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
8236                                  ;* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
8237                                  ;* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
8238                                  ;* TEST WILL NOT BE RUN.
8239                                  ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
8240                                  ;*               375,373,367,357,337,277,177
8241                                  ;* 8-BIT CHARACTERS ARE USED.
8242                                  ;* TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
8243                                  ;* RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
8244                                  ;;*****************************************************************
8245  034464                          BGNTST
8246  034464                                                                  T25::
8247  034464  012737  000031  002422          MOV     #25.,TSTNUM     ;SET TEST NO.
8248  034472  012737  034654  002334          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
8249  034500  004737  003262                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
8250  034504  004737  011004                  JSR     PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
8251  034510  000000                          0
8252  034512  013737  002410  034530          MOV     MODINT,6$       ;SET MODEM SELECTION
8253  034520  004737  010312                  JSR     PC,SETUP        ;PROGRAM THE USYRT
8254  034524  000226                          SYNCH
8255  034526  000311                          CRC2!CRC1!STRIP!DDCMP   ;CHAR MODE, NO ERR DET
8256  034530  000000          6$:     .WORD   0                       ;MODEM SELECTION GOES HERE
```

```
8257  034532  000000                          000
8258  034534  004737  012070                  JSR     PC,SETMTM       ;IF MOD LPBK, SET MAINT BIT, TEST FOR TESTMD SET
8259  034540  004737  010172                  JSR     PC,LODATA       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
8260  034544  012737  000177  003064          MOV     #177,RCVBUF+78. ;SET LAST DATA CHAR IN BUFFER
8261  034552  004737  012412                  JSR     PC,TSTMLB       ;TEST FOR CARR SET IF MODEM LOOPBACK
8262  034556  012737  000012  002352          MOV     #12,REGNUM      ;SET LU REG NO. = 12
8263  034564  012703  002746                  MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
8264  034570  013702  002252          9$:     MOV     TCOUNT,R2               ;INIT TIMER
8265  034574  004737  003360          10$:    JSR     PC,READLU       ;READ REG 12
8266  034600  132737  000020  002336          BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
8267  034606  001011                          BNE     12$             ;BR IF YES
8268  034610  005202                          INC     R2              ;INCREMENT TIMER
8269  034612  001370                          BNE     10$             ;BR IF NO TIME-OUT YET
8270  034614  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
8271                                  ;REPORT IRDY NOT SET
8272  034620                                  ERRDF   17,EM17,ERR7
8273  034620  104455                                                                  TRAP    C$ERDF
8274  034622  000021                                                                  .WORD   17
8275  034624  013655                                                                  .WORD   EM17
8276  034626  020342                                                                  .WORD   ERR7
8277  034630  000411                          BR      24$             ;ESCAPE TO END OF TEST
8278  034632  012337  034642          12$:    MOV     (R3)+,16$
8279  034636  004737  007302                  JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
8280  034642  000000                  16$:    0
8281  034644  000000                          0
8282  034646  020327  003066                  CMP     R3,#RCVBUF+80.  ;SEE IF ALL CHARS CHECKED YET
8283  034652  103746                          BLO     9$              ;BR IF NOT YET
8284  034654                          24$:
8285  034654                          ENDTST
8286  034654                                                                  L10077:
8287  034654  104401                                                                  TRAP    C$ETST
8288
8289
8290
8291
8292
8293                                  ;;*************************************************************************
8294                                  .SBTTL      TEST 26 - DATA TEST - BIT MODE, CRC-CCITT-1
8295                                  ;*
8296                                  ;* A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH CRC-CCITT-1 ERROR
8297                                  ;* DETECTION. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
8298                                  ;* AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
8299                                  ;* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
8300                                  ;* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
8301                                  ;* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
8302                                  ;* TEST WILL NOT BE RUN.
8303                                  ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
8304                                  ;*               375,373,367,357,337,277,177
8305                                  ;* 8-BIT CHARACTERS ARE USED.
8306                                  ;* TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
8307                                  ;* RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
8308                                  ;;*************************************************************************
8309  034656                          BGNTST
8310  034656                                                                  T26::
8311  034656  012737  000032  002422          MOV     #26.,TSTNUM     ;SET TEST NO.
8312  034664  012737  035046  002334          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
```

```
8313  034672  004737  003262                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
8314  034676  004737  011004                  JSR     PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
8315  034702  000000                          0
8316  034704  013737  002410  034722          MOV     MODINT,6$       ;SET MODEM SELECTION
8317  034712  004737  010312                  JSR     PC,SETUP        ;PROGRAM THE USYRT
8318  034716  000000                          000
8319  034720  000000                          000                     ;BIT MODE CRC-CCITT-1
8320  034722  000000                  6$:     .WORD   0               ;MODEM SELECTION GOES HERE
8321  034724  000000                          000
8322  034726  004737  012070                  JSR     PC,SETMTM       ;IF MOD LPBK, SET MAINT BIT, TEST FOR TESTMD SET
8323  034732  004737  010172                  JSR     PC,LODATA       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
8324  034736  012737  101177  003064          MOV     #CRCCHK!RXEBL!177,RCVBUF+78. ;SET LAST DATA CHAR IN BUFFER
8325  034744  004737  012412                  JSR     PC,TSTMLB       ;TEST FOR CARR SET IF MODEM LOOPBACK
8326  034750  012737  000012  002352          MOV     #12,REGNUM      ;SET LU REG NO. = 12
8327  034756  012703  002746                  MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
8328  034762  013702  002252          9$:     MOV     TCOUNT,R2               ;INIT TIMER
8329  034766  004737  003360          10$:    JSR     PC,READLU       ;READ REG 12
8330  034772  132737  000020  002336          BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
8331  035000  001011                          BNE     12$             ;BR IF YES
8332  035002  005202                          INC     R2              ;INCREMENT TIMER
8333  035004  001370                          BNE     10$             ;BR IF NO TIME-OUT YET
8334  035006  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
8335                                  ;REPORT IRDY NOT SET
8336  035012                                  ERRDF   17,EM17,ERR7
8337  035012  104455                                                                  TRAP    C$ERDF
8338  035014  000021                                                                  .WORD   17
8339  035016  013655                                                                  .WORD   EM17
8340  035020  020342                                                                  .WORD   ERR7
8341  035022  000411                          BR      24$             ;ESCAPE TO END OF TEST
8342  035024  012337  035034          12$:    MOV     (R3)+,16$
8343  035030  004737  007302                  JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
8344  035034  000000                  16$:    0
8345  035036  000000                          0
8346  035040  020327  003066                  CMP     R3,#RCVBUF+80.  ;SEE IF ALL CHARS CHECKED YET
8347  035044  103746                          BLO     9$              ;BR IF NOT YET
8348  035046                          24$:
8349  035046                          ENDTST
8350  035046                                                                  L10100:
8351  035046  104401                                                                  TRAP    C$ETST
8352
8353
8354
8355
8356
8357                                  ;;*************************************************************************
8358                                  .SBTTL      TEST 27 - DATA TEST - BIT MODE, CRC-CCITT-0
8359                                  ;*
8360                                  ;* A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH CRC-CCITT-0 ERROR
8361                                  ;* DETECTION. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
8362                                  ;* AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
8363                                  ;* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
8364                                  ;* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
8365                                  ;* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
8366                                  ;* TEST WILL NOT BE RUN.
8367                                  ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
8368                                  ;*               375,373,367,357,337,277,177
```

```
                                ;* 8-BIT CHARACTERS ARE USED.
                                ;* TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
                                ;* RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
                                ;;*************************************************************************
035050                          BGNTST
035050                                                                          T27::
035050  012737  000033  002422          MOV     #27.,TSTNUM     ;SET TEST NO.
035056  012737  035240  002334          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
035064  004737  003262                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
035070  004737  011004                  JSR     PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
035074  000000                          0
035076  013737  002410  035114          MOV     MODINT,6$       ;SET MODEM SELECTION
035104  004737  010312                  JSR     PC,SETUP        ;PROGRAM THE USYRT
035110  000000                          000
035112  000100                          CRC1                    ;BIT MODE, CRC-CCITT-0
035114  000000                  6$:     .WORD   0               ;MODEM SELECTION GOES HERE
035116  000000                          000
035120  004737  012070                  JSR     PC,SETMTM       ;IF MOD LPBK, SET MAINT BIT, TEST FOR TESTMD SET
035124  004737  010172                  JSR     PC,LODATA       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
035130  012737  101177  003064          MOV     #CRCCHK!RXEBL!177,RCVBUF+78. ;SET LAST DATA CHAR IN BUFFER
035136  004737  012412                  JSR     PC,TSTMLB       ;TEST FOR CARR SET IF MODEM LOOPBACK
035142  012737  000012  002352          MOV     #12,REGNUM      ;SET LU REG NO. = 12
035150  012703  002746                  MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
035154  013702  002252          9$:     MOV     TCOUNT,R2               ;INIT TIMER
035160  004737  003360          10$:    JSR     PC,READLU       ;READ REG 12
035164  132737  000020  002336          BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
035172  001011                          BNE     12$             ;BR IF YES
035174  005202                          INC     R2              ;INCREMENT TIMER
035176  001370                          BNE     10$             ;BR IF NO TIME-OUT YET
035200  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                ;REPORT IRDY NOT SET
035204                                  ERRDF   17,EM17,ERR7
035204  104455                                                                  TRAP    C$ERDF
035206  000021                                                                  .WORD   17
035210  013655                                                                  .WORD   EM17
035212  020342                                                                  .WORD   ERR7
035214  000411                          BR      24$             ;ESCAPE TO END OF TEST
035216  012337  035226          12$:    MOV     (R3)+,16$
035222  004737  007302                  JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
035226  000000                  16$:    0
035230  000000                          0
035232  020327  003066                  CMP     R3,#RCVBUF+80.  ;SEE IF ALL CHARS CHECKED YET
035236  103746                          BLO     9$              ;BR IF NOT YET
035240                          24$:
035240                          ENDTST
035240                                                                  L10101:
035240  104401                                                                  TRAP    C$ETST


                                ;;*************************************************************************
                                .SBTTL      TEST 28 - DATA TEST - CHAR MODE, CRC-16
                                ;*
                                ;* A MESSAGE IS INITIATED IN CHAR MODE, WITH CRC-16 ERROR
```

```
                        ;* DETECTION. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
                        ;* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
                        ;* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
                        ;* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
                        ;* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
                        ;* TEST WILL NOT BE RUN.
                        ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
                        ;*               375,373,367,357,337,277,177
                        ;* 8-BIT CHARACTERS ARE USED.
                        ;* TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
                        ;* RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
                        ;;*****************************************************************************
035242                  BGNTST
035242                                                                          T28::
035242  012737  000034  002422          MOV     #28.,TSTNUM     ;SET TEST NO.
035250  012737  035432  002334          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
035256  004737  003262                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
035262  004737  011004                  JSR     PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
035266  000000                          0
035270  013737  002410  035306          MOV     MODINT,6$       ;SET MODEM SELECTION
035276  004737  010312                  JSR     PC,SETUP        ;PROGRAM THE USYRT
035302  000226                          SYNCH
035304  000011                          STRIP!DDCMP
035306  000000                  6$:     .WORD   0               ;MODEM SELECTION GOES HERE
035310  000000                          000
035312  004737  012070                  JSR     PC,SETMTM       ;IF MOD LPBK, SET MAINT BIT, TEST FOR TESTMD SET
035316  004737  010172                  JSR     PC,LODATA       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
035322  012737  100577  003064          MOV     #CRCCHK!RXBCC!177,RCVBUF+78. ;SET LAST DATA CHAR IN BUFFER
035330  004737  012412                  JSR     PC,TSTMLB       ;TEST FOR CARR SET IF MODEM LOOPBACK
035334  012737  000012  002352          MOV     #12,REGNUM      ;SET LU REG NO. = 12
035342  012703  002746                  MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
035346  013702  002252          9$:     MOV     TCOUNT,R2               ;INIT TIMER
035352  004737  003360          10$:    JSR     PC,READLU       ;READ REG 12
035356  132737  000020  002336          BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
035364  001011                          BNE     12$             ;BR IF YES
035366  005202                          INC     R2              ;INCREMENT TIMER
035370  001370                          BNE     10$             ;BR IF NO TIME-OUT YET
035372  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                        ;REPORT IRDY NOT SET
035376                                  ERRDF   17,EM17,ERR7
035376  104455                                                                  TRAP    C$ERDF
035400  000021                                                                  .WORD   17
035402  013655                                                                  .WORD   EM17
035404  020342                                                                  .WORD   ERR7
035406  000411                          BR      24$             ;ESCAPE TO END OF TEST
035410  012337  035420          12$:    MOV     (R3)+,16$
035414  004737  007302                  JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
035420  000000                  16$:    0
035422  000000                          0
035424  020327  003066                  CMP     R3,#RCVBUF+80.  ;SEE IF ALL CHARS CHECKED YET
035430  103746                          BLO     9$              ;BR IF NOT YET
035432                          24$:
035432                          ENDTST
035432                                                                  L10102:
035432  104401                                                                  TRAP    C$ETST
```

```




                                               ;;*****************************************************************
                                               .SBTTL      TEST 29 - DATA TEST - CHAR MODE, ODD VRC
                                               ;*
                                               ;* A MESSAGE IS INITIATED IN CHAR MODE, WITH ODD VRC ERROR DETECTION
                                               ;* SELECTED. THE MESSAGE CONSISTS OF 5 SYNCH^, PAT A REPEATED 2 TIMES,
                                               ;* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
                                               ;* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
                                               ;* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
                                               ;* TEST WILL BE RUN ON THAT '"ITERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
                                               ;* TEST WILL NOT BE RUN.
                                               ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
                                               ;*               375,373,367,357,337,277,177
                                               ;* 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).
                                               ;* TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
                                               ;* RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
                                               ;;*****************************************************************
035434                                         BGNTST
035434                                                                                     T29::
035434  012737  000035  002422                         MOV     #29.,TSTNUM     ;SET TEST NO.
035442  012737  035620  002334                         MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
035450  004737  003262                                 JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
035454  004737  011004                                 JSR     PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
035460  000000                                         0
035462  013737  002410  035500                         MOV     MODINT,6$       ;SET MODEM SELECTION
035470  004737  010312                                 JSR     PC,SETUP        ;PROGRAM THE USYRT
035474  000026                                         026
035476  000111                                         CRC1!STRIP!DDCMP
035500  000000                                 6$:     .WORD   0               ;MODEM SELECTION GOES HERE
035502  000347                                         TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0
035504  004737  012070                                 JSR     PC,SETMTM       ;IF MOD LPBK, SET MAINT BIT, TEST FOR TESTMD SET
035510  004737  010172                                 JSR     PC,LODATA       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
035514  004737  012412                                 JSR     PC,TSTMLB       ;TEST FOR CARR SET IF MODEM LOOPBACK
035520  012737  000012  002352                         MOV     #12,REGNUM      ;SET LU REG NO. = 12
035526  012703  002746                                 MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
035532  013702  002252                         9$:     MOV     TCOUNT,R2               ;INIT TIMER
035536  004737  003360                         10$:    JSR     PC,READLU       ;READ REG 12
035542  132737  000020  002336                         BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
035550  001011                                         BNE     12$             ;BR IF YES
035552  005202                                         INC     R2              ;INCREMENT TIMER
035554  001370                                         BNE     10$             ;BR IF NO TIME-OUT YET
035556  004737  004214                                 JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                               ;REPORT IRDY NOT SET
035562                                                 ERRDF   17,EM17,ERR7
035562  104455                                                                                     TRAP    C$ERDF
035564  000021                                                                                     .WORD   17
035566  013655                                                                                     .WORD   EM17
035570  020342                                                                                     .WORD   ERR7
035572  000412                                         BR      24$             ;ESCAPE TO END OF TEST
035574  112337  035606                         12$:    MOVB    (R3)+,16$       ;GET AN EXPECTED DATA BYTE
035600  005203                                         INC     R3              ;INCREMENT POINTER
035602  004737  007302                                 JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
035606  100000                                 16$:    BCCCHK
```

```
8537  035610  000000                          0
8538  035612  020327  003066                  CMP     R3,#RCVBUF+80.  ;SEE IF ALL CHARS CHECKED YET
8539  035616  103745                          BLO     9$              ;BR IF NOT YET
8540  035620                          24$:
8541  035620                          ENDTST
8542  035620                                                                  L10103:
8543  035620  104401                                                                 TRAP    C$ETST
8544
8545
8546
8547
8548
8549                                  ;;*************************************************************************
8550                                  .SBTTL      TEST 30 - DATA TEST - CHAR MODE, EVEN VRC
8551                                  ;*
8552                                  ;* A MESSAGE IS INITIATED IN CHAR MODE, WITH EVEN VRC ERROR DETECTION
8553                                  ;* SELECTED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
8554                                  ;* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
8555                                  ;* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
8556                                  ;* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
8557                                  ;* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
8558                                  ;* TEST WILL NOT BE RUN.
8559                                  ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
8560                                  ;*               375,373,367,357,337,277,177
8561                                  ;* 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).
8562                                  ;* TO RUN REMOTE MODEM LOOP BACK TESTS, THE PHONE LINK MAY HAVE TO BE
8563                                  ;* RE-ESTABLISHED AT THE BEGINNING OF THIS TEST.
8564                                  ;;*************************************************************************
8565  035622                          BGNTST
8566  035622                                                                  T30::
8567  035622  012737  000036  002422          MOV     #30.,TSTNUM     ;SET TEST NO.
8568  035630  012737  036006  002334          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
8569  035636  004737  003262                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
8570  035642  004737  011004                  JSR     PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
8571  035646  000000                          0
8572  035650  013737  002410  035666          MOV     MODINT,6$       ;SET MODEM SELECTION
8573  035656  004737  010312                  JSR     PC,SETUP        ;PROGRAM THE USYRT
8574  035662  000026                          026
8575  035664  000211                          CRC2!STRIP!DDCMP
8576  035666  000000                  6$:     .WORD   0               ;MODEM SELECTION GOES HERE
8577  035670  000347                          TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0
8578  035672  004737  012070                  JSR     PC,SETMTM       ;IF MOD LPBK, SET MAINT BIT, TEST FOR TESTMD SET
8579  035676  004737  010172                  JSR     PC,LODATA       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
8580  035702  004737  012412                  JSR     PC,TSTMLB       ;TEST FOR CARR SET IF MODEM LOOPBACK
8581  035706  012737  000012  002352          MOV     #12,REGNUM      ;SET LU REG NO. = 12
8582  035714  012703  002746                  MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
8583  035720  013702  002252          9$:     MOV     TCOUNT,R2               ;INIT TIMER
8584  035724  004737  003360          10$:    JSR     PC,READLU       ;READ REG 12
8585  035730  132737  000020  002336          BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
8586  035736  001011                          BNE     12$             ;BR IF YES
8587  035740  005202                          INC     R2              ;INCREMENT TIMER
8588  035742  001370                          BNE     10$             ;BR IF NO TIME-OUT YET
8589  035744  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
8590                                  ;REPORT IRDY NOT SET
8591  035750                                  ERRDF   17,EM17,ERR7
8592  035750  104455                                                                  TRAP    C$ERDF
```

```
8593  035752  000021                                                          .WORD   17
8594  035754  013655                                                          .WORD   EM17
8595  035756  020342                                                          .WORD   ERR7
8596  035760  000412                      BR      24$             ;ESCAPE TO END OF TEST
8597  035762  112337  035774      12$:    MOVB    (R3)+,16$       ;GET AN EXPECTED DATA CHAR
8598  035766  005203                      INC     R3              ;INCREMENT POINTER
8599  035770  004737  007302              JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
8600  035774  100000      16$:    BCCCHK
8601  035776  000000                      0
8602  036000  020327  003066              CMP     R3,#RCVBUF+80.  ;SEE IF ALL CHARS CHECKED YET
8603  036004  103745                      BLO     9$              ;BR IF NOT YET
8604  036006                      24$:
8605  036006                      ENDTST
8606  036006                                                              L10104:
8607  036006  104401                                                             TRAP    C$ETST
8608
8609
8610
8611
8612
8613                              ;;*****************************************************************************
8614                              .SBTTL      TEST 31 - CONTIGUOUS ONES IN SEC. STA. ADRS. MODE, BIT MODE
8615                              ;*
8616                              ;* IN THIS TEST, A MESSAGE CONSISTING OF 5 ONES CHARS (377 OCT)
8617                              ;* IS SENT IN SECONDARY STATION ADDRESS MODE, WITH THE STATION ADRS
8618                              ;* FOR THIS LINE = 377. THE PROGRAM CHECKS FOR CORRECT RECEPTION OF
8619                              ;* THE FIRST CHARACTER (STATION ADDRESS) AND THE REMAINING 4
8620                              ;* ONES CHARACTERS (DATA). THIS TEST EXERCISES THE SECONDARY STATION
8621                              ;* ADDRESS LOGIC, AND CHECKS THAT THE SEC. STA. ADRS. CAN BE BIT-STUFFED
8622                              ;* AND TRANSMITTED AND RECEIVED CORRECTLY.
8623                              ;;*****************************************************************************
8624  036010                      BGNTST
8625  036010                                                                  T31::
8626  036010  012737  036114  002334      MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
8627  036016  004737  005226              JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
8628  036022  000377                      377
8629  036024  000320                      CRC2!CRC1!SECA          ;BIT MODE, NO ERROR DET, SEC ADR MODE
8630  036026  004737  010734              JSR     PC,LODSIL       ;LOAD 5 377-CHARS INTO TX SILO
8631  036032  000377                      377
8632  036034  000005                      5
8633  036036  004737  010734              JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
8634  036042  001000                      TXEOM
8635  036044  000002                      2
8636  036046  004737  004742              JSR     PC,STPLU        ;CLOCK MORE THAN ENTIRE MSG
8637  036052  000240                      160.
8638  036054  004737  007302              JSR     PC,CKDATA       ;RCV SEC ADRS = 377
8639  036060  000377                      377
8640  036062  000000                      0
8641  036064  012701  000003              MOV     #3,R1           ;RCV 3 MORE 377 CHARS
8642  036070  004737  007302      6$:     JSR     PC,CKDATA
8643  036074  000377                      377
8644  036076  000000                      0
8645  036100  005301                      DEC     R1
8646  036102  001372                      BNE     6$
8647  036104  004737  007302              JSR     PC,CKDATA       ;RCV LAST 377 CHAR, CHK EBLK = 1
8648  036110  001377                      1377
```

```
036112  000000                                     0
036114  004737  003262                     24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
036120                                     ENDTST
036120                                                                     L10105:
036120  104401                                                                     TRAP    C$ETST





                                           ;;*********************************************************************
                                           .SBTTL      TEST 32 - DDCMP MESSAGE TEST - CHAR MODE
                                           ;*
                                           ;* IN THIS TEST, THREE USYRT MESSAGES ARE SENT TO SIMULATE A DDCMP HEADER,
                                           ;* DDCMP DATA MESSAGE, AND THE START OF A NEW DDCMP HEADER.
                                           ;* FIRST, THE DATA IN PATTERN A IS TRANSMITTED AND RECEIVED
                                           ;* AND THEN CRC (CRC-16) IS SENT, FOLLOWED BY THE DATA IN PATTERN A
                                           ;* AGAIN AND THE CRC ON THAT DATA, AND FINALLY THE DATA IN 'MSG1'' IS
                                           ;* SENT WITH ITS CORRESPONDING CRC.
                                           ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
                                           ;*               375,373,367,357,337,277,177
                                           ;*   MSG1 = SYNCH,SYNCH,SYNCH,SYNCH,000,125,252,377,000,SYNCH,SYNCH
                                           ;;*********************************************************************
036122                                     BGNTST
036122                                                                     T32::
036122  012737  036700  002334                     MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
                                           ;-------------------------------------------------------------------
                                           ; TRANSMIT AND RCV ENTIRE MSG
                                           ;-------------------------------------------------------------------
036130  004737  005226                             JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
036134  000226                                     SYNCH
036136  000011                                     STRIP!DDCMP
036140  004737  010562                             JSR     PC,LDBYTS       ;LOAD 20 WORDS OF PAT A INTO TX SILO
036144  002543                                     PATA
036146  000024                                     20.
036150  004737  010734                             JSR     PC,LODSIL       ;LOAD AN EOM INTO TX SILO
036154  001000                                     TXEOM
036156  000001                                     1
036160  004737  010562                             JSR     PC,LDBYTS       ;LOAD 20 WORDS OF PAT A INTO TX SILO
036164  002543                                     PATA
036166  000024                                     20.
036170  004737  010734                             JSR     PC,LODSIL       ;LOAD 1 EOM INTO TX SILO
036174  001000                                     TXEOM
036176  000001                                     1
036200  004737  010734                             JSR     PC,LODSIL       ;LOAD 3 SOM'S INTO TX SILO
036204  000400                                     TXSOM
036206  000003                                     3
036210  004737  010504                             JSR     PC,LODMSG       ;LOAD MSG1 INTO TX SILO
036214  002654                                     MSG1
036216  000013                                     11.
036220  004737  004742                             JSR     PC,STPLU        ;CLOCK HDR MSG AND CRC CHARS
036224  000300                                     192.
036226  012737  000013  002352                     MOV     #13,REGNUM      ;SET REG. NO. = 13
036234  004737  003360                             JSR     PC,READLU       ;READ REG 13
036240  032737  000040  002336                     BIT     #RTS,REDBYT     ;SEE IF RTS SET
036246  001010                                     BNE     2$              ;BR IF RTS SET
```

```
8705  036250  004737  004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
8706                          ;REPORT RTS NOT SET
8707  036254                  ERRDF   60,EM60,ERR7
8708  036254  104455                                                  TRAP    C$ERDF
8709  036256  000074                                                  .WORD   60
8710  036260  015010                                                  .WORD   EM60
8711  036262  020342                                                  .WORD   ERR7
8712  036264  000137  036700          JMP     24$             ;EXIT TEST
8713  036270  004737  004742  2$:     JSR     PC,STPLU        ;CLK DATA MSG AND FIRST CRC CHAR
8714  036274  000250                  168.
8715  036276  012703  000040          MOV     #32.,R3         ;SET COUNTER FOR CHECKING RTS
8716  036302  004737  004742  4$:     JSR     PC,STPLU        ;CLOCK LINE UNIT FOR 1 CYCLE
8717  036306  000001                  1
8718  036310  004737  003360          JSR     PC,READLU       ;READ REG 13
8719  036314  032737  000040  002336  BIT     #RTS,REDBYT     ;CHK FOR RTS SET
8720  036322  001007                  BNE     5$              ;BR IF RTS SET
8721  036324  004737  004214          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
8722                          ;REPORT RTS NOT SET
8723  036330                  ERRDF   60,EM60,ERR7
8724  036330  104455                                                  TRAP    C$ERDF
8725  036332  000074                                                  .WORD   60
8726  036334  015010                                                  .WORD   EM60
8727  036336  020342                                                  .WORD   ERR7
8728  036340  000557          BR      24$
8729  036342  005303  5$:     DEC     R3              ;DECR COUNTER
8730  036344  001356          BNE     4$              ;BR IF NOT DONE YET
8731                          ;-------------------------------------------------------------
8732                          ; READ AND CHK HEADER AND CRC
8733                          ;-------------------------------------------------------------
8734  036346  012701  002543          MOV     #PATA,R1        ;INIT PATTERN A POINTER
8735  036352  112137  036362  7$:     MOVB    (R1)+,8$        ;GET AN EXPECTED CHAR
8736  036356  004737  007302          JSR     PC,CKDATA       ;READ AND CHK A CHAR
8737  036362  000000          8$:     .WORD   0
8738  036364  000000                  0
8739  036366  020127  002565          CMP     R1,#PATB-2      ;SEE IF CHKING NEXT-TO-LAST CHAR YET
8740  036372  103767                  BLO     7$              ;BR IF NOT YET
8741  036374  004737  007302          JSR     PC,CKDATA       ;READ AND CHK CHAR, BCC=0
8742  036400  100277                  CRCCHK!277
8743  036402  000000                  0
8744  036404  004737  007302          JSR     PC,CKDATA       ;READ AND CHK LAST CHAR, BCC=1
8745  036410  100577                  CRCCHK!RXBCC!177
8746  036412  000000                  0
8747  036414  004737  007302          JSR     PC,CKDATA       ;READ AND CHK HI CRC BYTE
8748  036420  000156                  156
8749  036422  000000                  0
8750  036424  004737  007302          JSR     PC,CKDATA       ;READ AND CHK LO CRC BYTE
8751  036430  000236                  236
8752  036432  000000                  0
8753                          ;-------------------------------------------------------------
8754                          ; READ AND CHK DATA MSG AND CRC
8755                          ;-------------------------------------------------------------
8756  036434  012701  002543          MOV     #PATA,R1        ;INIT PATTERN A POINTER
8757  036440  112137  036450  9$:     MOVB    (R1)+,12$       ;GET AN EXPECTED CHAR
8758  036444  004737  007302          JSR     PC,CKDATA       ;READ AND CHK A CHAR
8759  036450  000000          12$:    .WORD   0
8760  036452  000000                  0
```

```
8761  036454  020127  002565          CMP     R1,#PATB-2      ;SEE IF CHKING NEXT-TO-LAST CHAR YET
8762  036460  103767                  BLO     9$              ;BR IF NOT YET
8763  036462  004737  007302          JSR     PC,CKDATA       ;READ AND CHK CHAR, BCC=0
8764  036466  100277                  CRCCHK!277
8765  036470  000000                  0
8766  036472  004737  007302          JSR     PC,CKDATA       ;READ AND CHK LAST CHAR, BCC=1
8767  036476  100577                  CRCCHK!RXBCC!177
8768  036500  000000                  0
8769  036502  004737  007302          JSR     PC,CKDATA       ;READ AND CHK HI CRC BYTE
8770  036506  000156                  156
8771  036510  000000                  0
8772  036512  004737  007302          JSR     PC,CKDATA       ;READ AND CHK LO CRC BYTE
8773  036516  000236                  236
8774  036520  000000                  0
8775                          ;-------------------------------------------------------------------------
8776                          ; CLOCK 3RD MESSAGE ('MSG1'' DATA)
8777                          ;-------------------------------------------------------------------------
8778  036522  012737  000012  002352          MOV     #12,REGNUM      ;SET REG NO. = 12
8779  036530  112737  000200  002340          MOVB    #IC.WRIBYT      ;SET IC TO CLEAR RECEIVER FOR NEW MSG
8780  036536  004737  003436                  JSR     PC,WRITLU
8781  036542  012737  000013  002352          MOV     #13,REGNUM      ;RESTORE REG NO. TO 13
8782  036550  004737  004742                  JSR     PC,STPLU        ;CLOCK THE REST OF MSG
8783  036554  000150                          104.
8784  036556  004737  003360                  JSR     PC,READLU       ;READ REG 13
8785  036562  032737  000040  002336          BIT     #RTS,REDBYT     ;SEE IF RTS IS CLEARED
8786  036570  001407                          BEQ     14$             ;BR IF RTS CLEARED
8787  036572  004737  004214                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
8788                          ;REPORT RTS NOT CLEARED
8789  036576                          ERRDF   65,EM65,ERR7
8790  036576  104455                                                  TRAP    C$ERDF
8791  036600  000101                                                  .WORD   65
8792  036602  015106                                                  .WORD   EM65
8793  036604  020342                                                  .WORD   ERR7
8794  036606  000434                  BR      24$
8795                          ;-------------------------------------------------------------------------
8796                          ;READ AND CHECK 3RD MESSAGE AND CRC
8797                          ;-------------------------------------------------------------------------
8798  036610  004737  007302  14$:    JSR     PC,CKDATA       ;READ AND CHECK 000 DATA CHAR
8799  036614  000000                  000
8800  036616  000000                  0
8801  036620  004737  007302          JSR     PC,CKDATA       ;READ AND CHECK 125 DATA CHAR
8802  036624  000125                  125
8803  036626  000000                  0
8804  036630  004737  007302          JSR     PC,CKDATA       ;READ AND CHECK 252 DATA CHAR
8805  036634  000252                  252
8806  036636  000000                  0
8807  036640  004737  007302          JSR     PC,CKDATA       ;READ AND CHECK 377 DATA CHAR, AND BCC=0
8808  036644  100377                  CRCCHK!377
8809  036646  000000                  0
8810  036650  004737  007302          JSR     PC,CKDATA       ;READ AND CHECK 000 DATA CHAR, AND BCC=1
8811  036654  100400                  CRCCHK!RXBCC!000
8812  036656  000000                  0
8813  036660  004737  007302          JSR     PC,CKDATA       ;READ AND CHK HI CRC BYTE
8814  036664  000160                  160
8815  036666  000000                  0
8816  036670  004737  007302          JSR     PC,CKDATA       ;READ AND CHK LO CRC BYTE
```

```
CZDMSF.P11    30-SEP-81 15:40                TEST 32 - DDCMP MESSAGE TEST - CHAR MODE

  8817  036674  000034                              034
  8818  036676  000000                              0
  8819  036700  004737  003262          24$:        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
  8820  036704                         ENDTST
  8821  036704                                                                          L10106:
  8822  036704  104401                                                                          TRAP    C$ETST
  8823
  8824
  8825
  8826
  8827
```

```
8828                            .SBTTL  HARDWARE PARAMETER CODING SECTION
8829
8830
8831
8832                            ;////////////////////////////////////////////////////////////////////////
8833                            ;/ THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
8834                            ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
8835                            ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
8836                            ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
8837                            ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
8838                            ;/ WITH THE OPERATOR.
8839                            ;////////////////////////////////////////////////////////////////////////
8840
8841  036706                            BGNHRD
8842  036706  000035                                                          .WORD L10107-L$HARD/2
8843  036710                                                          L$HARD::
8844
8845  036710                            GPRMA   ADDRES,0,0,160000,177776,YES
8846  036710  000031                                                          .WORD   T$CODE
8847  036712  037002                                                          .WORD   ADDRES
8848  036714  160000                                                          .WORD   T$LOLIM
8849  036716  177776                                                          .WORD   T$HILIM
8850  036720                            GPRMD   SWPAC1,2,0,377,0,056,YES
8851  036720  001032                                                          .WORD   T$CODE
8852  036722  037030                                                          .WORD   SWPAC1
8853  036724  000377                                                          .WORD   377
8854  036726  000000                                                          .WORD   T$LOLIM
8855  036730  000056                                                          .WORD   T$HILIM
8856  036732                            GPRMD   SWPAC2,4,0,377,0,377,YES
8857  036732  002032                                                          .WORD   T$CODE
8858  036734  037104                                                          .WORD   SWPAC2
8859  036736  000377                                                          .WORD   377
8860  036740  000000                                                          .WORD   T$LOLIM
8861  036742  000377                                                          .WORD   T$HILIM
8862  036744                            GPRMD   SWPAC3,6,0,377,0,377,YES
8863  036744  003032                                                          .WORD   T$CODE
8864  036746  037141                                                          .WORD   SWPAC3
8865  036750  000377                                                          .WORD   377
8866  036752  000000                                                          .WORD   T$LOLIM
8867  036754  000377                                                          .WORD   T$HILIM
8868  036756                            GPRMD   LOOPBK,10,0,7,0,7,YES
8869  036756  004032                                                          .WORD   T$CODE
8870  036760  037176                                                          .WORD   LOOPBK
8871  036762  000007                                                          .WORD   7
8872  036764  000000                                                          .WORD   T$LOLIM
8873  036766  000007                                                          .WORD   T$HILIM
8874  036770                            GPRMD   BAUDRT,12,0,7,0,7,YES
8875  036770  005032                                                          .WORD   T$CODE
8876  036772  037405                                                          .WORD   BAUDRT
8877  036774  000007                                                          .WORD   7
8878  036776  000000                                                          .WORD   T$LOLIM
8879  037000  000007                                                          .WORD   T$HILIM
8880
8881  037002                            ENDHRD
8882                                                                    .EVEN
8883  037002                                                    L10107:
```

```
 8884
 8885  037002  042504  044526  042503  ADDRES: .ASCIZ  /DEVICE CSR ADDRESS : /
 8886  037010  041440  051123  040440
 8887  037016  042104  042522  051523
 8888  037024  035040  000040
 8889  037030  034115  030062  020063  SWPAC1: .ASCIZ  /M8203 REG 11 (E134 SW10,9 , E121 SW9,10) : /
 8890  037036  042522  020107  030461
 8891  037044  024040  030505  032063
 8892  037052  051440  030527  026060
 8893  037060  020071  020054  030505
 8894  037066  030462  051440  034527
 8895  037074  030454  024460  035040
 8896  037102  000040
 8897  037104  034115  030062  020063  SWPAC2: .ASCIZ  /M8203 REG 15 (E134 SW8-1) : /
 8898  037112  042522  020107  032461
 8899  037120  024040  030505  032063
 8900  037126  051440  034127  030455
 8901  037134  020051  020072     000
 8902  037141     115  031070  031460  SWPAC3: .ASCIZ  /M8203 REG 16 (E121 SW8-1) : /
 8903  037146  051040  043505  030440
 8904  037154  020066  042450  031061
 8905  037162  020061  053523  026470
 8906  037170  024461  035040  000040
 8907  037176  042523  042514  052103  LOOPBK: .ASCII  /SELECT TURNAROUND TYPE; 0=H3254&H3255, 1=H325,/<15><12>
 8908  037204  052040  051125  040516
 8909  037212  047522  047125  020104
 8910  037220  054524  042520  020073
 8911  037226  036460  031510  032462
 8912  037234  023064  031510  032462
 8913  037242  026065  030440  044075
 8914  037250  031063  026065  005015
 8915  037256  020040  031040  044075          .ASCII  /   2=H3250, 3=H3251, 4=INTEGRAL MODEM HDX SWITCH,/<15><12>
 8916  037264  031063  030065  020054
 8917  037272  036463  031510  032462
 8918  037300  026061  032040  044475
 8919  037306  052116  043505  040522
 8920  037314  020114  047515  042504
 8921  037322  020115  042110  020130
 8922  037330  053523  032111  044103
 8923  037336  006454     012
 8924  037341     040  020040  036465          .ASCIZ  /   5=MOD LOC, 6=MOD REM, 7=NONE) : /
 8925  037346  047515  020104  047514
 8926  037354  026103  033040  046475
 8927  037362  042117  051040  046505
 8928  037370  020054  036467  047516
 8929  037376  042516  020051  020072
 8930  037404     000
 8931  037405     123  046105  041505  BAUDRT: .ASCII  /SELECT BAUD RATE; TYPE '0' FOR 2.4K; '1' FOR 4.8K;/<15><12>
 8932  037412  020124  040502  042125
 8933  037420  051040  052101  035505
 8934  037426  052040  050131  020105
 8935  037434  030047  020047  047506
 8936  037442  020122  027062  045464
 8937  037450  020073  030447  020047
 8938  037456  047506  020122  027064
 8939  037464  045470  006473     012
```

CZDMSF.P11 30-SEP-81 15:40 HARDWARE PARAMETER CODING SECTION

```
8940  037471  047  023462  043040          .ASCII  /'2' FOR 9.6K; '3' FOR 19.2K; '4' FOR 56K; '5' FOR 250K;/<15><12>
8941  037476  051117  034440  033056
8942  037504  035513  023440  023463
8943  037512  043040  051117  030440
8944  037520  027071  045462  020073
8945  037526  032047  020047  047506
8946  037534  020122  033065  035513
8947  037542  023440  023465  043040
8948  037550  051117  031040  030065
8949  037556  035513  005015
8950  037562  033047  020047  047506          .ASCIZ  /'6' FOR 500K; OR '7' FOR 1 MEG BAUD : /
8951  037570  020122  030065  045460
8952  037576  020073  051117  023440
8953  037604  023467  043040  051117
8954  037612  030440  046440  043505
8955  037620  041040  052501  020104
8956  037626  020072  000
8957          037632                  .EVEN
8958
8959
8960
8961
8962
8963
```

```
                                .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.
                                ;////////////////////////////////////////////////////////////////

037632                                  BGNSFT
037632  000016                                                          .WORD L10110-L$SOFT/2
037634                                                          L$SOFT::

037634                                  GPRML   ISMANI,0,1,YES
037634  000130                                                          .WORD   T$CODE
037636  037670                                                          .WORD   ISMANI
037640  000001                                                          .WORD   1
037642                                  GPRML   ISPRNT,2,1,YES
037642  001130                                                          .WORD   T$CODE
037644  037746                                                          .WORD   ISPRNT
037646  000001                                                          .WORD   1
037650                                  GPRML   ISWPAK,4,1,YES
037650  002130                                                          .WORD   T$CODE
037652  040015                                                          .WORD   ISWPAK
037654  000001                                                          .WORD   1
037656                                  GPRMD   TIMCNT,6,0,177777,0,177777,YES
037656  003032                                                          .WORD   T$CODE
037660  040046                                                          .WORD   TIMCNT
037662  177777                                                          .WORD   177777
037664  000000                                                          .WORD   T$LOLIM
037666  177777                                                          .WORD   T$HILIM

037670                                  ENDSFT
                                                                        .EVEN
037670                                                          L10110:

037670  047504  046440  047101  ISMANI: .ASCIZ  /DO MAN. INTERVEN. TO MOUNT TEST CONNECTOR(S) /
037676  020056  047111  042524
037704  053122  047105  020056
037712  047524  046440  052517
037720  052116  052040  051505
037726  020124  047503  047116
037734  041505  047524  024122
037742  024523  000040
037746  046101  047514  020127  ISPRNT: .ASCIZ  /ALLOW SWITCH PACK AND AX3-15 PRINTOUT /
037754  053523  052111  044103
037762  050040  041501  020113
037770  047101  020104  054101
037776  026463  032461  050040
040004  044522  052116  052517
040012  020124     000
040015     101  046114  053517  ISWPAK: .ASCIZ  /ALLOW SWITCH PACK TESTS /
040022  051440  044527  041524
```

```
CZDMSF.P11    30-SEP-81 15:40          SOFTWARE PARAMETER CODING SECTION

  9020  040030  020110  040520  045503
  9021  040036  052040  051505  051524
  9022  040044  000040
  9023  040046  051515  020107  044524  TIMCNT: .ASCIZ  /MSG TIMER VALUE (0-177777), 0 = LONGEST TIME-OUT : /
  9024  040054  042515  020122  040526
  9025  040062  052514  020105  030050
  9026  040070  030455  033467  033467
  9027  040076  024467  020054  020060
  9028  040104  020075  047514  043516
  9029  040112  051505  020124  044524
  9030  040120  042515  047455  052125
  9031  040126  035040  000040
  9032                            .EVEN
  9033
  9034
  9035
  9036
  9037
  9038
  9039
  9040
  9041                            ;****** PATCH AREA FOR DEBUG ******
  9042  040132                    PATCH:
  9043          040232                    .=.+100
  9044                            ;FIRST FREE ADDRESS MUST BE <= 40500 TO LEAVE ROOM FOR 16 PTABLES BEFORE DRS.
  9045  040232  000240                    NOP
  9046  040234  000240                    NOP
  9047  040236  000240                    NOP
  9048                            ;************************************
  9049
  9050
  9051
  9052  040240                            ENDMOD
  9053
  9054  040240                            LASTAD
  9055                                                                  .EVEN
  9056  040240  000000                                                  .WORD   0
  9057  040242  000000                                                  .WORD   0
  9058  040244                    L$LAST::
  9059
  9060          000001            .END
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- USER SYMBOLS

```
ABORT = 000004          2260#
ADDRES  037002          4701    4717    4792    4909    5011    5136    5238    5347    5457    8847    8885#
ADR   = 000020 G        2208#
ANBITS  002533          2667#
APA   = 000200          2438#
ASBC0 = 000020          2395#
ASBC1 = 000040          2394#
ASBC2 = 000100          2393#
ASSEMB= 000010          1894
AXNUM   002354          2588#   3016    3054    3067    3092    3094    3100*   3106*   3107    3109*   3111    3326    3334*
                        3377*   3565    3571*   3616*   3622*   3939    3941*   3949*   3960*   4087    4101*   4227*   4242
                        4245*   4266*   4270*   6315*   6347*   6350*   6362*   6818*   7130*   7192*   7287*   7344*   7741*
AX0.15= 002274          2508#   3099    4867    4976    5094    5203    5305    5422    5524
AX0.16= 002276          2509#   4866    4975    5093    5202    5304    5421    5523
AX1   = 000001          2291#   2363#
AX1.15= 002300          2510#   4865    4974    5092    5201    5303    5420    5522
AX1.16= 002302          2511#   4864    4973    5091    5200    5302    5419    5521
AX2   = 000002          2290#   2362#
AX2.15= 002304          2512#   4884    4993    5111    5220    5322    5439    5541
AX2.16= 002306          2513#   4883    4992    5110    5219    5321    5438    5540
AX3.15= 002310          2514#   4882    4991    5109    5218    5320    5437    5539
AX3.16= 002312          2515#   4881    4990    5108    5217    5319    5436    5538
AX315U= 000372          2458#
A11     025612          6469#   6503
A12     034264          7796    8155#
BADDAT  002360          2590#   3753*   3754*   3783*   3784*   4738    4813    5040    5368    6388*   6709*   6748*   6787*
                        7465*   7512*   7815*   7831*   7848*   7862*
BAUDRT  037405          8876    8931#
BCC   = 000001          2339#   3787    3789    3799
BCCCHK= 100000          2523#   3785    7325    7333    7382    7390    8536    8600
BDRATE  002446          2620#   4121    4185    5712*
BIT0  = 000001 G        2181#   2241    2253    2262    2291    2303    2315    2327    2339    2351    2363    2375    2387
                        2399    2411    2420    2432    2445    2457    2468    2549    3094    3135    3280    3476    3525
                        3573    5652    7331    7388
BIT00 = 000001 G        2170#   2181
BIT01 = 000002 G        2169#   2180
BIT02 = 000004 G        2168#   2179
BIT03 = 000010 G        2167#   2178
BIT04 = 000020 G        2166#   2177
BIT05 = 000040 G        2165#   2176
BIT06 = 000100 G        2164#   2175
BIT07 = 000200 G        2163#   2174
BIT08 = 000400 G        2162#   2173
BIT09 = 001000 G        2161#   2172
BIT1  = 000002 G        2180#   2240    2252    2261    2280    2290    2302    2314    2326    2338    2350    2362    2374
                        2386    2398    2410    2419    2431    2444    2456    2467    2550    3159    3452    3595
BIT10 = 002000 G        2160#   2478    2490
BIT11 = 004000 G        2159#   2477    2489
BIT12 = 010000 G        2158#
BIT13 = 020000 G        2157#
BIT14 = 040000 G        2156#
BIT15 = 100000 G        2155#   2521    2523    2524    2526    3248    3410
BIT2  = 000004 G        2179#   2239    2251    2260    2279    2289    2301    2313    2325    2337    2349    2361    2373
                        2385    2397    2409    2418    2430    2443    2455    2466
BIT3  = 000010 G        2178#   2238    2250    2259    2278    2288    2300    2312    2324    2336    2348    2360    2372
                        2384    2396    2408    2417    2429    2442    2454
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- USER SYMBOLS

```
BIT4   = 000020 G   2177# 2237  2249  2277  2287  2299  2311  2323  2335  2347  2359  2371  2383
                    2395  2407  2428  2441  2453
BIT5   = 000040 G   2176# 2248  2269  2276  2286  2298  2310  2322  2334  2346  2358  2370  2382
                    2394  2406  2427  2440  2452  2465
BIT6   = 000100 G   2175# 2236  2247  2268  2275  2285  2297  2309  2321  2333  2345  2357  2369
                    2381  2393  2405  2426  2439  2451  2464
BIT7   = 000200 G   2174# 2235  2246  2258  2267  2274  2296  2308  2320  2332  2344  2356  2368
                    2380  2392  2404  2416  2425  2438  2450  2463  3747  3748
BIT8   = 000400 G   2173# 2480  2492
BIT9   = 001000 G   2172# 2479  2491
BOE    = 000400 G   2212#
BPOLL  = 000100     2268# 8072
BSEL1    002426    2611# 2880* 2882* 2883* 2897* 2898* 2903* 2995* 2997* 3098* 3252* 3253* 3255*
                    3353* 3354* 3359* 3372* 3955* 3957* 4100* 4229* 4283* 4353* 5670* 5671* 6817*
                    7431* 7506* 7521* 7523* 7826* 7885* 7886* 7887* 7968* 8025*
BSEL4    002430    2612# 2933  2954*
CARR   = 000001     2351# 7509  8141
CBLPBK = 000001     2534# 5684  5687
CHKABT   011742     4242#
CHPCHK = 100000     2521# 6621  7000  7035  7070
CHPTYP   002406     2601# 3249  3367* 3371* 3411  5607* 6352  6358
CKDATA   007302     3738# 5940  5947  5996  6001  6004  6051  6056  6059  6121  6124  6131  6145
                    6148  6194  6197  6230  6233  6491  6494  6580  6583  6637  6640  6884  6887
                    6890  6893  6896  6914  6917  6920  6923  6926  6945  6948  6951  6954  6957
                    7011  7014  7017  7020  7023  7046  7049  7052  7055  7058  7081  7084  7087
                    7090  7093  7324  7332  7381  7389  7449  7541  7601  7661  8215  8279  8343
                    8407  8471  8535  8599  8638  8642  8647  8736  8741  8744  8747  8750  8758
                    8763  8766  8769  8772  8798  8801  8804  8807  8810  8813  8816
CKLPBK   011004     4086# 7422  7498  7573  7633  7799  8186  8250  8314  8378  8442  8506  8570
CRCCHK = 100000     2524# 3921  4036  6495  6584  6897  6927  6958  7024  7059  7094  8324  8388
                    8452  8742  8745  8764  8767  8808  8811
CRCTY0 = 000001     2445#
CRCTY1 = 000002     2444#
CRCTY2 = 000004     2443#
CRC1   = 000100     2297# 5929  5982  6037  6065  6305  6937  7067  7077  7138  7295  7313  7730
                    8191  8255  8383  8511  8629
CRC2   = 000200     2296# 5929  5982  6037  6065  6305  7200  7352  7370  7730  8191  8255  8575
                    8629
CS     = 000004     2349# 4360  8129
C$AU   = 000052     1894# 5898
C$AUTO = 000061     1894# 5826
C$BRK  = 000022     1894# 4320  5731
C$BSEG = 000004     1894#
C$BSUB = 000002     1894# 6439  6472  6533  6564  6678  6726  6765  7134  7196  7291  7348  7808
                    7883  7912  7937  7962  7989  8019  8069  8083  8111
C$CEFG = 000045     1894#
C$CLCK = 000062     1894#
C$CLEA = 000012     1894# 5845
C$CLOS = 000035     1894#
C$CLP1 = 000006     1894#
C$CVEC = 000036     1894#
C$DCLN = 000044     1894#
C$DODU = 000051     1894# 5821
C$DRPT = 000024     1894#
C$DU   = 000053     1894# 5873
C$EDIT = 000003     1894# 1976
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- USER SYMBOLS

```
C$ERDF= 000055   1894#  3142  3152  3166  3176  3287  3297  3459  3469  3483  3493  3532  3542
                 3580  3590  3602  3612  3762  3771  3794  3804  3817  3827  3840  3850  3863
                 3873  4252  4262  4299  4368  6026  6077  6162  6332  6375  6392  6713  6752
                 6791  7164  7176  7226  7238  7443  7469  7516  7535  7595  7655  7819  7835
                 7895  7924  7949  7975  8000  8032  8044  8055  8094  8122  8134  8146  8209
                 8273  8337  8401  8465  8528  8592  8708  8724  8790
C$ERHR= 000056   1894#
C$ERRO= 000060   1894#
C$ERSF= 000054   1894#
C$ERSO= 000057   1894#
C$ESCA= 000010   1894#  7169  7181  7231  7243  7824  7840  7900  8037  8127  8139
C$ESEG= 000005   1894#
C$ESUB= 000003   1894#  6464  6500  6558  6589  6720  6759  6798  7188  7250  7340  7397  7867
                 7905  7931  7956  7982  8007  8062  8076  8101  8153
C$ETST= 000001   1894#  5953  6084  6169  6203  6239  6277  6406  6507  6592  6646  6801  6844
                 6964  7100  7254  7401  7476  7549  7609  7669  7709  7761  8159  8223  8287
                 8351  8415  8479  8543  8607  8653  8822
C$EXIT= 000032   1894#
C$GETB= 000026   1894#
C$GETW= 000027   1894#
C$GMAN= 000043   1894#  5746
C$GPHR= 000042   1894#  5661
C$GPLO= 000030   1894#
C$GPRI= 000040   1894#
C$INIT= 000011   1894#  5759
C$INLP= 000020   1894#
C$MANI= 000050   1894#  5719
C$MEM = 000031   1894#
C$MSG = 000023   1894#  4709  4782  4892  5001  5119  5228  5330  5447  5549
C$OPEN= 000034   1894#
C$PNTB= 000014   1894#  4705  4721  4727  4735  4743  4796  4802  4810  4818  4905  4913  4919
                 4927  5015  5023  5029  5037  5045  5132  5140  5146  5154  5242  5248  5256
                 5343  5351  5357  5365  5373  5461  5467  5475
C$PNTF= 000017   1894#  4137  4151  4313  4329  5706  5728  5740  5869  6694  6702  6741  6780
                 6831  6839
C$PNTS= 000016   1894#
C$PNTX= 000015   1894#  4751  4761  4768  4778  4826  4836  4843  4853  4861  4871  4878  4888
                 4935  4945  4952  4962  4970  4980  4987  4997  5053  5063  5070  5080  5088
                 5098  5105  5115  5162  5172  5179  5189  5197  5207  5214  5224  5264  5274
                 5281  5291  5299  5309  5316  5326  5381  5391  5398  5408  5416  5426  5433
                 5443  5483  5493  5500  5510  5518  5528  5535  5545
C$QIO = 000377   1894#
C$RDBU= 000007   1894#
C$REFG= 000047   1894#  5621  5627  5633  5639
C$RESE= 000033   1894#  5862
C$REVI= 000003   1894#  1975
C$RFLA= 000021   1894#
C$RPT = 000025   1894#  5568
C$SEFG= 000046   1894#
C$SPRI= 000041   1894#  5812
C$SVEC= 000037   1894#
C$TPRI= 000013   1894#
C32BCC= 000040   2452#  2458
C32ENB= 000004   2455#
DDC   = 000100   2439#
DDCMP = 000001   2303#  2375#  5982  6037  6065  6616  6623  6628  6633  6876  6997  7007  7138
```

```
                        7200    7295    7313    7352    7370    7427    7503    7578    7638    7693    7730    7859    7861
                        8255    8447    8511    8575    8680
DEVMAP  002414          2604#   5647*   5667*
DEVPTR  002416          2605#   5652*   5665*   5667    5668*
DFPTBL  002226 G        2091#
DH1     015137          4652#   4731    4923    5252    5361
DH2     015161          4656#   4747    4822    4931    5049    5158    5260    5377    5479
DH3     015210          4660#   4746    4821    4930    5048    5157    5259    5376    5478
DH4     015246          4665#   4764    4839    4948    5066    5175    5277    5394    5496
DH5     015310          3093    4671#
DH6     015313          3096    4672#
DH7     015316          4673#   4857    4966    5084    5193    5295    5412    5514
DH8     015350          4678#   4856    4965    5083    5192    5294    5411    5513
DH9     015407          4684#   4874    4983    5101    5210    5312    5429    5531
DIAGMC= 000000          1894
DISILO  002404          2600#   3019    3056    3070    5606*   6224*   6311*   6402*   7740*   7757*
DISSI = 000040          2286#   2358#   6309    6311    7738    7740
DTR   = 000100          2275#   2345#   4219    5693    5696    8022    8027
EBLK  = 000002          2338#   3810    3812    3822
EF.CON= 000036 G        2188#   5638
EF.NEW= 000035 G        2189#   5632
EF.PWR= 000034 G        2190#
EF.RES= 000037 G        2187#   5626
EF.STA= 000040 G        2186#   5620
EM10    013606          3178    4495#
EM11    013623          3289    4498#   6334
EM12    013640          3299    4501#
EM17    013655          3461    4504#   7445    7537    7597    7657    8211    8275    8339    8403    8467    8530    8594
EM18    013672          3471    4507#
EM19    013707          3485    4510#
EM2     013461          4477#   7837
EM20    013724          3495    4513#
EM21    013741          3534    4516#
EM22    013756          3544    4519#
EM28    013773          3592    4522#
EM29    014010          3582    4525#
EM3     013520          4483#   7471    7821
EM30    014025          3614    4528#
EM31    014042          3604    4264    4531#   6377
EM34    014057          3773    4534#
EM35    014105          3796    4538#
EM36    014121          3806    4541#
EM37    014135          3819    4544#
EM38    014152          3829    4547#
EM39    014167          3842    4550#   6164
EM40    014203          3852    4254    4553#
EM41    014217          3865    4556#   6028    6079
EM42    014234          3875    4559#
EM43    014251          4562#   6715
EM44    014303          4567#   6754
EM45    014335          4572#   6793
EM46    014367          4577#
EM47    014422          4582#   6394
EM48    014455          4587#   7166
EM49    014510          4592#   7178
EM50    014543          4597#   7240
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- USER SYMBOLS

```
EM51     014577            4602#   7228
EM52     014633            4301    4607#   7977
EM54     014667            3764    4612#
EM55     014711            4616#   8034
EM56     014725            4619#   7897    8057
EM57     014742            4622#   8046
EM58     014757            4625#   8096
EM59     014773            4628#   8002
EM60     015010            4631#   8124    8710    8726
EM61     015024            4370    4633#   8136
EM62     015037            4635#   8148
EM63     015054            4638#   7926
EM64     015071            4641#   7951
EM65     015106            4644#   8792
EM66     015122            4646#   7518
EM7      013537            3144    4486#
EM8      013554            3154    4489#
EM9      013571            3168    4492#
ENAX   = 000004            2289#   2361#   3018    3069
ENDIT    022776            5642    5756#
ENDPAT   002653            2764#   6502
EOM    = 000002            2261#
ERRFLG   002330            2578#
ERROR1   002376            2597#   3014*   3027*   3052*   3078*   5608*
ERR1     015452 G          4698#
ERR10    021502 G          5454#   6378
ERR2     015504 G          4714#   6716    6755    6794    7472    7822    7838
ERR3     016012 G          4789#   6395
ERR4     016474 G          3145    3155    3169    3179    3290    3300    3462    3472    3486    3496    3535    3545    3765
                           4302    4371    4899#
ERR5     017152 G          5008#
ERR6     017664 G          3583    3593    3605    3615    4255    4265    5126#
ERR7     020342 G          5235#   6029    6080    6165    6335    7167    7179    7229    7241    7446    7519    7538    7598
                           7658    7898    7927    7952    7978    8003    8035    8047    8058    8097    8125    8137    8149
                           8212    8276    8340    8404    8468    8531    8595    8711    8727    8793
ERR8     020774 G          3774    3797    3807    3820    3830    3843    3853    3866    3876    5337#
EVL    = 000004 G          2206#
ESEND  = 002100            1894#
ESLOAD= 000035             1894#   1999
FMT1     012530            4387#   4702    4718    4793    4910    5012    5137    5239    5348    5458
FMT10    012773            4418#   4902    5129    5340
FMT11    013024            4423#   5020
FMT12    013063            4429#   6699
FMT13    013146            4438#   6738
FMT14    013212            4444#   6777
FMT15    013256            4450#   6836
FMT16    023000            5725    5761#
FMT17    023117            5737    5775#
FMT18    013325            4457#   6691    6828
FMT19    013367            4134    4463#
FMT2     012540            4389#   4724    4799    4916    5026    5143    5245    5354    5464
FMT26    013420            4148    4468#
FMT27    023402            5866    5875#
FMT3     012562            4392#   4740    4815    5042    5370
FMT4     012624            4398#   4748    4823    4858    4932    4967    5050    5085    5159    5194    5261    5296    5378
                           5413    5480    5515
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- USER SYMBOLS

```
FMT5     012637   4400# 4758  4833  4868  4942  4977  5060  5095  5169  5204  5271  5306  5388
                  5423  5490  5525
FMT6     012667   4405# 4775  4850  4885  4959  4994  5077  5112  5186  5221  5288  5323  5405
                  5440  5507  5542
FMT7     012722   4410# 4732  4924  5253  5362
FMT8     012732   4412# 4807  5034  5151  5472
FMT9     012766   4417# 4765  4840  4875  4949  4984  5067  5102  5176  5211  5278  5313  5395
                  5430  5497  5532
FRSPAS   002366   2593# 5650* 6683  6730  6769  6820
FRSTIM   002364   2592# 5610  5617*
F$AU   = 000015   1894# 5895  5897
F$AUTO= 000020    1894# 5808  5825
F$BGN  = 000040   1894# 1900  4698  4714  4789  4899  5008  5126  5235  5337  5454  5564  5584
                  5602  5808  5840  5859  5895  5925  5952  5975  6083  6108  6168  6184  6202
                  6219  6238  6255  6276  6296  6405  6433  6438  6463  6471  6499  6506  6527
                  6532  6557  6563  6588  6591  6612  6645  6672  6677  6719  6725  6758  6764
                  6797  6800  6815  6843  6868  6963  6990  7099  7126  7133  7169  7181  7187
                  7195  7231  7243  7249  7253  7283  7290  7339  7347  7396  7400  7418  7475
                  7494  7548  7569  7608  7629  7668  7688  7708  7725  7760  7794  7807  7824
                  7840  7866  7882  7900  7904  7911  7930  7936  7955  7961  7981  7988  8006
                  8018  8037  8061  8068  8075  8082  8100  8110  8127  8139  8152  8158  8182
                  8222  8246  8286  8310  8350  8374  8414  8458  8478  8502  8542  8566  8606
                  8625  8652  8673  8821  8842  8977  9053
F$CLEA= 000007     1894# 5840  5844
F$DU   = 000016   1894# 5859  5872
F$END  = 000041   1894# 1900  4710  4783  4893  5002  5120  5229  5331  5448  5550  5569  5760
                  5827  5846  5874  5899  5925  5952  5954  5975  6083  6085  6108  6168  6170
                  6184  6202  6204  6219  6238  6240  6255  6276  6278  6296  6405  6407  6433
                  6438  6463  6465  6471  6499  6501  6506  6508  6527  6532  6557  6559  6563
                  6588  6590  6591  6593  6612  6645  6647  6672  6677  6719  6721  6725  6758
                  6760  6764  6797  6799  6800  6802  6815  6843  6845  6868  6963  6965  6990
                  7099  7101  7126  7133  7169  7181  7187  7189  7195  7231  7243  7249  7251
                  7253  7255  7283  7290  7339  7341  7347  7396  7398  7400  7402  7418  7475
                  7477  7494  7548  7550  7569  7608  7610  7629  7668  7670  7688  7708  7710
                  7725  7760  7762  7794  7807  7824  7840  7866  7868  7882  7900  7904  7906
                  7911  7930  7932  7936  7955  7957  7961  7981  7983  7988  8006  8008  8018
                  8037  8061  8063  8068  8075  8077  8082  8100  8102  8110  8127  8139  8152
                  8154  8158  8160  8182  8222  8224  8246  8286  8288  8310  8350  8352  8374
                  8414  8416  8438  8478  8480  8502  8542  8544  8566  8606  8608  8625  8652
                  8654  8673  8821  8823  8884  9002  9053
F$HARD= 000004     1894# 8842  8882
F$HW   = 000013   1894# 2089  2101
F$INIT= 000006    1894# 5602  5758
F$JMP  = 000050   1894#
F$MOD  = 000000   1894# 1900  9053
F$MSG  = 000011   1894# 4698  4708  4714  4781  4789  4891  4899  5000  5008  5118  5126  5227
                  5235  5329  5337  5446  5454  5548
F$PROT= 000021    1894# 5584  5589
F$PWR  = 000017   1894#
F$RPT  = 000012   1894# 5564  5567
F$SEG  = 000003   1894#
F$SOFT= 000005    1894# 8977  9000
F$SRV  = 000010   1894#
F$SUB  = 000002   1894# 6439  6463  6472  6499  6533  6557  6564  6588  6678  6719  6726  6758
                  6765  6797  7134  7187  7196  7249  7291  7339  7348  7396  7808  7866  7883
                  7904  7912  7930  7937  7955  7962  7981  7989  8006  8019  8061  8069  8075
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- USER SYMBOLS

```
                        8083    8100    8111    8152
F$SW   = 000014         1894#   2115    2125
F$TEST= 000001          1894#   5926    5952    5976    6083    6109    6168    6185    6202    6220    6238    6256    6276
                        6297    6405    6434    6506    6528    6591    6613    6645    6673    6800    6816    6843    6869
                        6963    6991    7099    7127    7253    7284    7400    7419    7475    7495    7548    7570    7608
                        7630    7668    7689    7708    7726    7760    7795    8158    8183    8222    8247    8286    8311
                        8350    8375    8414    8439    8478    8503    8542    8567    8606    8626    8652    8674    8821
GETALL  004214          3091#   3139    3149    3163    3173    3284    3294    3456    3466    3480    3490    3529    3539
                        3577    3587    3599    3609    3759    3768    3791    3801    3814    3824    3837    3847    3860
                        3870    4249    4259    4296    4365    6023    6074    6159    6329    6372    6389    6710    6749
                        6788    7161    7173    7223    7235    7440    7466    7513    7532    7592    7652    7816    7832
                        7892    7921    7946    7972    7997    8029    8041    8052    8091    8119    8131    8143    8206
                        8270    8334    8398    8462    8525    8589    8705    8721    8787
GETPRM  022344          5643    5655#   5666
GETREG  003504          2968#   3097
GOAH   = 000010         2259#
GOODAT  002356          2589#   3751*   3752*   3781*   3782*   4739    4814    5041    5369    6385*   6386*   6387*   6708*
                        6747*   6786*   7464*   7511*   7814*   7830*   7847*   7861*
G$CNTO= 000200          1894#
G$DELM= 000372          1894#
G$DISP= 000003          1894#
G$EXCP= 000400          1894#
G$HILI= 000002          1894#
G$LOLI= 000001          1894#
G$NO   = 000000         1894#   5749
G$OFFS= 000400          1894#   5749    8846    8851    8857    8863    8869    8875    8981    8985    8989    8993
G$OFSI= 000376          1894#   5749    8846    8851    8857    8863    8869    8875    8981    8985    8989    8993
G$PRMA= 000001          1894#   8846
G$PRMD= 000002          1894#   8851    8857    8863    8869    8875    8993
G$PRML= 000000          1894#   5749    8981    8985    8989
G$RADA= 000140          1894#
G$RADB= 000000          1894#
G$RADD= 000040          1894#
G$RADL= 000120          1894#   5749    8981    8985    8989
G$RADO= 000020          1894#   8846    8851    8857    8863    8869    8875    8993
G$XFER= 000004          1894#
G$YES  = 000010         1894#   8846    8851    8857    8863    8869    8875    8981    8985    8989    8993
HDX    = 000020         2277#   2347#   7461    7694    7734    8085    8089
HELP   = 000001         1885#   1924    2017    2071    2856
HOE    = 100000 G       2219#
H325   = 000001         2529#   7853    7869    8048
H3250  = 000002         2530#   7855    7872
H3251  = 000003         2531#
H3254  = 000000         2528#
IACT   = 000100         2333#   3527    3537
IACTIV  006246          3517#   3668    3689    6127    6270    6455    6459    6549    6553    7457    7755
IBE    = 010000 G       2216#
IC     = 000200         2267#   2332#   6008    6152    6268    8779
ICIR   = 000010         2372#   3478    3488    7859    7861
IDL    = 000010         2442#
IDLE   = 000040         2298#   6188
IDU    = 000040 G       2209#
IER    = 020000 G       2217#
IERR   = 000002         2302#   3713    3719
INITRN  005226          3324#   5927    5980    6035    6063    6113    6137    6186    6221    6257    6298    6441    6477
                        6535    6566    6614    6995    7030    7065    7136    7198    7293    7350    7691    8112    8627
```

```
                    8678
INTFLG  002326      2576#
INTGRL= 000010      2454#   2458    4091    4106    4123    4141    4196    7423    7500    7575    7635
INTMDM= 000004      2532#   5685
IRDY  = 000020      2335#   3454    3464    3679    7436    7528    7588    7648    8202    8266    8330    8394    8458
                    8521    8585
ISIRDY  005762      3444#   3670    3683    3687    3691    6272    7704
ISMANI  037670      8982    9003#
ISPRNT  037746      8986    9011#
ISR   = 000100 G    2210#
ISWPAK  040015      8990    9018#
IXE   = 004000 G    2215#
I$AU  = 000041      1894#   5895#   5899#
I$AUTO= 000041      1894#   5808#   5827#
I$CLN = 000041      1894#   5840#   5846#
I$DU  = 000041      1894#   5859#   5874#
I$HRD = 000041      8842#   8884#
I$INIT= 000041      1894#   5602#   5760#
I$MOD = 000041      1894#   1900#   9053#
I$MSG = 000041      1894#   4698#   4710#   4714#   4783#   4789#   4893#   4899#   5002#   5008#   5120#   5126#   5229#
                    5235#   5331#   5337#   5448#   5454#   5550#
I$PROT= 000040      1894#   5584#
I$PTAB= 000041      1894#
I$PWR = 000041      1894#
I$RPT = 000041      1894#   5564#   5569#
I$SEG = 000041      1894#   5925    5975    6108    6184    6219    6255    6296    6433    6438    6471    6527    6532
                    6563    6612    6672    6677    6725    6764    6815    6868    6990    7126    7133    7195    7283
                    7290    7347    7418    7494    7569    7629    7688    7725    7794    7807    7882    7911    7936
                    7961    7988    8018    8068    8082    8110    8182    8246    8310    8374    8438    8502    8566
                    8625    8673
I$SETU= 000041      1894#
I$SFT = 000041      8977#   9002#
I$SRV = 000041      1894#
I$SUB = 000041      1894#   5925    5975    6108    6184    6219    6255    6296    6433    6438#   6463#   6465#   6'71#
                    6499#   6501#   6527    6532#   6557#   6559#   6563#   6588#   6590#   6612    6672    6677#   6719#
                    6721#   6725#   6758#   6760#   6764#   6797#   6799#   6815    6868    6990    7126    7133#   7169
                    7181    7187#   7189#   7195#   7231    7243    7249#   7251#   7283    7290#   7339#   7341#   7347#
                    7396#   7398#   7418    7494    7569    7629    7688    7725    7794    7807#   7824    7840    7866#
                    7868#   7882#   7900    7904#   7906#   7911#   7930#   7932#   7936#   7955#   7957#   7961#   7981#
                    7983#   7988#   8006#   8008#   8018#   8037    8061#   8063#   8068#   8075#   8077#   8082#   8100#
                    8102#   8110#   8127    8139    8152#   8154#   8182    8246    8310    8374    8438    8502    8566
                    8625    8673
I$TST = 000041      1894#   5925#   5952#   5954#   5975#   6083#   6085#   6108#   6168#   6170#   6184#   6202#   6204#
                    6219#   6238#   6240#   6255#   6276#   6278#   6296#   6405#   6407#   6433#   6438    6471    6506#
                    6508#   6527#   6532    6563    6591#   6593#   6612#   6645#   6647#   6672#   6677    6725    6764
                    6800#   6802#   6815#   6843#   6845#   6868#   6963#   6965#   6990#   7099#   7101#   7126#   7133
                    7195    7253#   7255#   7283#   7290    7347    7400#   7402#   7418#   7475#   7477#   7494#   7548#
                    7550#   7569#   7608#   7610#   7629#   7668#   7670#   7688#   7708#   7710#   7725#   7760#   7762#
                    7794#   7807    7882    7911    7936    7961    7988    8018    8068    8082    8110    8158#   8160#
                    8182#   8222#   8224#   8246#   8286#   8288#   8310#   8350#   8352#   8374#   8414#   8416#   8438#
                    8478#   8480#   8502#   8542#   8544#   8566#   8606#   8608#   8625#   8652#   8654#   8673#   8821#
                    8823#
I422  = 000200      2450#   2458    4091    4110    4156    4172    4174    4179    4202    7424    7500    7575    7634
J$JMP = 000167      1894#
LDBYTS  010562      3998#   5930    7139    7201    7299    7356    8681    8687
LDMSG1  010644      4023#   7430    7522    7582    7642
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- USER SYMBOLS

```
LDTXSI  004662          3223#  3344   3345   3403   3910   3923   3924   3980   4006   4056   5934   5935
LM1     012326          4310   4335#
LM2     012357          4326   4340#
LOADAT  002362          2591#  5018
LODATA  010172          3904#  8195   8259   8323   8387   8451   8515   8579
LODMSG  010504          3973#  4028   6116   6140   6189   6617   6879   6909   6940   7001   7036   7071   7697
                        8696
LODSIL  010734          3907   4025   4052#  5983   5990   6012   6015   6038   6045   6066   6225   6260   6444
                        6447   6450   6480   6483   6486   6538   6541   6544   6569   6572   6575   7142   7204
                        7296   7302   7353   7359   8630   8633   8684   8690   8693
LOE   = 040000 G        2218#
LOGDEV  002316          2572#  5649*  5656*  5657   5660   5820   5865
LOOPBK  037176          8870   8907#
LOOPIN  003562          2994#
LOT   = 000010 G        2207#
LPBCON  002444          2619#  5680*  5685   5689*  7797   7853   7855   7869   7872   8048
LULOOP= 000010          2238#  2903   3098   3252   3353   3955   3957   4100   4229   4283   4353   6817   7431
                        7506   7521   7523   7826   7887   7968   8025
LULP  = 000040          2269#  2334#  6313
LUREG   002254          2500   2501   2502   2503   2504   2505   2506   2507   2508   2509   2510   2511   2512
                        2513   2514   2515   2566#
LUR10 = 002254          2500#  2970   4757   4832   4941   5059   5168   5270   5387   5489
LUR11 = 002256          2501#  4756   4831   4940   5058   5167   5269   5386   5488
LUR12 = 002260          2502#  4755   4830   4939   5057   5166   5268   5385   5487
LUR13 = 002262          2503#  4754   4829   4938   5056   5165   5267   5384   5486
LUR14 = 002264          2504#  4774   4849   4958   5076   5185   5287   5404   5506
LUR15 = 002266          2505#  4773   4848   4957   5075   5184   5286   5403   5505
LUR16 = 002270          2506#  4772   4847   4956   5074   5183   5285   5402   5504
LUR17 = 002272          2507#  4771   4846   4955   5073   5182   5284   5401   5503
LUSWI1  002434          2615#  5676*  6706   6708
LUSWI2  002436          2616#  5677*  6745   6747
LUSWI3  002440          2617#  5678*  6784   6786
LU2MOD  002000 G        1900#
L$ACP   002110 G        2006#
L$APT   002036 G        1964#
L$AU    023434 G        1991   5895#
L$AUT   002070 G        1990#
L$AUTO  023270 G        2007   5808#
L$CCP   002106 G        2004#
L$CLEA  023350 G        2005   5840#
L$CO    002032 G        1960#
L$DEPO  002011 G        1942#
L$DESC  003154 G        1997   2841#
L$DESP  002076 G        1996#
L$DEVP  002060 G        1982#
L$DISP  002124 G        1967   2037#
L$DLY   002116 G        2012#
L$DTP   002040 G        1966#
L$DTYP  002034 G        1962#
L$DU    023352 G        1993   5859#
L$DUT   002072 G        1992#
L$DVTY  003146 G        1983   2833#
L$EF    002052 G        1977#
L$ENVI  002044 G        1970#
L$ETP   002102 G        2000#
L$EXP1  002046 G        1972#
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- USER SYMBOLS

```
L$EXP4  002064 G        1986#
L$EXP5  002066 G        1988#
L$HARD  036710 G        1949    8842    8843#
L$HIME  002120 G        2014#
L$HPCP  002016 G        1948#
L$HPTP  002022 G        1952#
L$HW    002226 G        1953    2089    2090#
L$ICP   002104 G        2002#
L$INIT  022144 G        2003    5602#
L$LADP  002026 G        1956#
L$LAST  040244 G        1957    9058#
L$LOAD  002100 G        1998#
L$LUN   002074 G        1994#
L$MREV  002050 G        1974#
L$NAME  002000 G        1931#
L$PRIO  002042 G        1968#
L$PROT  022136 G        2009    5584#
L$PRT   002112 G        2008#
L$REPP  002062 G        1984#
L$REV   002010 G        1940#
L$RPT   022134 G        5564#
L$SOFT  037634 G        1951    8977    8978#
L$SPC   002056 G        1980#
L$SPCP  002020 G        1950#
L$SPTP  002024 G        1954#
L$STA   002030 G        1958#
L$SW    002244 G        1955    2115    2116#
L$TEST  002114 G        2010#
L$TIML  002014 G        1946#
L$UNIT  002012 G        1944#   5657
L10000  002242          2089    2101#
L10001  002254          2115    2125#
L10002  015502          4708#
L10003  016010          4781#
L10004  016472          4891#
L10005  017150          5000#
L10006  017662          5118#
L10007  020340          5227#
L10010  020772          5329#
L10011  021500          5446#
L10012  022132          5548#
L10013  022134          5567#
L10015  022776          5758#
L10016  023346          5825#
L10017  023350          5844#
L10020  023400          5872#
L10021  023434          5897#
L10022  023564          5952#
L10023  024222          6083#
L10024  024454          6168#
L10025  024536          6202#
L10026  024626          6238#
L10027  024732          6276#
L10030  025502          6405#
L10031  025742          6506#
L10032  025604          6463#
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- USER SYMBOLS

```
L10033  025724          6499#
L10034  026146          6591#
L10035  026044          6557#
L10036  026144          6588#
L10037  026302          6645#
L10040  026754          6800#
L10041  026466          6719#
L10042  026620          6758#
L10043  026752          6797#
L10044  027070          6843#
L10045  027426          6963#
L10046  030036          7099#
L10047  030510          7253#
L10050  030260          7170    7182    7187#
L10051  030502          7232    7244    7249#
L10052  031212          7400#
L10053  030746          7339#
L10054  031204          7396#
L10055  031474          7475#
L10056  031746          7548#
L10057  032132          7608#
L10060  032316          7668#
L10061  032422          7708#
L10062  032640          7760#
L10063  034270          8158#
L10064  033174          7825    7841    7866#
L10065  033316          7901    7904#
L10066  033404          7930#
L10067  033472          7955#
L10070  033566          7981#
L10071  033646          8006#
L10072  034020          8038    8061#
L10073  034050          8075#
L10074  034130          8100#
L10075  034262          8128    8140    8152#
L10076  034462          8222#
L10077  034654          8286#
L10100  035046          8350#
L10101  035240          8414#
L10102  035432          8478#
L10103  035620          8542#
L10104  036006          8606#
L10105  036120          8652#
L10106  036704          8821#
L10107  037002          8842    8883#
L10110  037670          8977    9001#
MAINT1= 000010          2278#   5693    7964
MAINT2= 000004          2279#   4160    4191    5696    8022
MCLK  = 000002          2374#   7852
MCLR  = 000100          2236#   2897    2898
MIFLAG  002244          2119#   5715
MLWBYT  002450          2621#   2905    4215    5690*   5693*   5696*   7695    7735
MODINT  002410          2602#   4091*   4123*   4161*   4165*   4172*   4188*   8188    8252    8316    8380    8444    8508
                        8572
MODR  = 000010          2348#   7461    7812    7814    8039
MPCSR   002424          2610#   4700    4716    4791    4908    5010    5135    5237    5346    5456    5669*   5736    5815
```

```
                        6690    6827
MSG1    002654          2773#   4029    4031    4034    6880    6910    6941    7002    7037    7072    7698    8697
MSG2    002702          2785#   6618
MSG3    002716          2792#   6117    6141    6190
MSTCLR  003262          2894#   3329    5950    6081    6166    6200    6236    6274    6302    6403    6504    6643    6679
                        6727    6766    6816    6873    6903    6934    6961    7097    7251    7398    7421    7497    7572
                        7632    7706    7727    7758    7809    7884    7913    7938    7963    7990    8020    8070    8084
                        8156    8185    8249    8313    8377    8441    8505    8569    8650    8819
MVIOX = 021000          2543#   2929
MVIXO = 122000          2544#   2951
NEWST   022320          5635    5648#   5658
OACT  = 000100          2321#   3282    3292    3357    6324
OACTIV  005040          3272#   3332    3349    3364    3414    7455
OC    = 000200          2258#   2320#
OCOR  = 000020          2371#   3161    3171    3369
OP    = 000002          2456#   2458    4091    7424    7500    7575    7635
ORDY  = 000020          2323#   3137    3147
OSIRDY  004350          3127#   3330    3347    3374    3418    3424    7702
OVRR  = 000010          2336#   3856    3858    3868    6021    6072
O$APTS= 000000          1894#   1958
O$AU  = 000001          1894#   1923#   1990
O$BGNR= 000000          1894#   1984
O$BGNS= 000001          1894#   1923#   1950
O$DU  = 000001          1894#   1923#   1992
O$ERRT= 000000          1894#   2000
O$GNSW= 000001          1894#   1923#   1954
O$POIN= 000001          1894#   1923#   2016
O$SETU= 000000          1894#   1944    9056
PATA    002543          2678#   3912    8682    8688    8734    8756
PATB    002567          2701#   3917    8739    8761
PATCH   040132          9042#
PATQ    002577          2713#   7140    7202
PATR    002607          2723#   7300    7322    7357    7379
PATS    002626          2740#   5931    5938    5943
PATT    002646          2758#   6468
PNT   = 001000 G        2213#
POLL  = 000200          2274#   7914
PRI   = 002000 G        2214#
PRIOR   002322          2574#
PRI00 = 000000 G        2202#
PRI01 = 000040 G        2201#
PRI02 = 000100 G        2200#
PRI03 = 000140 G        2199#
PRI04 = 000200 G        2198#
PRI05 = 000240 G        2197#
PRI06 = 000300 G        2196#
PRI07 = 000340 G        2195#   5811    5814
PRNFLG  002246          2120#   6685    6732    6771    6822
PSTACK  002320          2573#   3181    3302    3498    3547    3617    3885    4267    5604*
RAB   = 000004          2337#   3833    3835    3845    6157
RABT  = 000004          2397#   4247
RAX15   002342          2583#   3031*   3032*   3102    4141    4156    4163    4174    4179    4182    4196    4199    4202
                        4208    6835
RAX16   002344          2584#   3035*   3036*   3104    3575    3585    3597    3607    4247    4257    6365    6380    6388
RCVBUF  002746          2810#   3911    4032    7433    7452    7525    7544    7585    7604    7645    7664    8196*   8199
                        8218    8260*   8263    8282    8324*   8327    8346    8388*   8391    8410    8452*   8455    8474
```

```
                    8518    8538    8582    8602
RCV1ST  006766      3657#   5986    6018    6041    6069    6119    6129    6143    6154    6192    6228    6263    6489
                    6578    6635
RDALL = 000004      2301#   6568
RDAX  = 000020      2287#   2359#   3018
RDRXSI  006706      3633#   3744    6020    6071    6156
READAX  003612      3012#   3101    3572    4102    4246    6364    6819
READLU  003360      2921#   2972    3022    3030    3034    3073    3134    3158    3279    3356    3368    3451    3475
                    3524    3635    3639    3678    3756    4290    4306    4316    4359    6323    6681    6729    6768
                    7155    7217    7435    7460    7508    7527    7587    7647    7811    7827    7844    7851    7889
                    7918    7943    7969    7994    8026    8088    8116    8201    8265    8329    8393    8457    8520
                    8584    8702    8718    8784
READY = 000200      2356#   3023    3074
REDBYT  002336      2581#   2933*   2934*   2973    3023    3031    3035    3074    3137    3147    3161    3171    3282
                    3292    3357    3369    3454    3464    3478    3488    3527    3537    3636    3640    3679    3757
                    4291    4307    4317    4360    6324    6682*   6698    6706    6709    6737    6745    6748    6776
                    6784    6787    7159    7171    7221    7233    7436    7461*   7462    7465    7509    7512    7528
                    7588    7648    7812    7815    7828    7831    7845    7848    7852*   7858*   7859    7862    7890
                    7919    7944    7970    7995    8027    8039    8050    8089    8117    8129    8141    8202    8266
                    8330    8394    8458    8521    8585    8703    8719    8785
REDDAT  002452      2624#
REGNUM  002352      2587#   2896    2904*   2907*   2923    2949    2969    2971*   2975*   2976    2978*   3013    3015*
                    3029*   3033*   3037*   3051    3053*   3058*   3062*   3066*   3079*   3127    3133*   3157*   3180*
                    3186*   3223    3225*   3228*   3231*   3272    3278*   3301*   3307*   3325    3340*   3352*   3366*
                    3378*   3444    3450*   3474*   3497*   3503*   3517    3523*   3546*   3552*   3633    3634*   3638*
                    3641*   3659    3662*   3694*   3710    3711*   3722*   3739    3757    3767*   3777*   3884*   3889*
                    3940    3945*   3959*   4086    4214*   4218*   4228*   4286*   4357*   4730    4922    5019    5251
                    5360    6007*   6062*   6151*   6267*   6308*   6312*   6320*   6680*   6728*   6767*   7131*   7193*
                    7288*   7345*   7432*   7459*   7507*   7524*   7584*   7644*   7690*   7733*   7737*   7810*   7850*
                    7888*   7915*   7917*   7940*   7942*   7965*   7967*   7992*   8021*   8071*   8086*   8115*   8198*
                    8262*   8326*   8390*   8454*   8517*   8581*   8701*   8778*   8781*
REG0    002462      2634#
REG1    002464      2635#
REG2    002466      2636#   5742*   5748    5753
REG3    002470      2637#
REG4    002472      2638#
REG5    002474      2639#
REG6    002476      2640#
REG7    002500      2641#
REOM  = 000002      2398#   3597    3607    4257    6365    6387
RERR  = 000200      2392#
RETADR  002334      2580#   3182    3303    3499    3548    3618    3886    4139    4268    4303    4372    5926*   5976*
                    6109*   6185*   6220*   6256*   6297*   6440*   6473*   6534*   6565*   6613*   6872*   6994*   7135*
                    7197*   7292*   7349*   7420*   7496*   7571*   7631*   7689*   7726*   7796*   8184*   8248*   8312*
                    8376*   8440*   8504*   8568*   8626*   8674*
RING  = 000200      2344#   7461    7812    7814    7890    8050
ROMI  = 000002      2240#   2880    2882    2883    2995    2997
ROMO  = 000004      2239#   2880    2882    2883    2995    2997
ROR   = 000010      2396#
RRDYTO= 000001      2549#   3014    3027
RSEOM   006434      3565#   3672
RSOM  = 000001      2399#   3575    3585
RTS   = 000040      2346#   8117    8703    8719    8785
RUN   = 000200      2235#   2898    2997    7886
RXABT = 002000      2490#   6125    6149    6234
RXBCC = 000400      2492#   3921    4036    7333    7390    8452    8745    8767    8811
```

```
RXEBL = 001000          2491#  5946   6125   6149   6198   6234   6495   6584   8196   8324   8388
RXLEN0= 000001          2468#  3745   7146   7208   7306   7363   8513   8577
RXLEN1= 000002          2467#  3745   7146   7208   7306   7363   8513   8577
RXLEN2= 000004          2466#  3745   7146   7208   7306   7363   8513   8577
RXOVR = 004000          2489#
RXWORD  002402          2599#  3636*  3637*  3640*  3748*  3749   3754   3778   3784   3789   3799   3812   3822
                        3835   3845   3858   3868   6021   6072   6157
RX0   = 000001          2315#  2387#
RX1   = 000002          2314#  2386#
RX2   = 000004          2313#  2385#
RX3   = 000010          2312#  2384#
RX4   = 000020          2311#  2383#
RX5   = 000040          2310#  2382#
RX6   = 000100          2309#  2381#
RX7   = 000200          2308#  2380#
R14NRW  002532          2664#
SAVE4   002372          2595#  5612*  5615   5822
SAVE6   002374          2596#  5613*  5616   5823
SAVLEN  002412          2603#  2909*  3745   3954*  5609*  7307*  7364*
SCRACH  002314          2571#
SEC   = 000020          2441#
SECA  = 000020          2299#  6443   6479   6537   6568   8629
SELFR = 000040          2276#  7939
SELSBY= 000002          2280#  7991
SEL4    002430          2613#  5672*  5673*
SEL6    002432          2614#  2881*  2996*  5674*  5675*
SETMTM  012070          4281#  7581   7641   8194   8258   8322   8386   8450   8514   8578
SETUP   010312          3939#  6303   6874   6904   6935   7425   7501   7576   7636   7728   8189   8253   8317
                        8381   8445   8509   8573
SFPTBL  002244 G        2117#
SIGQ  = 000100          2369#  7858   7919
SIGR  = 000200          2368#  7858   7944
SOM   = 000001          2262#
STALL   004652          3210#  3254   3256   3355   3360   3373
STARES  002370          2594#  4126   4130   4145   5645*  5651*  5700
STARST  022310          5623   5629   5644#
STBY  = 000002          2350#  7461   7995
STEPLU= 000020          2237#  3253   3255   3354   3359   3372
STEPMP= 000001          2241#  2882   2883
STPCLK  003224          2879#  2931   2955
STPERR  007200          3710#  6622   6627   6632   7006   7041   7076   7312   7369
STPLU   004742          3244#  3420   3674   3717   3881   5936   5988   5993   6043   6048   6265   6321   6338
                        6342   6345   6356   6367   6453   6457   6547   6551   6620   6625   6630   6882   6912
                        6943   7004   7009   7039   7044   7074   7079   7148   7150   7153   7210   7212   7215
                        7309   7315   7319   7366   7372   7376   7700   7745   7750   7753   8636   8699   8713
                        8716   8782
STR   = 000040          2440#
STRIP = 000010          2300#  5982   6037   6065   6616   6623   6628   6633   6876   6997   7007   7138   7200
                        7295   7313   7352   7370   7427   7503   7578   7638   7693   8255   8447   8511   8575
                        8680
STS237  033650          7871   7874   8015#
SUBRPC  002324          2575#  3128   3129   3131*  3132*  3185*  3273   3274   3276*  3277*  3306*  3327*  3328*
                        3380*  3400*  3401*  3427*  3445   3446   3448*  3449*  3502*  3518   3519   3521*  3522*
                        3551*  3566   3567   3569*  3570*  3621*  3660*  3661*  3697*  3740*  3741*  3891*  3961*
                        4089*  4090*  4230*  4243*  4244*  4284*  4285*  4332*  4355*  4356*  4373*  4901   5128
                        5339   5605*
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- USER SYMBOLS

SVCGBL= 000000

1894# 1900 1907# 1931 1940 1942 1944 1946 1948 1950 1952 1954 1956
1958 1960 1962 1964 1966 1968 1970 1972 1974 1977 1980 1982 1984
1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010
2012 2014 2037 2090 2091 2116 2117 2833 2841 4698 4714 4789 4899
5008 5126 5235 5337 5454 5564 5584 5602 5808 5840 5859 5895 8843
8978 9058# 9059

SVCINS= 000001

1894# 1904# 1932 1933 1934 1935 1936 1937 1938 1939 1941 1943 1945
1947 1949 1951 1953 1955 1957 1959 1961 1963 1965 1967 1969 1971
1973 1975 1976 1978 1979 1981 1983 1985 1987 1989 1991 1993 1995
1997 1999 2001 2003 2005 2007 2009 2011 2013 2015 2036 2038 2039
2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052
2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065
2066 2067 2068 2069 2089 2115 2834 2835 2842 2849 3142 3143 3144
3145 3152 3153 3154 3155 3166 3167 3168 3169 3176 3177 3178 3179
3287 3288 3289 3290 3297 3298 3299 3300 3459 3460 3461 3462 3469
3470 3471 3472 3483 3484 3485 3486 3493 3494 3495 3496 3532 3533
3534 3535 3542 3543 3544 3545 3580 3581 3582 3583 3590 3591 3592
3593 3602 3603 3604 3605 3612 3613 3614 3615 3762 3763 3764 3765
3771 3772 3773 3774 3794 3795 3796 3797 3804 3805 3806 3807 3817
3818 3819 3820 3827 3828 3829 3830 3840 3841 3842 3843 3850 3851
3852 3853 3863 3864 3865 3866 3873 3874 3875 3876 4133 4134 4135
4136 4137 4138 4148 4149 4150 4151 4152 4252 4253 4254 4255 4262
4263 4264 4265 4299 4300 4301 4302 4310 4311 4312 4313 4314 4320
4326 4327 4328 4329 4330 4368 4369 4370 4371 4700 4701 4702 4703
4704 4705 4706 4709 4716 4717 4718 4719 4720 4721 4722 4724 4725
4726 4727 4728 4730 4731 4732 4733 4734 4735 4736 4738 4739 4740
4741 4742 4743 4744 4746 4747 4748 4749 4750 4751 4752 4754 4755
4756 4757 4758 4759 4760 4761 4762 4764 4765 4766 4767 4768 4769
4771 4772 4773 4774 4775 4776 4777 4778 4779 4782 4791 4792 4793
4794 4795 4796 4797 4799 4800 4801 4802 4803 4805 4806 4807 4808
4809 4810 4811 4813 4814 4815 4816 4817 4818 4819 4821 4822 4823
4824 4825 4826 4827 4829 4830 4831 4832 4833 4834 4835 4836 4837
4839 4840 4841 4842 4843 4844 4846 4847 4848 4849 4850 4851 4852
4853 4854 4856 4857 4858 4859 4860 4861 4862 4864 4865 4866 4867
4868 4869 4870 4871 4872 4874 4875 4876 4877 4878 4879 4881 4882
4883 4884 4885 4886 4887 4888 4889 4892 4901 4902 4903 4904 4905
4906 4908 4909 4910 4911 4912 4913 4914 4916 4917 4918 4919 4920
4922 4923 4924 4925 4926 4927 4928 4930 4931 4932 4933 4934 4935
4936 4938 4939 4940 4941 4942 4943 4944 4945 4946 4948 4949 4950
4951 4952 4953 4955 4956 4957 4958 4959 4960 4961 4962 4963 4965
4966 4967 4968 4969 4970 4971 4973 4974 4975 4976 4977 4978 4979
4980 4981 4983 4984 4985 4986 4987 4988 4990 4991 4992 4993 4994
4995 4996 4997 4998 5001 5010 5011 5012 5013 5014 5015 5016 5018
5019 5020 5021 5022 5023 5024 5026 5027 5028 5029 5030 5032 5033
5034 5035 5036 5037 5038 5040 5041 5042 5043 5044 5045 5046 5048
5049 5050 5051 5052 5053 5054 5056 5057 5058 5059 5060 5061 5062
5063 5064 5066 5067 5068 5069 5070 5071 5073 5074 5075 5076 5077
5078 5079 5080 5081 5083 5084 5085 5086 5087 5088 5089 5091 5092
5093 5094 5095 5096 5097 5098 5099 5101 5102 5103 5104 5105 5106
5108 5109 5110 5111 5112 5113 5114 5115 5116 5119 5128 5129 5130
5131 5132 5133 5135 5136 5137 5138 5139 5140 5141 5143 5144 5145
5146 5147 5149 5150 5151 5152 5153 5154 5155 5157 5158 5159 5160
5161 5162 5163 5165 5166 5167 5168 5169 5170 5171 5172 5173 5175
5176 5177 5178 5179 5180 5182 5183 5184 5185 5186 5187 5188 5189
5190 5192 5193 5194 5195 5196 5197 5198 5200 5201 5202 5203 5204
5205 5206 5207 5208 5210 5211 5212 5213 5214 5215 5217 5218 5219

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- USER SYMBOLS

	5220	5221	5222	5223	5224	5225	5228	5237	5238	5239	5240	5241	5242
	5243	5245	5246	5247	5248	5249	5251	5252	5253	5254	5255	5256	5257
	5259	5260	5261	5262	5263	5264	5265	5267	5268	5269	5270	5271	5272
	5273	5274	5275	5277	5278	5279	5280	5281	5282	5284	5285	5286	5287
	5288	5289	5290	5291	5292	5294	5295	5296	5297	5298	5299	5300	5302
	5303	5304	5305	5306	5307	5308	5309	5310	5312	5313	5314	5315	5316
	5317	5319	5320	5321	5322	5323	5324	5325	5326	5327	5330	5339	5340
	5341	5342	5343	5344	5346	5347	5348	5349	5350	5351	5352	5354	5355
	5356	5357	5358	5360	5361	5362	5363	5364	5365	5366	5368	5369	5370
	5371	5372	5373	5374	5376	5377	5378	5379	5380	5381	5382	5384	5385
	5386	5387	5388	5389	5390	5391	5392	5394	5395	5396	5397	5398	5399
	5401	5402	5403	5404	5405	5406	5407	5408	5409	5411	5412	5413	5414
	5415	5416	5417	5419	5420	5421	5422	5423	5424	5425	5426	5427	5429
	5430	5431	5432	5433	5434	5436	5437	5438	5439	5440	5441	5442	5443
	5444	5447	5456	5457	5458	5459	5460	5461	5462	5464	5465	5466	5467
	5468	5470	5471	5472	5473	5474	5475	5476	5478	5479	5480	5481	5482
	5483	5484	5486	5487	5488	5489	5490	5491	5492	5493	5494	5496	5497
	5498	5499	5500	5501	5503	5504	5505	5506	5507	5508	5509	5510	5511
	5513	5514	5515	5516	5517	5518	5519	5521	5522	5523	5524	5525	5526
	5527	5528	5529	5531	5532	5533	5534	5535	5536	5538	5539	5540	5541
	5542	5543	5544	5545	5546	5549	5568	5620	5621	5623	5626	5627	5629
	5632	5633	5635	5638	5639	5641	5660	5661	5662	5664	5703	5704	5705
	5706	5707	5719	5722	5725	5726	5727	5728	5729	5731	5736	5737	5738
	5739	5740	5741	5746	5747	5748	5749	5750	5751	5759	5811	5812	5820
	5821	5826	5845	5862	5865	5866	5867	5868	5869	5870	5873	5898	5953
	6026	6027	6028	6029	6077	6078	6079	6080	6084	6162	6163	6164	6165
	6169	6203	6239	6277	6332	6333	6334	6335	6375	6376	6377	6378	6392
	6393	6394	6395	6406	6439	6464	6472	6500	6507	6533	6558	6564	6589
	6592	6646	6678	6690	6691	6692	6693	6694	6695	6698	6699	6700	6701
	6702	6703	6713	6714	6715	6716	6720	6726	6737	6738	6739	6740	6741
	6742	6752	6753	6754	6755	6759	6765	6776	6777	6778	6779	6780	6781
	6791	6792	6793	6794	6798	6801	6827	6828	6829	6830	6831	6832	6835
	6836	6837	6838	6839	6840	6844	6964	7100	7134	7164	7165	7166	7167
	7169	7170	7176	7177	7178	7179	7181	7182	7188	7196	7226	7227	7228
	7229	7231	7232	7238	7239	7240	7241	7243	7244	7250	7254	7291	7340
	7348	7397	7401	7443	7444	7445	7446	7469	7470	7471	7472	7476	7516
	7517	7518	7519	7535	7536	7537	7538	7549	7595	7596	7597	7598	7609
	7655	7656	7657	7658	7669	7709	7761	7808	7819	7820	7821	7822	7824
	7825	7835	7836	7837	7838	7840	7841	7867	7883	7895	7896	7897	7898
	7900	7901	7905	7912	7924	7925	7926	7927	7931	7937	7949	7950	7951
	7952	7956	7962	7975	7976	7977	7978	7982	7989	8000	8001	8002	8003
	8007	8019	8032	8033	8034	8035	8037	8038	8044	8045	8046	8047	8055
	8056	8057	8058	8062	8069	8076	8083	8094	8095	8096	8097	8101	8111
	8122	8123	8124	8125	8127	8128	8134	8135	8136	8137	8139	8140	8146
	8147	8148	8149	8153	8159	8209	8210	8211	8212	8223	8273	8274	8275
	8276	8287	8337	8338	8339	8340	8351	8401	8402	8403	8404	8415	8465
	8466	8467	8468	8479	8528	8529	8530	8531	8543	8592	8593	8594	8595
	8607	8653	8708	8709	8710	8711	8724	8725	8726	8727	8790	8791	8792
	8793	8822	8842	8846	8847	8848	8849	8851	8852	8853	8854	8855	8857
	8858	8859	8860	8861	8863	8864	8865	8866	8867	8869	8870	8871	8872
	8873	8875	8876	8877	8878	8879	8882	8977	8981	8982	8983	8985	8986
	8987	8989	8990	8991	8993	8994	8995	8996	8997	9000	9055	9056	9057
SVCSUB= 000001	1894#	1906#	6438	6471	6532	6563	6677	6725	6764	7133	7195	7290	7347
	7807	7882	7911	7936	7961	7988	8018	8068	8082	8110			
SVCTAG= 000001	1894#	1908#	2101	2125	4708	4781	4891	5000	5118	5227	5329	5446	5548
	5567	5752	5758	5825	5844	5872	5897	5952	6083	6168	6202	6238	6276

```
                    6405    6463    6499    6506    6557    6588    6591    6645    6719    6758    6797    6800    6843
                    6963    7099    7187    7249    7253    7339    7396    7400    7475    7548    7608    7668    7708
                    7760    7866    7904    7930    7955    7981    8006    8061    8075    8100    8152    8158    8222
                    8286    8350    8414    8478    8542    8606    8652    8821    8883    9001
SVCTST= 000001       1894#   1905#   5925    5975    6108    6184    6219    6255    6296    6433    6527    6612    6672
                    6815    6868    6990    7126    7283    7418    7494    7569    7629    7688    7725    7794    8182
                    8246    8310    8374    8438    8502    8566    8625    8673
SWIFLG  002250       2121#   6704    6743    6782
SWPAC1  037030       8852    8889#
SWPAC2  037104       8858    8897#
SWPAC3  037141       8864    8902#
SW0    = 000002      2326#
SW1    = 000004      2325#
SW2    = 000010      2324#
SW3    = 000040      2322#
SYNCH  = 000226      2433#   5981    6013    6036    6064    6615    6875    6996    7426    7502    7577    7637    7692
                    7729    8254    8446    8679
SYN0   = 000001      2432#
SYN1   = 000002      2431#
SYN2   = 000004      2430#
SYN3   = 000010      2429#
SYN4   = 000020      2428#
SYN5   = 000040      2427#
SYN6   = 000100      2426#
SYN7   = 000200      2425#
SSLSYM= 010000       1894#   2102#   2126#   4709#   4782#   4892#   5001#   5119#   5228#   5330#   5447#   5549#   5568#
                    5747    5752    5753#   5759#   5826#   5845#   5873#   5898#   5953#   6084#   6169#   6203#   6239#
                    6277#   6406#   6464#   6500#   6507#   6558#   6589#   6592#   6646#   6720#   6759#   6798#   6801#
                    6844#   6964#   7100#   7188#   7250#   7254#   7340#   7397#   7401#   7476#   7549#   7609#   7669#
                    7709#   7761#   7867#   7905#   7931#   7956#   7982#   8007#   8062#   8076#   8101#   8153#   8159#
                    8223#   8287#   8351#   8415#   8479#   8543#   8607#   8653#   8822#   8884#   9002#
TCCHEK= 100000       2526#   4092    7800
TCOUNT  002252       2122#   7434    7526    7586    7646    8200    8264    8328    8392    8456    8519    8583
TEOM   = 000002      2419#   6354    6360
TERR   = 000200      2416#
TEST   = 000001      2457#   4091    7424    7500    7575    7635
TESTMD= 000004       2373#   4291    4307    4317    7970
TIMCNT  040046       8994    9023#
TIMFLG  002332       2579#
TMP0    002502       2644#   3093*   3096*   4805    5032    5149    5470
TMP1    002504       2645#   3111*   3112*   4806    5033    5150    5471
TMP2    002506       2646#
TMP3    002510       2647#
TMP4    002512       2648#
TMP5    002514       2649#
TMP6    002516       2650#
TMP7    002520       2651#
TSOM   = 000001      2420#   6317    7743
TSTCON  002442       2618#   4094    4098    4119    4158    4170    4189    4205    4281    4287    4351    5679*   5680
                    5682*   5684*   5687    5691    5694    5698
TSTMLB  012412       4351#   7583    7643    8197    8261    8325    8389    8453    8516    8580
TSTNUM  002422       2607#   4133    7419*   7495*   7570*   7630*   7795*   8183*   8247*   8311*   8375*   8439*   8503*
                    8567*
TXAB   = 000004      2418#
TXABT  = 002000      2478#   2796
TXCHAR  005610       3398#   6998    7033    7068
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- USER SYMBOLS

```
TXDATA= 000040   2370#  7159  7171  7221  7233  7859  7861
TXEN  = 000100   2285#  2357#  6224  6309  6311  7738  7740
TXEOM = 001000   2479#  2780  2781  2782  2783  2789  2790  3922  5933  6451  6487  6545  6576
                 7143  7205  7307  7360  8634  8685  8691
TXGA  = 000010   2417#
TXGOA = 004000   2477#
TXLEN0= 000040   2465#  3745  6399  7146  7208  7306  7363  8513  8577
TXLEN1= 000100   2464#  3745  6301  7146  7208  7306  7363  8513  8577
TXLEN2= 000200   2463#  3745  7146  7208  7306  7363  8513  8577
TXSOM = 000400   2480#  2773  2774  2785  2786  2797  2798  3336  3908  4026  6013  7297  7354
                 8694
TXWORD  002400   2598#  3224*  3226  3229  3336*  3337*  3402*  3913*  3914*  3915  3922*  3979*  4004*
                 4005*  4053*  5933*
TX0   = 000001   2253#  2411#
TX1   = 000002   2252#  2410#
TX2   = 000004   2251#  2409#
TX3   = 000010   2250#  2408#
TX4   = 000020   2249#  2407#
TX5   = 000040   2248#  2406#
TX6   = 000100   2247#  2405#
TX7   = 000200   2246#  2404#
TYPEY   023206   5750  5785#
T$ARGC= 000002   1932#  1933#  1934#  1935#  1936#  1937#  4133#  4138  4148#  4152  4310#  4314  4326#
                 4330  4700#  4706  4716#  4722  4724#  4728  4730#  4736  4738#  4744  4746#  4752
                 4754#  4762  4764#  4769  4771#  4779  4791#  4797  4799#  4803  4805#  4811  4813#
                 4819  4821#  4827  4829#  4837  4839#  4844  4846#  4854  4856#  4862  4864#  4872
                 4874#  4879  4881#  4889  4901#  4906  4908#  4914  4916#  4920  4922#  4928  4930#
                 4936  4938#  4946  4948#  4953  4955#  4963  4965#  4971  4973#  4981  4983#  4988
                 4990#  4998  5010#  5016  5018#  5024  5026#  5030  5032#  5038  5040#  5046  5048#
                 5054  5056#  5064  5066#  5071  5073#  5081  5083#  5089  5091#  5099  5101#  5106
                 5108#  5116  5128#  5133  5135#  5141  5143#  5147  5149#  5155  5157#  5163  5165#
                 5173  5175#  5180  5182#  5190  5192#  5198  5200#  5208  5210#  5215  5217#  5225
                 5237#  5243  5245#  5249  5251#  5257  5259#  5265  5267#  5275  5277#  5282  5284#
                 5292  5294#  5300  5302#  5310  5312#  5317  5319#  5327  5339#  5344  5346#  5352
                 5354#  5358  5360#  5366  5368#  5374  5376#  5382  5384#  5392  5394#  5399  5401#
                 5409  5411#  5417  5419#  5427  5429#  5434  5436#  5444  5456#  5462  5464#  5468
                 5470#  5476  5478#  5484  5486#  5494  5496#  5501  5503#  5511  5513#  5519  5521#
                 5529  5531#  5536  5538#  5546  5703#  5707  5725#  5729  5736#  5741  5865#  5870
                 6690#  6695  6698#  6703  6737#  6742  6776#  6781  6827#  6832  6835#  6840
T$CODE= 003032   5749#  8846#  8851#  8857#  8863#  8869#  8875#  8981#  8985#  8989#  8993#
T$ERRN= 000101   1894#  3143#  3153#  3167#  3177#  3288#  3298#  3460#  3470#  3484#  3494#  3533#  3543#
                 3581#  3591#  3603#  3613#  3763#  3772#  3795#  3805#  3818#  3828#  3841#  3851#  3864#
                 3874#  4253#  4263#  4300#  4369#  6027#  6078#  6163#  6333#  6376#  6393#  6714#  6753#
                 6792#  7165#  7177#  7227#  7239#  7444#  7470#  7517#  7536#  7596#  7656#  7820#  7836#
                 7896#  7925#  7950#  7976#  8001#  8033#  8045#  8056#  8095#  8123#  8135#  8147#  8210#
                 8274#  8338#  8402#  8466#  8529#  8593#  8709#  8725#  8791#
T$EXCP= 000000   8846#  8850  8851#  8856  8857#  8862  8863#  8868  8869#  8874  8875#  8880  8993#
                 8998
T$FLAG= 000040   7169#  7181#  7231#  7243#  7824#  7840#  7900#  8037#  8127#  8139#
T$GMAN= 000000   1894#
T$HILI= 177777   8846#  8849  8851#  8855  8857#  8861  8863#  8867  8869#  8873  8875#  8879  8993#
                 8997
T$LAST= 000001   1894#  9056#
T$LOLI= 000000   8846#  8848  8851#  8854  8857#  8860  8863#  8866  8869#  8872  8875#  8878  8993#
                 8996
T$LSYM= 010000   1894#  2102  2126  4709  4782  4892  5001  5119  5228  5330  5447  5549  5568
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    5759    5826    5845    5873    5898    5953    6084    6169    6203    6239    6277    6406    6464
                    6500    6507    6558    6589    6592    6646    6720    6759    6798    6801    6844    6964    7100
                    7188    7250    7254    7340    7397    7401    7476    7549    7609    7669    7709    7761    7867
                    7905    7931    7956    7982    8007    8062    8076    8101    8153    8159    8223    8287    8351
                    8415    8479    8543    8607    8653    8822    8884    9002
T$LTNO= 000040      9059#
T$NEST= 177777      1894#   1900#   2089#   2101#   2115#   2125#   4698#   4708#   4714#   4781#   4789#   4891#   4899#
                    5000#   5008#   5118#   5126#   5227#   5235#   5329#   5337#   5446#   5454#   5548#   5564#   5567#
                    5584#   5589#   5602#   5758#   5808#   5825#   5840#   5844#   5859#   5872#   5895#   5897#   5926#
                    5952#   5976#   6083#   6109#   6168#   6185#   6202#   6220#   6238#   6256#   6276#   6297#   6405#
                    6434#   6439#   6463#   6472#   6499#   6506#   6528#   6533#   6557#   6564#   6588#   6591#   6613#
                    6645#   6673#   6678#   6719#   6726#   6758#   6765#   6797#   6800#   6816#   6843#   6869#   6963#
                    6991#   7099#   7127#   7134#   7187#   7196#   7249#   7253#   7284#   7291#   7339#   7348#   7396#
                    7400#   7419#   7475#   7495#   7548#   7570#   7608#   7630#   7668#   7689#   7708#   7726#   7760#
                    7795#   7809#   7866#   7883#   7904#   7912#   7930#   7937#   7955#   7962#   7981#   7989#   8006#
                    8019#   806 #   8069#   8075#   8083#   8100#   8111#   8152#   8158#   8183#   8222#   8247#   8286#
                    8311#   8350#   8375#   8414#   8439#   8478#   8503#   8542#   8567#   8606#   8626#   8652#   8674#
                    8821#   8842#   8882#   8977#   9000#   9053#
T$NS0 = 000000      1900#   9053
T$NS1 = 000005      2089#   2101    2115#   2125    4698#   4708    4714#   4781    4789#   4891    4899#   5000    5008#
                    5118    5126#   5227    5235#   5329    5337#   5446    5454#   5548    5564#   5567    5584#   5589
                    5602#   5758    5808#   5825    5840#   5844    5859#   5872    5895#   5897    5926#   5952    5976#
                    6083    6109#   6168    6185#   6202    6220#   6238    6256#   6276    6297#   6405    6434#   6506
                    6528#   6591    6613#   6645    6673#   6800    6816#   6843    6869#   6963    6991#   7099    7127#
                    7253    7284#   7400    7419#   7475    7495#   7548    7570#   7608    7630#   7668    7689#   7708
                    7726#   7760    7795#   8158    8183#   8222    8247#   8286    8311#   8350    8375#   8414    8439#
                    8478    8503#   8542    8567#   8606    8626#   8652    8674#   8821    8842#   8882    8977#   9000
T$NS2 = 000002      6439#   6463    6472#   6499    6533#   6557    6564#   6588    6678#   6719    6726#   6758    6765#
                    6797    7134#   7187    7196#   7249    7291#   7339    7348#   7396    7808#   7866    7883#   7904
                    7912#   7930    7937#   7955    7962#   7981    7989#   8006    8019#   8061    8069#   8075    8083#
                    8100    8111#   8152
T$PTNU= 000000      1894#
T$SAVL= 177777      1894#
T$SEGL= 177777      1894#
T$SUBN= 000000      1894#   5925#   5975#   6108#   6184#   6219#   6255#   6296#   6433#   6438#   6471#   6527#   6532#
                    6563#   6612#   6672#   6677#   6725#   6764#   6815#   6868#   6990#   7126#   7133#   7195#   7283#
                    7290#   7347#   7418#   7494#   7569#   7629#   7688#   7725#   7794#   7807#   7882#   7911#   7936#
                    7961#   7988#   8018#   8068#   8082#   8110#   8182#   8246#   8310#   8374#   8438#   8502#   8566#
                    8625#   8673#
T$TAGL= 177777      1894#
T$TAGN= 010111      1894#   2089#   2115#   4698#   4714#   4789#   4899#   5008#   5126#   5235#   5337#   5454#   5564#
                    5584#   5602#   5808#   5840#   5859#   5895#   5926#   5976#   6109#   6185#   6220#   6256#   6297#
                    6434#   6439#   6472#   6528#   6533#   6564#   6613#   6673#   6678#   6726#   6765#   6816#   6869#
                    6991#   7127#   7134#   7196#   7284#   7291#   7348#   7419#   7495#   7570#   7630#   7689#   7726#
                    7795#   7808#   7883#   7912#   7937#   7962#   7989#   8019#   8069#   8083#   8111#   8183#   8247#
                    8311#   8375#   8439#   8503#   8567#   8626#   8674#   8842#   8977#
T$TEMP= 000000      2038#   2039#   2040#   2041#   2042#   2043#   2044#   2045#   2046#   2047#   2048#   2049#   2050#
                    2051#   2052#   2053#   2054#   2055#   2056#   2057#   2058#   2059#   2060#   2061#   2062#   2063#
                    2064#   2065#   2066#   2067#   2068#   2069#   2070#   2101#   2125#   4708#   4781#   4891#   5000#
                    5118#   5227#   5329#   5446#   5548#   5567#   5589#   5749#   5758#   5825#   5844#   5872#   5897#
                    5952#   6083#   6168#   6202#   6238#   6276#   6405#   6463#   6499#   6506#   6557#   6588#   6591#
                    6645#   6719#   6758#   6797#   6800#   6843#   6963#   7099#   7169#   7170    7181#   7182    7187#
                    7231#   7232    7243#   7244    7249#   7253#   7339#   7396#   7400#   7475#   7548#   7608#   7668#
                    7708#   7760#   7824#   7825    7840#   7841    7866#   7900#   7901    7904#   7930#   7955#   7981#
                    8006#   8037#   8038    8061#   8075#   8100#   8127#   8128    8139#   8140    8152#   8158#   8222#
                    8286#   8350#   8414#   8478#   8542#   8606#   8652#   8821#   8846#   8851#   8857#   8863#   8869#
```

```
                    8875#  8882#  8981#  8985#  8989#  8993#  9000#  9053#
T$TEST= 000040      1894#  5925#  5975#  6108#  6184#  6219#  6255#  6296#  6433#  6438   6471   6527#  6532
                    6563   6612#  6672#  6677   6725   6764   6815#  6868#  6990#  7126#  7133   7195   7283#
                    7290   7347   7418#  7494#  7569#  7629#  7688#  7725#  7794#  7807   7882   7911   7936
                    7961   7988   8018   8068   8082   8110   8182#  8246#  8310#  8374#  8438#  8502#  8566#
                    8625#  8673#  9059
T$TSTM= 177777      1894#  3142   3152   3166   3176   3287   3297   3459   3469   3483   3493   3532   3542
                    3580   3590   3602   3612   3762   3771   3794   3804   3817   3827   3840   3850   3863
                    3873   4137   4151   4252   4262   4299   4313   4320   4329   4368   4705   4709   4721
                    4727   4735   4743   4751   4761   4768   4778   4782   4796   4802   4810   4818   4826
                    4836   4843   4853   4861   4871   4878   4888   4892   4905   4913   4919   4927   4935
                    4945   4952   4962   4970   4980   4987   4997   5001   5015   5023   5029   5037   5045
                    5053   5063   5070   5080   5088   5098   5105   5115   5119   5132   5140   5146   5154
                    5162   5172   5179   5189   5197   5207   5214   5224   5228   5242   5248   5256   5264
                    5274   5281   5291   5299   5309   5316   5326   5330   5343   5351   5357   5365   5373
                    5381   5391   5398   5408   5416   5426   5433   5443   5447   5461   5467   5475   5483
                    5493   5500   5510   5518   5528   5535   5545   5549   5568   5621   5627   5633   5639
                    5661   5706   5719   5728   5731   5740   5746   5759   5812   5821   5826   5845   5862
                    5869   5873   5898   5953   6026   6077   6084   6162   6169   6203   6239   6277   6332
                    6375   6392   6406   6439   6464   6472   6500   6507   6533   6558   6564   6589   6592
                    6646   6678   6694   6702   6713   6720   6726   6741   6752   6759   6785   6780   6791
                    6798   6801   6831   6839   6844   6964   7100   7134   7164   7169   7176   7181   7188
                    7196   7226   7231   7238   7243   7250   7254   7291   7340   7348   7397   7401   7443
                    7469   7476   7516   7535   7549   7595   7609   7655   7669   7709   7761   7808   7819
                    7824   7835   7840   7867   7883   7895   7900   7905   7912   7924   7931   7937   7949
                    7956   7962   7975   7982   7989   8000   8007   8019   8032   8037   8044   8055   8062
                    8069   8076   8083   8094   8101   8111   8122   8127   8134   8139   8146   8153   8159
                    8209   8223   8273   8287   8337   8351   8401   8415   8465   8479   8528   8543   8592
                    8607   8653   8708   8724   8790   8822
T$TSTS= 000001      1894#  5926#  5976#  6109#  6185#  6220#  6256#  6297#  6434#  6528#  6613#  6673#  6816#
                    6869#  6991#  7127#  7284#  7419#  7495#  7570#  7630#  7689#  7726#  7795#  8183#  8247#
                    8311#  8375#  8439#  8503#  8567#  8626#  8674#
T$$AU = 010021      5895#  5897
T$$AUT= 010016      5808#  5825
T$$CLE= 010017      5840#  5844
T$$DU = 010020      5859#  5872
T$$HAR= 010107      8842#  8883
T$$HW = 010000      2089#  2101
T$$INI= 010015      5602#  5758
T$$MSG= 010012      4698#  4708   4714#  4781   4789#  4891   4899#  5000   5008#  5118   5126#  5227   5235#
                    5329   5337#  5446   5454#  5548
T$$PRO= 010014      5584#
T$$RPT= 010013      5564#  5567
T$$SOF= 010110      8977#  9001
T$$SUB= 010075      6439#  6463   6472#  6499   6533#  6557   6564#  6588   6678#  6719   6726#  6758   6765#
                    6797   7134#  7169   7181   7187   7196#  7231   7253   7249   7291#  7339   7348#  7396
                    7808#  7824   7840   7866   7883#  7900   7904   7912#  7930   7937#  7955   7962#  7981
                    7989#  8006   8019#  8037   8061   8069#  8075   8083#  8100   8111#  8127   8139   8152
T$$SW = 010001      2115#  2125
T$$TES= 010106      5926#  5952   5976#  6083   6109#  6168   6185#  6202   6220#  6238   6256#  6276   6297#
                    6405   6434#  6506   6528#  6591   6613#  6645   6673#  6800   6816#  6843   6869#  6963
                    6991#  7099   7127#  7253   7284#  7400   7419#  7475   7495#  7548   7570#  7608   7630#
                    7668   7689#  7708   7726#  7760   7795#  8158   8183#  8222   8247#  8286   8311#  8350
                    8375#  8414   8439#  8478   8503#  8542   8567#  8606   8626#  8652   8674#  8821
T1      023436 G.   2038   5925#
T10     026150 G.   2047   6612#
```

T11	026304	G	2048	6672#								
T11.1	026304		6677#									
T11.2	026470		6725#									
T11.3	026622		6764#									
T12	026756	G	2049	6815#								
T13	027072	G	2050	6868#								
T14	027430	G	2051	6990#								
T15	030040	G	2052	7126#								
T15.1	030054		7133#									
T15.2	030276		7195#									
T16	030512	G	2053	7283#								
T16.1	030526		7290#									
T16.2	030764		7347#									
T17	031214	G	2054	7418#								
T18	031476	G	2055	7494#								
T19	031750	G	2056	7569#								
T2	023566	G	2039	5975#								
T20	032134	G	2057	7629#								
T21	032320	G	2058	7688#								
T22	032424	G	2059	7725#								
T23	032642	G	2060	7794#								
T23.1	032672		7807#									
T23.10	034132		8110#									
T23.2	033226		7882#									
T23.3	033320		7911#									
T23.4	033406		7936#									
T23.5	033474		7961#									
T23.6	033570		7988#									
T23.7	033650		8018#									
T23.8	034022		8068#									
T23.9	034052		8082#									
T24	034272	G	2061	8182#								
T25	034464	G	2062	8246#								
T26	034656	G	2063	8310#								
T27	035050	G	2064	8374#								
T28	035242	G	2065	8438#								
T29	035434	G	2066	8502#								
T3	024224	G	2040	6108#								
T30	035622	G	2067	8566#								
T31	036010	G	2068	8625#								
T32	036122	G	2069	8673#								
T4	024456	G	2041	6184#								
T5	024540	G	2042	6219#								
T6	024630	G	2043	6255#								
T7	024734	G	2044	6296#								
T8	025504	G	2045	6433#								
T8.1	025504		6438#									
T8.2	025612		6471#									
T9	025744	G	2046	6527#								
T9.1	025744		6532#									
T9.2	026046		6563#									
UAM =	000200	G	2211#									
UNIT	002420		2606#									
UNRR =	000001		2327#	3757								
UPBITS	002522		2654#									
V35 =	000020		2453#	2458	4091	4108	4161	4163	4165	4199	7424	7499	7575	7635

```
WAIT50  004634          3196#  3346   3404   3676   3983   4009   4059   4293   4321   4362   5709   7454
WAX   = 000010          2288#  2360#  3069
WAX15   002346          2585#  3059*  3060   3335*  3337   3942*  3951*  6316*  7145*  7207*  7305*  7362*  7742*
                        7747*
WAX16   002350          2586#  3063*  3064   3338*  3943*  3953*  3954   6317*  6340*  6348*  6351*  6354*  6360*
                        7146*  7208*  7306*  7307   7363*  7364   7743*  7748*
WRDYTO= 000002          2550#  3052   3078
WRIBYT  002340          2582#  2905*  2953*  2954   3016*  3017*  3018*  3019*  3054*  3055*  3056*  3060*  3064*
                        3067*  3068*  3069*  3070*  3226*  3229*  3342*  3712*  3713*  3719*  3947*  4160*  4191*
                        4213*  4215*  4219*  6008*  6010*  6152*  6268*  6309*  6313*  7694*  7695*  7734*  7735*
                        7738*  7842*  7914*  7939*  7964*  7991*  8022*  8072*  8085*  8779*
WRITAX  004000          3050#  3339   3944   3956   6318   6337   6341   6344   6349   6355   6361   7147   7209
                        7308   7365   7744   7749   7752
WRITLU  003436          2906   2947#  3020   3057   3061   3065   3071   3227   3230   3343   3714   3720   3948
                        4220   6009   6011   6153   6269   6310   6314   7696   7736   7739   7843   7916   7941
                        7966   7993   8023   8073   8087   8780
WTMSG   023236          5703   5789#
XYZ   = 000100          2451#  2458   4091   4113   4182   4188   4208   7424   7500   7574   7635
X$ALWA= 000000          1894#
X$FALS= 000040          1894#
X$OFFS= 000400          1894#
X$TRUE= 000020          1894#
$LSTIN= 000001          1902#
$LSTTA= 000001          1903#
.     = 040244          1885#  2566#  2765#  2810#  2849#  5880#  7170   7182   7232   7244   7825   7841   7901
                        8038   8128   8140   8957#  9043#
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- MACRO NAMES

```
BCOMPL      1#  1894#  5622  5628  5634  5663  5721
BERROR      1#  1894#
BGNAU       1#  1894#  5894
BGNAUT      1#  1894#  5807
BGNCLN      1#  1894#  5839
BGNDU       1#  1894#  5858
BGNHRD      1#  1894#  8841
BGNHW       1#  1894#  2088
BGNINI      1#  1894#  5601
BGNMOD      1#  1894#  1899
BGNMSG      1#  1894#  4697  4713  4788  4898  5007  5125  5234  5336  5453
BGNPRO      1#  1894#  5583
BGNPTA      1#  1894#
BGNRPT      1#  1894#  5563
BGNSEG      1#  1894#
BGNSET      1#  1894#
BGNSFT      1#  1894#  8976
BGNSRV      1#  1894#
BGNSUB      1#  1894#  6437  6470  6531  6562  6676  6724  6763  7132  7194  7289  7346  7806  7881
         7910   7935   7960  7987  8017  8067  8081  8109
BGNSW       1#  1894#  2114
BGNTST      1#  1894#  5924  5974  6107  6183  6218  6254  6295  6432  6526  6611  6671  6814  6867
         6989   7125   7282  7417  7493  7568  7628  7687  7724  7793  8181  8245  8309  8373  8437
         8501   8565   8624  8672
BNCOMP      1#  1894#  5640
BNERRO      1#  1894#
BREAK       1#  1894#  4319  5730
BRESET      1#  1894#  5861
CKLOOP      1#  1894#
CLOCK       1#  1894#
CLOSE       1#  1894#
CLRVEC      1#  1894#
COMMEN      1#  1894#
DELAY       1#  1894#
DESCRI      1#  1894#  2840
DEVTYP      1#  1894#  2832
DISPAT      1#  1894#  2035
DISPLA      1#  1894#
DOCLN       1#  1894#
DODU        1#  1894#  5819
DORPT       1#  1894#
ENDAU       1#  1894#  5896
ENDAUT      1#  1894#  5824
ENDCLN      1#  1894#  5843
ENDCOM      1#  1894#
ENDDU       1#  1894#  5871
ENDHRD      1#  1894#  8881
ENDHW       1#  1894#  2100
ENDINI      1#  1894#  5757
ENDMOD      1#  1894#  9052
ENDMSG      1#  1894#  4707  4780  4890  4999  5117  5226  5328  5445  5547
ENDPRO      1#  1894#  5588
ENDPTA      1#  1894#
ENDRPT      1#  1894#  5566
ENDSEG      1#  1894#
ENDSET      1#  1894#
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- MACRO NAMES

```
ENDSFT      1#    1894#   8999
ENDSRV      1#    1894#
ENDSUB      1#    1894#   6462    6498    6556    6587    6718    6757    6796    7186    7248    7338    7395    7865    7903
         7929     7954    7980    8005    8060    8074    8099    8151
ENDSW       1#    1894#   2124
ENDTST      1#    1894#   5951    6082    6167    6201    6237    6275    6404    6505    6590    6644    6799    6842    6962
         7098     7252    7399    7474    7547    7607    7667    7707    7759    8157    8221    8285    8349    8413    8477
         8541     8605    8651    8820
EQUALS      1#    1894#   2151
ERRDF       1#    1894#   3141    3151    3165    3175    3286    3296    3458    3468    3482    3492    3531    3541    3579
         3589     3601    3611    3761    3770    3793    3803    3816    3826    3839    3849    3862    3872    4251    4261
         4298     4367    6025    6076    6161    6331    6374    6391    6712    6751    6790    7163    7175    7225    7237
         7442     7468    7515    7534    7594    7654    7818    7834    7894    7923    7948    7974    7999    8031    8043
         8054     8093    8121    8133    8145    8208    8272    8336    8400    8464    8527    8591    8707    8723    8789
ERRHRD      1#    1894#
ERROR       1#    1894#
ERRSF       1#    1894#
ERRSOF      1#    1894#
ERRTBL      1#    1894#
ESCAPE      1#    1894#   7168    7180    7230    7242    7823    7839    7899    8036    8126    8138
EXIT        1#    1894#
FEQUAL      1#    1894#
GETBYT      1#    1894#
GETPRI      1#    1894#
GETWOR      1#    1894#
GMANIA      1#    1894#
GMANID      1#    1894#
GMANIL      1#    1894#   5745
GPHARD      1#    1894#   5659
GPRMA       1#    1894#   8845
GPRMD       1#    1894#   8850    8856    8862    8868    8874    8992
GPRML       1#    1894#   5746#   5749    8980    8984    8988
HEADER      1#    1894#   1930
INLOOP      1#    1894#
IOSETU      1#    1894#
IOSTAR      1#    1894#
KT11        1#    1894#
LASTAD      1#    1894#   9054
MANUAL      1#    1894#   5718
MEMORY      1#    1894#
MSBYTE      1#    1894#   1931#   1937    1938    1939
MSCHEC      1#    1894#
MSCNTO      1#    1894#   5749#   8846#   8851#   8857#   8863#   8869#   8875#   8981#   8985#   8989#   8993#
MSCOUN      1#    1894#   4133#   4148#   4310#   4326#   4700#   4716#   4724#   4730#   4738#   4746#   4754#   4764#   4771#
         4791#    4799#   4805#   4813#   4821#   4829#   4839#   4846#   4856#   4864#   4874#   4881#   4901#   4908#   4916#
         4922#    4930#   4938#   4948#   4955#   4965#   4973#   4983#   4990#   5010#   5018#   5026#   5032#   5040#   5048#
         5056#    5066#   5073#   5083#   5091#   5101#   5108#   5128#   5135#   5143#   5149#   5157#   5165#   5175#   5182#
         5192#    5200#   5210#   5217#   5237#   5245#   5251#   5259#   5267#   5277#   5284#   5294#   5302#   5312#   5319#
         5339#    5346#   5354#   5360#   5368#   5376#   5384#   5394#   5401#   5411#   5419#   5429#   5436#   5456#   5464#
         5470#    5478#   5486#   5496#   5503#   5513#   5521#   5531#   5538#   5703#   5725#   5736#   5865#   6690#   6698#
         6737#    6776#   6827#   6835#
MSDATA      1#    1894#   1931#   1940    1942    1944    1946    1948    1950    1952    1954    1956    1958    1960    1962
         1964     1966    1968    1970#   1972    1974    1977    1980    1982    1984    1986    1988    1990    1992    1994
         1996     1998    2000    2002    2004    2006    2008    2010    2012    2014    2833#   2841#
MSDECR      1#    1894#   2101#   2125#   4708#   4781#   4891#   5000#   5118#   5227#   5329#   5446#   5548#   5567#   5589#
         5758#    5825#   5844#   5872#   5897#   5952#   6083#   6168#   6202#   6238#   6276#   6405#   6463#   6499#   6506#
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- MACRO NAMES

```
            6557#  6588#  6591#  6645#  6719#  6758#  6797#  6800#  6843#  6963#  7099#  7187#  7249#  7253#  7339#
            7396#  7400#  7475#  7548#  7608#  7668#  7708#  7760#  7866#  7904#  7930#  7955#  7981#  8006#  8061#
            8075#  8100#  8152#  8158#  8222#  8286#  8350#  8414#  8478#  8542#  8606#  8652#  8821#  8882#  9000#
            9053#
M$DEFA         1#  1894#  5749#  8846#  8851#  8857#  8863#  8869#  8875#  8981#  8985#  8989#  8993#
M$ENDE         1#  1894#  2101#  2125#  4708#  4781#  4891#  5000#  5118#  5227#  5329#  5446#  5548#  5567#  5758#
            5825#  5844#  5872#  5897#  5952#  6083#  6168#  6202#  6238#  6276#  6405#  6463#  6499#  6506#  6557#
            6588#  6591#  6645#  6719#  6758#  6797#  6800#  6843#  6963#  7099#  7187#  7249#  7253#  7339#  7396#
            7400#  7475#  7548#  7608#  7668#  7708#  7760#  7866#  7904#  7930#  7955#  7981#  8006#  8061#  8075#
            8100#  8152#  8158#  8222#  8286#  8350#  8414#  8478#  8542#  8606#  8652#  8821#  8882#  9000#  9053#
M$ERRI         1#  1894#  3142#  3152#  3166#  3176#  3287#  3297#  3459#  3469#  3483#  3493#  3532#  3542#  3580#
            3590#  3602#  3612#  3762#  3771#  3794#  3804#  3817#  3827#  3840#  3850#  3863#  3873#  4252#  4262#
            4299#  4368#  6026#  6077#  6162#  6332#  6375#  6392#  6713#  6752#  6791#  7164#  7176#  7226#  7238#
            7443#  7469#  7516#  7535#  7595#  7655#  7819#  7835#  7895#  7924#  7949#  7975#  8000#  8032#  8044#
            8055#  8094#  8122#  8134#  8146#  8209#  8273#  8337#  8401#  8465#  8528#  8592#  8708#  8724#  8790#
M$ESCA         1#  1894#  7169#  7170   7181#  7182   7231#  7232   7243#  7244   7824#  7825   7840#  7841   7900#
            7901   8037#  8038   8127#  8128   8139#  8140
M$ESCS         1#  1894#  7169#  7181#  7231#  7243#  7824#  7840#  7900#  8037#  8127#  8139#
M$EXCP         1#  1894#  8846#  8851#  8857#  8863#  8869#  8875#  8993#
M$EXIT         1#  1894#
M$EXSE         1#  1894#
M$EXTJ         1#  1894#
M$GEN          1#  1894#  1900#  1931#  1940#  1942#  1944#  1946#  1948#  1950#  1952#  1954#  1956#  1958#  1960#
            1962#  1964#  1966#  1968#  1970#  1972#  1974#  1977#  1980#  1982#  1984#  1986#  1988#  1990#  1992#
            1994#  1996#  1998#  2000#  2002#  2004#  2006#  2008#  2010#  2012#  2014#  2037#  2090#  2091#  2101#
            2116#  2117#  2125#  2833#  2841#  4698#  4708#  4714#  4781#  4789#  4891#  4899#  5000#  5008#  5118#
            5126#  5227#  5235#  5329#  5337#  5446#  5454#  5548#  5564#  5567#  5584#  5602#  5752#  5758#  5808#
            5825#  5840#  5844#  5859#  5872#  5895#  5897#  5925#  5952#  5975#  6083#  6108#  6168#  6184#  6202#
            6219#  6238#  6255#  6276#  6296#  6405#  6433#  6438#  6463#  6471#  6499#  6506#  6527#  6532#  6557#
            6563#  6588#  6591#  6612#  6645#  6672#  6677#  6719#  6725#  6758#  6764#  6797#  6800#  6815#  6843#
            6868#  6963#  6990#  7099#  7126#  7133#  7187#  7195#  7249#  7253#  7283#  7290#  7339#  7347#  7396#
            7400#  7418#  7475#  7494#  7548#  7569#  7608#  7629#  7668#  7688#  7708#  7725#  7760#  7794#  7807#
            7866#  7882#  7904#  7911#  7930#  7936#  7955#  7961#  7981#  7988#  8006#  8018#  8061#  8068#  8075#
            8082#  8100#  8110#  8152#  8158#  8182#  8222#  8246#  8286#  8310#  8350#  8374#  8414#  8438#  8478#
            8502#  8542#  8566#  8606#  8625#  8652#  8673#  8821#  8843#  8883#  8978#  9001#  9058#
M$GENB         1#  1894#  5746#  5747
M$GETS         1#  1894#  2101#  2125#  4708#  4781#  4891#  5000#  5118#  5227#  5329#  5446#  5548#  5567#  5589#
            5758#  5825#  5844#  5872#  5897#  5952#  6083#  6168#  6202#  6238#  6276#  6405#  6463#  6499#  6506#
            6557#  6588#  6591#  6645#  6719#  6758#  6797#  6800#  6843#  6963#  7099#  7187#  7249#  7253#  7339#
            7396#  7400#  7475#  7548#  7608#  7668#  7708#  7760#  7866#  7904#  7930#  7955#  7981#  8006#  8061#
            8075#  8100#  8152#  8158#  8222#  8286#  8350#  8414#  8478#  8542#  8606#  8652#  8821#  8882#  9000#
            9053#
M$GETT         1#  1894#  7169#  7181#  7231#  7243#  7824#  7840#  7900#  8037#  8127#  8139#
M$GNGB         1#  1894#  1900#  1931#  1940#  1942#  1944#  1946#  1948#  1950#  1952#  1954#  1956#  1958#  1960#
            1962#  1964#  1966#  1968#  1970#  1972#  1974#  1977#  1980#  1982#  1984#  1986#  1988#  1990#  1992#
            1994#  1996#  1998#  2000#  2002#  2004#  2006#  2008#  2010#  2012#  2014#  2036#  2037   2089#  2090
            2091   2115#  2116   2117   2833#  2841#  4698#  4714#  4789#  4899#  5008#  5126#  5235#  5337#  5454#
            5564#  5584#  5602#  5808#  5840#  5859#  5895#  8842#  8843   8977#  8978   9055#  9058
M$GNIN         1#  1894#  1931#  1932   1933   1934   1935   1936   1937#  1938#  1939#  1940#  1941   1942#  1943
            1944#  1945   1946#  1947   1948#  1949   1950#  1951   1952#  1953   1954#  1955   1956#  1957   1958#
            1959   1960#  1961   1962#  1963   1964#  1965   1966#  1967   1968#  1969   1970#  1971   1972#  1973
            1974#  1975   1976   1977#  1978   1979#  1980#  1981   1982#  1983   1984#  1985   1986#  1987   1988#
            1989   1990#  1991   1992#  1993   1994#  1995   1996#  1997   1998#  1999   2000#  2001   2002#  2003
            2004#  2005   2006#  2007   2008#  2009   2010#  2011   2012#  2013   2014#  2015   2036#  2038#  2039#
            2040#  2041#  2042#  2043#  2044#  2045#  2046#  2047#  2048#  2049#  2050#  2051#  2052#  2053#  2054#
            2055#  2056#  2057#  2058#  2059#  2060#  2061#  2062#  2063#  2064#  2065#  2066#  2067#  2068#  2069#
```

2089# 2115# 2833# 2834 2835 2841# 2842 2849 3142# 3143# 3144# 3145# 3152# 3153# 3154#
3155# 3166# 3167# 3168# 3169# 3176# 3177# 3178# 3179# 3287# 3288# 3289# 3290# 3297# 3298#
3299# 3300# 3459# 3460# 3461# 3462# 3469# 3470# 3471# 3472# 3483# 3484# 3485# 3486# 3493#
3494# 3495# 3496# 3532# 3533# 3534# 3535# 3542# 3543# 3544# 3545# 3580# 3581# 3582# 3583#
3590# 3591# 3592# 3593# 3602# 3603# 3604# 3605# 3612# 3613# 3614# 3615# 3762# 3763# 3764#
3765# 3771# 3772# 3773# 3774# 3794# 3795# 3796# 3797# 3804# 3805# 3806# 3807# 3817# 3818#
3819# 3820# 3827# 3828# 3829# 3830# 3840# 3841# 3842# 3843# 3850# 3851# 3852# 3853# 3863#
3864# 3865# 3866# 3873# 3874# 3875# 3876# 4133# 4134# 4135# 4136 4137# 4138 4148# 4149#
4150 4151# 4152 4252# 4253# 4254# 4255# 4262# 4263# 4264# 4265# 4299# 4300# 4301# 4302#
4310# 4311# 4312 4313# 4314 4320# 4326# 4327# 4328 4329# 4330 4368# 4369# 4370# 4371#
4700# 4701# 4702# 4703# 4704 4705# 4706 4709# 4716# 4717# 4718# 4719# 4720 4721# 4722
4724# 4725# 4726 4727# 4728 4730# 4731# 4732# 4733# 4734 4735# 4736 4738# 4739# 4740#
4741# 4742 4743# 4744 4746# 4747# 4748# 4749# 4750 4751# 4752 4754# 4755# 4756# 4757#
4758# 4759# 4760 4761# 4762 4764# 4765# 4766# 4767 4768# 4769 4771# 4772# 4773# 4774#
4775# 4776# 4777 4778# 4779 4782# 4791# 4792# 4793# 4794# 4795 4796# 4797 4799# 4800#
4801 4802# 4803 4805# 4806# 4807# 4808# 4809 4810# 4811 4813# 4814# 4815# 4816# 4817
4818# 4819 4821# 4822# 4823# 4824# 4825 4826# 4827 4829# 4830# 4831# 4832# 4833# 4834#
4835 4836# 4837 4839# 4840# 4841# 4842 4843# 4844 4846# 4847# 4848# 4849# 4850# 4851#
4852 4853# 4854 4856# 4857# 4858# 4859# 4860 4861# 4862 4864# 4865# 4866# 4867# 4868#
4869# 4870 4871# 4872 4874# 4875# 4876# 4877 4878# 4879 4881# 4882# 4883# 4884# 4885#
4886# 4887 4888# 4889 4892# 4901# 4902# 4903# 4904 4905# 4906 4908# 4909# 4910# 4911#
4912 4913# 4914 4916# 4917# 4918 4919# 4920 4922# 4923# 4924# 4925# 4926 4927# 4928
4930# 4931# 4932# 4933# 4934 4935# 4936 4938# 4939# 4940# 4941# 4942# 4943# 4944 4945#
4946 4948# 4949# 4950# 4951 4952# 4953 4955# 4956# 4957# 4958# 4959# 4960# 4961 4962#
4963 4965# 4966# 4967# 4968# 4969 4970# 4971 4973# 4974# 4975# 4976# 4977# 4978# 4979
4980# 4981 4983# 4984# 4985# 4986 4987# 4988 4990# 4991# 4992# 4993# 4994# 4995# 4996
4997# 4998 5001# 5010# 5011# 5012# 5013# 5014 5015# 5016 5018# 5019# 5020# 5021# 5022
5023# 5024 5026# 5027# 5028 5029# 5030 5032# 5033# 5034# 5035# 5036 5037# 5038 5040#
5041# 5042# 5043# 5044 5045# 5046 5048# 5049# 5050# 5051# 5052 5053# 5054 5056# 5057#
5058# 5059# 5060# 5061# 5062 5063# 5064 5066# 5067# 5068# 5069 5070# 5071 5073# 5074#
5075# 5076# 5077# 5078# 5079 5080# 5081 5083# 5084# 5085# 5086# 5087 5088# 5089 5091#
5092# 5093# 5094# 5095# 5096# 5097 5098# 5099 5101# 5102# 5103# 5104 5105# 5106 5108#
5109# 5110# 5111# 5112# 5113# 5114 5115# 5116 5119# 5128# 5129# 5130# 5131 5132# 5133
5135# 5136# 5137# 5138# 5139 5140# 5141 5143# 5144# 5145 5146# 5147 5149# 5150# 5151#
5152# 5153 5154# 5155 5157# 5158# 5159# 5160# 5161 5162# 5163 5165# 5166# 5167# 5168#
5169# 5170# 5171 5172# 5173 5175# 5176# 5177# 5178 5179# 5180 5182# 5183# 5184# 5185#
5186# 5187# 5188 5189# 5190 5192# 5193# 5194# 5195# 5196 5197# 5198 5200# 5201# 5202#
5203# 5204# 5205# 5206 5207# 5208 5210# 5211# 5212# 5213 5214# 5215 5217# 5218# 5219#
5220# 5221# 5222# 5223 5224# 5225 5228# 5237# 5238# 5239# 5240# 5241 5242# 5243 5245#
5246# 5247 5248# 5249 5251# 5252# 5253# 5254# 5255 5256# 5257 5259# 5260# 5261# 5262#
5263 5264# 5265 5267# 5268# 5269# 5270# 5271# 5272# 5273 5274# 5275 5277# 5278# 5279#
5280 5281# 5282 5284# 5285# 5286# 5287# 5288# 5289# 5290 5291# 5292 5294# 5295# 5296#
5297# 5298 5299# 5300 5302# 5303# 5304# 5305# 5306# 5307# 5308 5309# 5310 5312# 5313#
5314# 5315 5316# 5317 5319# 5320# 5321# 5322# 5323# 5324# 5325 5326# 5327 5330# 5339#
5340# 5341# 5342 5343# 5344 5346# 5347# 5348# 5349# 5350 5351# 5352 5354# 5355# 5356
5357# 5358 5360# 5361# 5362# 5363# 5364 5365# 5366 5368# 5369# 5370# 5371# 5372 5373#
5374 5376# 5377# 5378# 5379# 5380 5381# 5382 5384# 5385# 5386# 5387# 5388# 5389# 5390
5391# 5392 5394# 5395# 5396# 5397 5398# 5399 5401# 5402# 5403# 5404# 5405# 5406# 5407
5408# 5409 5411# 5412# 5413# 5414# 5415 5416# 5417 5419# 5420# 5421# 5422# 5423# 5424#
5425 5426# 5427 5429# 5430# 5431# 5432 5433# 5434 5436# 5437# 5438# 5439# 5440# 5441#
5442 5443# 5444 5447# 5456# 5457# 5458# 5459# 5460 5461# 5462 5464# 5465# 5466 5467#
5468 5470# 5471# 5472# 5473# 5474 5475# 5476 5478# 5479# 5480# 5481# 5482 5483# 5484
5486# 5487# 5488# 5489# 5490# 5491# 5492 5493# 5494 5496# 5497# 5498# 5499 5500# 5501
5503# 5504# 5505# 5506# 5507# 5508# 5509 5510# 5511 5513# 5514# 5515# 5516# 5517 5518#
5519 5521# 5522# 5523# 5524# 5525# 5526# 5527 5528# 5529 5531# 5532# 5533# 5534 5535#
5536 5538# 5539# 5540# 5541# 5542# 5543# 5544 5545# 5546 5549# 5568# 5620# 5621# 5623#

```
                6439#  6464#  6471#  6472#  6500#  6507#  6527#  6528#  6532#  6533#  6558#  6563#  6564#  6589#  6592#
                6612#  6613#  6646#  6672#  6673#  6677#  6678#  6694#  6702#  6713#  6720#  6725#  6726#  6741#  6752#
                6759#  6764#  6765#  6780#  6791#  6798#  6801#  6815#  6816#  6831#  6839#  6844#  6868#  6869#  6964#
                6990#  6991#  7100#  7126#  7127#  7133#  7134#  7164#  7169#  7176#  7181#  7188#  7195#  7196#  7226#
                7231#  7238#  7243#  7250#  7254#  7283#  7284#  7290#  7291#  7340#  7347#  7348#  7397#  7401#  7418#
                7419#  7443#  7469#  7476#  7494#  7495#  7516#  7535#  7549#  7569#  7570#  7595#  7609#  7629#  7630#
                7655#  7669#  7688#  7689#  7709#  7725#  7726#  7761#  7794#  7795#  7807#  7808#  7819#  7824#  7835#
                7840#  7867#  7882#  7883#  7895#  7900#  7905#  7911#  7912#  7924#  7931#  7936#  7937#  7949#  7956#
                7961#  7962#  7975#  7982#  7988#  7989#  8000#  8007#  8018#  8019#  8032#  8037#  8044#  8055#  8062#
                8068#  8069#  8076#  8082#  8083#  8094#  8101#  8110#  8111#  8122#  8127#  8134#  8139#  8146#  8153#
                8159#  8182#  8183#  8209#  8223#  8246#  8247#  8273#  8287#  8310#  8311#  8337#  8351#  8374#  8375#
                8401#  8415#  8438#  8439#  8465#  8479#  8502#  8503#  8528#  8543#  8566#  8567#  8592#  8607#  8625#
                8626#  8653#  8673#  8674#  8708#  8724#  8790#  8822#  8842#  8977#
MSIOSE             1#  1894#
MSLDRO             1#  1894#  5620#  5626#  5632#  5638#  5660#  5811#  5820#
MSMASK             1#  1894#
MSMCHI             1#  1894#
MSMCLO             1#  1894#
MSMSK1             1#  1894#
MSPOP              1#  1894#  2101#  2125#  4708#  4781#  4891#  5000#  5118#  5227#  5329#  5446#  5548#  5567#  5589#
                5758#  5825#  5844#  5872#  5897#  5952#  6083#  6168#  6202#  6238#  6276#  6405#  6463#  6499#  6506#
                6557#  6588#  6591#  6645#  6719#  6758#  6797#  6800#  6843#  6963#  7099#  7187#  7249#  7253#  7339#
                7396#  7400#  7475#  7548#  7608#  7668#  7708#  7760#  7866#  7904#  7930#  7955#  7981#  8006#  8061#
                8075#  8100#  8152#  8158#  8222#  8286#  8350#  8414#  8478#  8542#  8606#  8652#  8821#  8882#  9000#
                9053#
MSPRIN             1#  1894#  4133#  4148#  4310#  4326#  4700#  4716#  4724#  4730#  4738#  4746#  4754#  4764#  4771#
                4791#  4799#  4805#  4813#  4821#  4829#  4839#  4846#  4856#  4864#  4874#  4881#  4901#  4908#  4916#
                4922#  4930#  4938#  4948#  4955#  4965#  4973#  4983#  4990#  5010#  5018#  5026#  5032#  5040#  5048#
                5056#  5066#  5073#  5083#  5091#  5101#  5108#  5128#  5135#  5143#  5149#  5157#  5165#  5175#  5182#
                5192#  5200#  5210#  5217#  5237#  5245#  5251#  5259#  5267#  5277#  5284#  5294#  5302#  5312#  5319#
                5339#  5346#  5354#  5360#  5368#  5376#  5384#  5394#  5401#  5411#  5419#  5429#  5436#  5456#  5464#
                5470#  5478#  5486#  5496#  5503#  5513#  5521#  5531#  5538#  5703#  5725#  5736#  5865#  6690#  6698#
                6737#  6776#  6827#  6835#
MSPUSH             1#  1894#  1900#  2089#  2115#  4698#  4714#  4789#  4899#  5008#  5126#  5235#  5337#  5454#  5564#
                5584#  5602#  5808#  5840#  5859#  5895#  5925#  5926   5975#  5976   6108#  6109   6184#  6185   6219#
                6220   6255#  6256   6296#  6297   6433#  6434   6438#  6439   6471#  6472   6527#  6528   6532#  6533
                6563#  6564   6612#  6613   6672#  6673   6677#  6678   6725#  6726   6764#  6765   6815#  6816   6868#
                6869   6990#  6991   7126#  7127   7133#  7134   7195#  7196   7283#  7284   7290#  7291   7347#  7348
                7418#  7419   7494#  7495   7569#  7570   7629#  7630   7688#  7689   7725#  7726   7794#  7795   7807#
                7808   7882#  7883   7911#  7912   7936#  7937   7961#  7962   7988#  7989   8018#  8019   8068#  8069
                8082#  8083   8110#  8111   8182#  8183   8246#  8247   8310#  8311   8374#  8375   8438#  8439   8502#
                8503   8566#  8567   8625#  8626   8673#  8674   8842#  8977#
MSPUT              1#  1894#  4133#  4148#  4310#  4326#  4700#  4716#  4724#  4730#  4738#  4746#  4754#  4764#  4771#
                4791#  4799#  4805#  4813#  4821#  4829#  4839#  4846#  4856#  4864#  4874#  4881#  4901#  4908#  4916#
                4922#  4930#  4938#  4948#  4955#  4965#  4973#  4983#  4990#  5010#  5018#  5026#  5032#  5040#  5048#
                5056#  5066#  5073#  5083#  5091#  5101#  5108#  5128#  5135#  5143#  5149#  5157#  5165#  5175#  5182#
                5192#  5200#  5210#  5217#  5237#  5245#  5251#  5259#  5267#  5277#  5284#  5294#  5302#  5312#  5319#
                5339#  5346#  5354#  5360#  5368#  5376#  5384#  5394#  5401#  5411#  5419#  5429#  5436#  5456#  5464#
                5470#  5478#  5486#  5496#  5503#  5513#  5521#  5531#  5538#  5703#  5725#  5736#  5865#  6690#  6698#
                6737#  6776#  6827#  6835#
MSPUT1             1#  1894#  4133#  4134   4135   4148#  4149   4310#  4311   4326#  4327   4700#  4701   4702   4703
                4716#  4717   4718   4719   4724#  4725   4730#  4731   4732   4733   4738#  4739   4740   4741   4746#
                4747   4748   4749   4754#  4755   4756   4757   4758   4759   4764#  4765   4766   4771#  4772   4773
                4774   4775   4776   4791#  4792   4793   4794   4799#  4800   4805#  4806   4807   4808   4813#  4814
                4815   4816   4821#  4822   4823   4824   4829#  4830   4831   4832   4833   4834   4839#  4840   4841
                4846#  4847   4848   4849   4850   4851   4856#  4857   4858   4859   4864#  4865   4866   4867   4868
```

```
        4869  4874#  4875  4876  4881#  4882  4883  4884  4885  4886  4901#  4902  4903  4908#  4909
        4910  4911   4916#  4917  4922#  4923  4924  4925  4930#  4931  4932  4933  4938#  4939  4940
        4941  4942   4943  4948#  4949  4950  4955#  4956  4957  4958  4959  4960  4965#  4966  4967
        4968  4973#  4974  4975  4976  4977  4978  4983#  4984  4985  4990#  4991  4992  4993  4994
        4995  5010#  5011  5012  5013  5018#  5019  5020  5021  5026#  5027  5032#  5033  5034  5035
        5040#  5041  5042  5043  5048#  5049  5050  5051  5056#  5057  5058  5059  5060  5061  5066#
        5067  5068  5073#  5074  5075  5076  5077  5078  5083#  5084  5085  5086  5091#  5092  5093
        5094  5095  5096  5101#  5102  5103  5108#  5109  5110  5111  5112  5113  5128#  5129  5130
        5135#  5136  5137  5138  5143#  5144  5149#  5150  5151  5152  5157#  5158  5159  5160  5165#
        5166  5167  5168  5169  5170  5175#  5176  5177  5182#  5183  5184  5185  5186  5187  5192#
        5193  5194  5195  5200#  5201  5202  5203  5204  5205  5210#  5211  5212  5217#  5218  5219
        5220  5221  5222  5237#  5238  5239  5240  5245#  5246  5251#  5252  5253  5254  5259#  5260
        5261  5262  5267#  5268  5269  5270  5271  5272  5277#  5278  5279  5284#  5285  5286  5287
        5288  5289  5294#  5295  5296  5297  5302#  5303  5304  5305  5306  5307  5312#  5313  5314
        5319#  5320  5321  5322  5323  5324  5339#  5340  5341  5346#  5347  5348  5349  5354#  5355
        5360#  5361  5362  5363  5368#  5369  5370  5371  5376#  5377  5378  5379  5384#  5385  5386
        5387  5388  5389  5394#  5395  5396  5401#  5402  5403  5404  5405  5406  5411#  5412  5413
        5414  5419#  5420  5421  5422  5423  5424  5429#  5430  5431  5436#  5437  5438  5439  5440
        5441  5456#  5457  5458  5459  5464#  5465  5470#  5471  5472  5473  5478#  5479  5480  5481
        5486#  5487  5488  5489  5490  5491  5496#  5497  5498  5503#  5504  5505  5506  5507  5508
        5513#  5514  5515  5516  5521#  5522  5523  5524  5525  5526  5531#  5532  5533  5538#  5539
        5540  5541  5542  5543  5703#  5704  5725#  5726  5736#  5737  5738  5865#  5866  5867  6690#
        6691  6692  6698#  6699  6700  6737#  6738  6739  6776#  6777  6778  6827#  6828  6829  6835#
        6836  6837
MSRADI     1#  1894#  5749#  8846#  8851#  8857#  8863#  8869#  8875#  8981#  8985#  8989#  8993#
MSRBRO     1#  1894#
MSRNRO     1#  1894#  5660#  5662
MSSETS     1#  1894#  1900#  2089#  2115#  4698#  4714#  4789#  4899#  5008#  5126#  5235#  5337#  5454#  5564#
        5584#  5602#  5808#  5840#  5859#  5895#  5926#  5976#  6109#  6185#  6220#  6256#  6297#  6434#  6439#
        6472#  6528#  6533#  6564#  6613#  6673#  6678#  6726#  6765#  6816#  6869#  6991#  7127#  7134#  7196#
        7284#  7291#  7348#  7419#  7495#  7570#  7630#  7689#  7726#  7795#  7808#  7883#  7912#  7937#  7962#
        7989#  8019#  8069#  8083#  8111#  8183#  8247#  8311#  8375#  8439#  8503#  8567#  8626#  8674#  8842#
        8977#
MSSTAR     1#  1894#
MSSVC      1#  1894#  3142  3152  3166  3176  3287  3297  3459  3469  3483  3493  3532  3542  3580
        3590  3602  3612  3762  3771  3794  3804  3817  3827  3840  3850  3863  3873  4133#  4137
        4148#  4151  4252  4262  4299  4310#  4313  4320#  4326#  4329  4368  4700#  4705  4708#  4709
        4716#  4721  4724#  4727  4730#  4735  4738#  4743  4746#  4751  4754#  4761  4764#  4768  4771#
        4778  4781#  4782  4791#  4796  4799#  4802  4805#  4810  4813#  4818  4821#  4826  4829#  4836
        4839#  4843  4846#  4853  4856#  4861  4864#  4871  4874#  4878  4881#  4888  4891#  4892  4901#
        4905  4908#  4913  4916#  4919  4922#  4927  4930#  4935  4938#  4945  4948#  4952  4955#  4962
        4965#  4970  4973#  4980  4983#  4987  4990#  4997  5000#  5001  5010#  5015  5018#  5023  5026#
        5029  5032#  5037  5040#  5045  5048#  5053  5056#  5063  5066#  5070  5073#  5080  5083#  5088
        5091#  5098  5101#  5105  5108#  5115  5118#  5119  5128#  5132  5135#  5140  5143#  5146  5149#
        5154  5157#  5162  5165#  5172  5175#  5179  5182#  5189  5192#  5197  5200#  5207  5210#  5214
        5217#  5224  5227#  5228  5237#  5242  5245#  5248  5251#  5256  5259#  5264  5267#  5274  5277#
        5281  5284#  5291  5294#  5299  5302#  5309  5312#  5316  5319#  5326  5329#  5330  5339#  5343
        5346#  5351  5354#  5357  5360#  5365  5368#  5373  5376#  5381  5384#  5391  5394#  5398  5401#
        5408  5411#  5416  5419#  5426  5429#  5433  5436#  5443  5446#  5447  5456#  5461  5464#  5467
        5470#  5475  5478#  5483  5486#  5493  5496#  5500  5503#  5510  5513#  5518  5521#  5528  5531#
        5535  5538#  5545  5548#  5549  5567#  5568  5620#  5621  5626#  5627  5632#  5633  5638#  5639
        5660#  5661  5703#  5706  5719#  5725#  5728  5731#  5736#  5740  5746#  5758#  5759  5811#  5812
        5820#  5821  5825#  5826  5844#  5845  5862#  5865#  5869  5872#  5873  5897#  5898  5952#  5953
        6026  6077  6083#  6084  6162  6168#  6169  6202#  6203  6238#  6239  6276#  6277  6332  6375
        6392  6405#  6406  6438#  6439  6463#  6464  6471#  6472  6499#  6500  6506#  6507  6532#  6533
        6557#  6558  6563#  6564  6588#  6589  6591#  6592  6645#  6646  6677#  6678  6690#  6694  6698#
```

CZDMSF.P11 30-SEP-81 15:40 CROSS REFERENCE TABLE -- MACRO NAMES

```
        3300    3459#   3460    3461    3462    3469#   3470    3471    3472    3483#   3484    3485    3486    3493#   3494
        3495    3496    3532#   3533    3534    3535    3542#   3543    3544    3545    3580#   3581    3582    3583    3590#
        3591    3592    3593    3602#   3603    3604    3605    3612#   3613    3614    3615    3762#   3763    3764    3765
        3771#   3772    3773    3774    3794#   3795    3796    3797    3804#   3805    3806    3807    3817#   3818    3819
        3820    3827#   3828    3829    3830    3840#   3841    3842    3843    3850#   3851    3852    3853    3863#   3864
        3865    3866    3873#   3874    3875    3876    4252#   4253    4254    4255    4262#   4263    4264    4265    4299#
        4300    4301    4302    4368#   4369    4370    4371    5746#   5748    5749#   6026#   6027    6028    6029    6077#
        6078    6079    6080    6162#   6163    6164    6165    6332#   6333    6334    6335    6375#   6376    6377    6378
        6392#   6393    6394    6395    6713#   6714    6715    6716    6752#   6753    6754    6755    6791#   6792    6793
        6794    7164#   7165    7166    7167    7176#   7177    7178    7179    7226#   7227    7228    7229    7238#   7239
        7240    7241    7443#   7444    7445    7446    7469#   7470    7471    7472    7516#   7517    7518    7519    7535#
        7536    7537    7538    7595#   7596    7597    7598    7655#   7656    7657    7658    7819#   7820    7821    7822
        7835#   7836    7837    7838    7895#   7896    7897    7898    7924#   7925    7926    7927    7949#   7950    7951
        7952    7975#   7976    7977    7978    8000#   8001    8002    8003    8032#   8033    8034    8035    8044#   8045
        8046    8047    8055#   8056    8057    8058    8094#   8095    8096    8097    8122#   8123    8124    8125    8134#
        8135    8136    8137    8146#   8147    8148    8149    8209#   8210    8211    8212    8273#   8274    8275    8276
        8337#   8338    8339    8340    8401#   8402    8403    8404    8465#   8466    8467    8468    8528#   8529    8530
        8531    8592#   8593    8594    8595    8708#   8709    8710    8711    8724#   8725    8726    8727    8790#   8791
        8792    8793    8846#   8851#   8857#   8863#   8869#   8875#   8981#   8985#   8989#   8993#   9056    9057
MSXFER     1#   1894#
OPEN       1#   1894#
POINTE     1#   1894#   1922
PRINTB     1#   1894#   4699    4715    4723    4729    4737    4790    4798    4804    4812    4900    4907    4915    4921
        5009    5017    5025    5031    5039    5127    5134    5142    5148    5236    5244    5250    5338    5345    5353
        5359    5367    5455    5463    5469
PRINTF     1#   1894#   4132    4147    4309    4325    5702    5724    5735    5864    6689    6697    6736    6775    6826
        6834
PRINTS     1#   1894#
PRINTX     1#   1894#   4745    4753    4763    4770    4820    4828    4838    4845    4855    4863    4873    4880    4929
        4937    4947    4954    4964    4972    4982    4989    5047    5055    5065    5072    5082    5090    5100    5107
        5156    5164    5174    5181    5191    5199    5209    5216    5258    5266    5276    5283    5293    5301    5311
        5318    5375    5383    5393    5400    5410    5418    5428    5435    5477    5485    5495    5502    5512    5520
        5530    5537
READBU     1#   1894#
READEF     1#   1894#   5619    5625    5631    5637
RFLAGS     1#   1894#
SETPRI     1#   1894#   5810
SETVEC     1#   1894#
SLASH      1#   1894#
STARS      1#   1894#
SVC        1#   1892#   1893
XFER       1#   1894#
XFERF      1#   1894#
XFERT      1#   1894#


. ABS.  040244      000


 ERRORS DETECTED:  0

 CZDMSF/I,CZDMSF.SEQ/CRF/SOL/NL:TOC=SVC34R.MLB,CZDMSF.P11
 RUN-TIME: 38 47 4 SECONDS
 RUN-TIME RATIO: 134/91=1.4
 CORE USED:  20K  (39 PAGES)
```