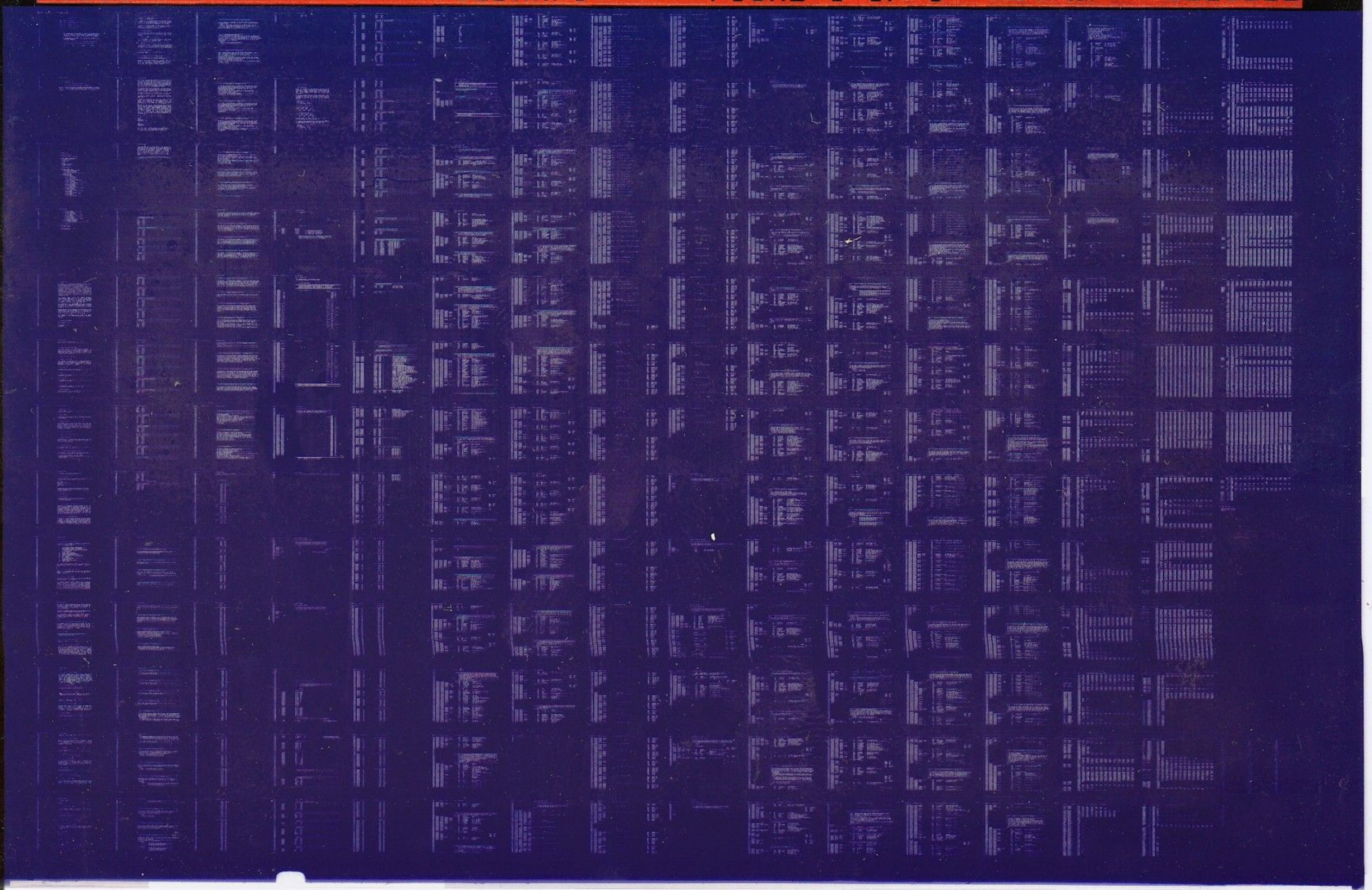
DMP-11,DMR-11, M8203 STATIC DIAG AH-E232F-MC
M8203 CZDMRF0 FICHE 1 OF 1



```
.REM @

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

PRODUCT CODE:   AC-E231F-MC

PRODUCT NAME:   CZDMRF0 M8203 STATIC DIAG #1

PRODUCT DATE:   FEBRUARY  1982

MAINTAINER:     DIAGNOSTIC ENGINEERING

AUTHOR:         DAVID HOFFMAN
```

```
36
37
38
39
40                          ****** MODIFICATION HISTORY ******
41
42                        THE MODIFICATION HISTORY BEGINS WITH VERSION F.
43
44          FOR THE 11-NOV-81 RELEASE BERT KLEINSCHMIDT MODIFIED VERSION E TO CREATE
45          VERSION F.  THE ONLY CHANGE THAT WAS MADE WAS THE UPDATING OF THE VERSION
46          LETTER (TO F) IN THE DOCUMENTATION AND HEADER CODE.  THIS CHANGE WAS MADE
47          TO CORRECT THE ACCIDENTAL RELEASE AND DISTRIBUTION OF A BAD COPY OF
48          CZDMRE.
```

CONTENTS

CZDMRF.P11 03-NOV-81 10:29 PROGRAM DOCUMENT

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

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
CZDMR-F-0
M8203 STATIC LOGIC TESTS - PART 1 OF 2
UNIT IS M8203
DR>
```

THE OPERATOR THEN PROCEEDS 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.

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.

6.3.6.2 EFFECT OF DROP COMMAND

THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.

6.3.7 PRINT COMMAND

****
PRI(NT)
****

6.3.7.1 EFFECT OF PRINT COMMAND

THE TOTAL NUMBER OF ERRORS FOR EACH UNIT SINCE THE LAST START OR RESTART COMMAND ARE PRINTED. THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

6.3.8 DISPLAY COMMAND

****
DIS(PLAY)/UNITS:<UNIT-LIST>
****

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

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

6.3.8.2 EFFECT OF DISPLAY COMMAND

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS THAT WERE DROPPED BY THE OPERATOR "DROP" COMMAND ARE SO DESIGNATED.

6.3.9 FLAGS COMMAND

****
FLA(GS)
****

6.3.9.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

6.3.10 ZFLAGS COMMAND

****
ZFL(AGS)
****

6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

6.3.11 CONTROL CHARACTERS

A CONTROL C (C) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES A RETURN TO COMMAND MODE.

A CONTROL Z (Z) ENTERED DURING ONE OF THE THREE OPERATOR DIALOGUES- HARD CORE QUESTIONS (SEE 6.2),HARDWARE DIALOGUE (SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.

A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES ALL TELETYPE OUTPUT TO BE SURPRESSED FOR THE REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER O IS TYPED, WHICH RESTORES NORMAL TELETYPE OUTPUT.

6.3.12 HARDWARE PARAMETERS

THE FOLLOWING 2 QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

1. DEVICE CSR ADDRESS : (O) 160170?

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SELO) RESIDE ON THE UNIBUS. THE ALLOWABLE RANGE IS 160000-177776 (OCTAL), AND THE DEFAULT VALUE IS 160170.

2. M8207 RUN SWITCH (E28 SW7) - TYPE 0 IF OFF, 1 IF ON : (O) 1 ?

THIS TELLS THE PROGRAM IF THE RUN SWITCH ON THE MICROPROCESSOR IS SET OR NOT. IF IT IS SET, RUN IS NOT INHIBITED, AND TESTS REQUIRING THE RUN STAIE CAN BE EXECUTED. THE ALLOWABLE VALUES ARE 0 AND 1, AND THE DEFAULT VALUE IS 1 (RUN IS NOT INHIBITED).

6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY PART 1 OF THE STATIC LOGIC TESTS.

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).

```
758
759
760                                 7.0 DEVICE INFORMATION TABLES
761
762
763
764                                 ;*****************************************************************************
765                                 * MAINTENANCE REGISTER - BSEL1
766                                 ;*****************************************************************************
767                                 RUN     = BIT7
768                                 MCLR    = BIT6
769                                 STEPLU  = BIT4
770                                 LULOOP  = BIT3
771                                 ROMO    = BIT2
772                                 ROMI    = BIT1
773                                 STEPMP  = BIT0
774
775                                 ;*****************************************************************************
776                                 * OBUS REG 10 - TRANSMITTER BUFFER
777                                 ;*****************************************************************************
778                                 TX7     = BIT7
779                                 TX6     = BIT6
780                                 TX5     = BIT5
781                                 TX4     = BIT4
782                                 TX3     = BIT3
783                                 TX2     = BIT2
784                                 TX1     = BIT1
785                                 TX0     = BIT0
786
787                                 ;*****************************************************************************
788                                 * OBUS REG 11
789                                 ;*****************************************************************************
790                                 OC      = BIT7
791                                 GOAH    = BIT3
792                                 ABORT   = BIT2
793                                 EOM     = BIT1
794                                 SOM     = BIT0
795
796                                 ;*****************************************************************************
797                                 * OBUS REG 12
798                                 ;*****************************************************************************
799                                 IC      = BIT7
800                                 BPOLL   = BIT6
801                                 LULP    = BIT5
802
803                                 ;*****************************************************************************
804                                 * OBUS REG 13
805                                 ;*****************************************************************************
806                                 POLL    = BIT7
807                                 DTR     = BIT6
808                                 SELFR   = BIT5
809                                 HDX     = BIT4
810                                 MAINT1  = BIT3
811                                 MAINT2  = BIT2
812                                 SELSBY  = BIT1
813
```

CZDMRF.P11 03-NOV-81 10:29 PROGRAM DOCUMENT

```
814                     ;*****************************************************************************
815                     * OBUS REG 14
816                     ;*****************************************************************************
817                     TXEN    = BIT6
818                     DISSI   = BIT5
819                     RDAX    = BIT4
820                     WAX     = BIT3
821                     ENAX    = BIT2
822                     AX2     = BIT1
823                     AX1     = BIT0
824
825                     ;*****************************************************************************
826                     * OBUS REG 17
827                     ;*****************************************************************************
828                     CRC2    = BIT7
829                     CRC1    = BIT6
830                     IDLE    = BIT5
831                     SECA    = BIT4
832                     STRIP   = BIT3
833                     RDALL   = BIT2
834                     ;ERR    = BIT1
835                     DDCMP   = BIT0
836
837                     ;*****************************************************************************
838                     * IBUS REG 10 - RECEIVER BUFFER
839                     ;*****************************************************************************
840                     RX7     = BIT7
841                     RX6     = BIT6
842                     RX5     = BIT5
843                     RX4     = BIT4
844                     RX3     = BIT3
845                     RX2     = BIT2
846                     RX1     = BIT1
847                     RX0     = BIT0
848
849                     ;*****************************************************************************
850                     * IBUS REG 11
851                     ;*****************************************************************************
852                     OC      = BIT7
853                     OACT    = BIT6
854                     SW3     = BIT5
855                     ORDY    = BIT4
856                     SW2     = BIT3
857                     SW1     = BIT2
858                     SW0     = BIT1
859                     UNRR    = BIT0
860
861                     ;*****************************************************************************
862                     * IBUS REG 12
863                     ;*****************************************************************************
864                     IC      = BIT7
865                     IACT    = BIT6
866                     LULP    = BIT5
867                     IRDY    = BIT4
868                     OVRR    = BIT3
869                     RAB     = BIT2
```

```
870                             EBLK    = BIT1
871                             BCC     = BIT0
872
873                             ;*****************************************************************************
874                             * IBUS REG 13
875                             ;*****************************************************************************
876                             RING    = BIT7
877                             DTR     = BIT6
878                             RTS     = BIT5
879                             HDX     = BIT4
880                             MODR    = BIT3
881                             CS      = BIT2
882                             STBY    = BIT1
883                             CARR    = BIT0
884
885                             ;*****************************************************************************
886                             * IBUS REG 14
887                             ;*****************************************************************************
888                             READY   = BIT7
889                             TXEN    = BIT6
890                             DISSI   = BIT5
891                             RDAX    = BIT4
892                             WAX     = BIT3
893                             ENAX    = BIT2
894                             AX2     = BIT1
895                             AX1     = BIT0
896
897                             ;*****************************************************************************
898                             * IBUS REG 17
899                             ;*****************************************************************************
900                             SIGR    = BIT7
901                             SIGQ    = BIT6
902                             TXDATA  = BIT5
903                             OCOR    = BIT4
904                             ICIR    = BIT3
905                             TESTMD  = BIT2
906                             MCLK  = BIT1
907                             DDCMP   = BIT0
908
909                             ;*****************************************************************************
910                             * AX0-15 - USYRT REG 0 (READ ONLY)
911                             ;*****************************************************************************
912                             RX7     = BIT7
913                             RX6     = BIT6
914                             RX5     = BIT5
915                             RX4     = BIT4
916                             RX3     = BIT3
917                             RX2     = BIT2
918                             RX1     = BIT1
919                             RX0     = BIT0
920
921                             ;*****************************************************************************
922                             * AX0-16 - USYRT REG 1 (READ ONLY)
923                             ;*****************************************************************************
924                             RERR    = BIT7
925                             ASBC2   = BIT6
```

```
926                     ASBC1   = BIT5
927                     ASBC0   = BIT4
928                     ROR     = BIT3
929                     RABT    = BIT2
930                     REOM    = BIT1
931                     RSOM    = BIT0
932
933                     ;*****************************************************************************************
934                     * AX1-15 - USYRT REG 2
935                     ;*****************************************************************************************
936                     TX7     = BIT7
937                     TX6     = BIT6
938                     TX5     = BIT5
939                     TX4     = BIT4
940                     TX3     = BIT3
941                     TX2     = BIT2
942                     TX1     = BIT1
943                     TX0     = BIT0
944
945                     ;*****************************************************************************************
946                     * AX1-16 - USYRT REG 3
947                     ;*****************************************************************************************
948                     TERR    = BIT7
949                     TXGA    = BIT3
950                     TXAB    = BIT2
951                     TEOM    = BIT1
952                     TSOM    = BIT0
953
954                     ;*****************************************************************************************
955                     * AX2-15 - USYRT REG 4
956                     ;*****************************************************************************************
957                     SYN7    = BIT7
958                     SYN6    = BIT6
959                     SYN5    = BIT5
960                     SYN4    = BIT4
961                     SYN3    = BIT3
962                     SYN2    = BIT2
963                     SYN1    = BIT1
964                     SYN0    = BIT0
965                     SYNCH   = 226
966
967                     ;*****************************************************************************************
968                     * AX2-16 - USYRT REG 5
969                     ;*****************************************************************************************
970                     APA     = BIT7
971                     DDC     = BIT6
972                     STR     = BIT5
973                     SEC     = BIT4
974                     IDL     = BIT3
975                     CRCTY2  = BIT2
976                     CRCTY1  = BIT1
977                     CRCTY0  = BIT0
978
979                     ;*****************************************************************************************
980                     * AX3-15 - USYRT REG 6
981                     ;*****************************************************************************************
```

```
                I422    = BIT7
                XYZ     = BIT6
                C32BCC  = BIT5
                V35     = BIT4
                INTGRL  = BIT3
                C32ENB  = BIT2
                OP      = BIT1
                TEST    = BIT0
                AX315U  = I422!XYZ!C32BCC!V35!INTGRL!OP

                ;*****************************************************************************
                * AX3-16 - USYRT REG 7
                ;*****************************************************************************
                TXLEN2  = BIT7
                TXLEN1  = BIT6
                TXLEN0  = BIT5
                RXLEN2  = BIT2
                RXLEN1  = BIT1
                RXLEN0  = BIT0
```

```
1006
1007
1008
1009
1010                                    8.0 TEST DESCRIPTIONS
1011
1012
1013
1014
1015
1016                                    ;*****************************************************************************
1017                                        TEST 1 - MICROPROCESSOR CSR ADDRESSING TEST (SEL0)
1018                                    *
1019                                    * THIS TEST ADDRESSES THE FIRST MICROPROCESSOR CSR (SEL0), TO MAKE SURE
1020                                    * THAT A NON-EXISTENT MEMORY TIME-OUT TRAP DOES NOT OCCUR WHILE
1021                                    * ATTEMPTING TO ADDRESS THE MICROPROCESSOR.
1022                                    ;*****************************************************************************
1023
1024
1025
1026
1027
1028                                    ;*****************************************************************************
1029                                        TEST 2 - INBUS/OUTBUS REG 14 INITIALIZATION TEST
1030                                    *
1031                                    * MASTER CLEAR (MCLR) IS SET IN THE MICROPROCESSOR, IBUS REG 14 IS READ
1032                                    * AND COMPARED TO 200.
1033                                    ;*****************************************************************************
1034
1035
1036
1037
1038
1039                                    ;*****************************************************************************
1040                                        TEST 3 - INBUS/OUTBUS REG 14 READ/WRITE BIT TEST
1041                                    *
1042                                    * WRITE, READ, AND COMPARE ALL WORDS OF DATA PATTERN A INTO REG 14,
1043                                    * A BYTE AT A TIME. NON-R/W BITS ARE MASKED OFF TO 0 BEFORE WRITING AND
1044                                    * READING.
1045                                    *    DATA PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
1046                                    *                     375,373,367,357,337,277,177.
1047                                    ;*****************************************************************************
1048
1049
1050
1051
1052
1053                                    ;*****************************************************************************
1054                                        TEST 4 - REG 14 MASTER CLEAR TEST
1055                                    * WRITE 377 INTO REG 14, ISSUE MASTER CLEAR, READ REG 14 AND COMPARE
1056                                    * TO 200.
1057                                    ;*****************************************************************************
1058
1059
1060
1061
```

```
1062
1063                    ;*****************************************************************************
1064                        TEST 5 - REG 14 UNIBUS RESET (INIT) TEST
1065                    * WRITE 377 INTO REG 14, ISSUE UNIBUS RESET (INIT), READ REG 14 AND COMPARE
1066                    * TO 200.
1067                    ;*****************************************************************************
1068
1069
1070
1071
1072
1073                    ;*****************************************************************************
1074                        TEST 6 - LINE UNIT FALSE SELECTION TEST
1075                    *
1076                    * FIRST, A MASTER CLEAR IS PERFORMED. THEN, THE PROGRAM SINGLE-STEPS THE
1077                    * MICROPROCESSOR THROUGH AN INSTRUCTION WHICH LOADS 041 (OCT) INTO THE MAR
1078                    * REGISTER (OBUS* ADRS 14). THEN, THE LINE UNIT REGISTER 14 IS READ AND CHECKED
1079                    * TO BE UNAFFECTED (STILL = 0). THIS TEST IS INTENDED TO DETECT A FALSE
1080                    * SELECTION OF THE LINE UNIT REGISTERS, WHEN THE LINE UNIT IS NOT BEING
1081                    * ACCESSED.
1082                    ;*****************************************************************************
1083
1084
1085
1086
1087
1088                    ;*****************************************************************************
1089                        TEST 7 - INBUS REG MASTER CLEAR TEST
1090                    *
1091                    * FIRST, ALL READ/WRITE BITS OF REGS 10-17 ARE SET BY LOADING A
1092                    * DIFFERENT WORD OF PATTERN G INTO EACH REG.  THEN,
1093                    * A MASTER CLEAR IS ISSUED AND EACH REG IS READ AND COMPARED TO A WORD OF
1094                    * PATTERN M, WHICH CONTAINS THE INITIALIZED STATES OF THE REGS. (UNPREDICTABLE
1095                    * BITS ARE MASKED OFF TO 0 BEFORE COMPARISON).
1096                    *   PATTERN G = 000,000,240,120,177,000,000,001
1097                    *   PATTERN M = 000,020,000,000,200,000,000,051
1098                    ;*****************************************************************************
1099
1100
1101
1102
1103
1104                    ;*****************************************************************************
1105                        TEST 8 - REGISTER 10-17 ADDRESSING TEST
1106                    *
1107                    * FIRST, A MASTER CLEAR IS ISSUED. THEN,
1108                    * WRITE A DIFFERENT WORD OF DATA PATTERN B INTO EACH OF REGS 10-17,
1109                    * AND AFTER EACH WRITE, READ AND COMPARE ALL REGS TO EXPECTED VALUES.
1110                    * UNPREDICTABLE BITS ARE MASKED OFF TO 0 WHEN READ FOR COMPARISON.
1111                    *    PATTERN B = 000,000,040,100,220,000,000,051
1112                    ;*****************************************************************************
1113
1114
1115
1116
1117
```

```
1118
1119                    ;*****************************************************************************
1120                        TEST 9 - REG 11 READ/WRITE BIT TEST
1121                    *
1122                    * WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN C INTO REG 11 :
1123                    *       DATA PATTERN C = 020,020,020.
1124                    ;*****************************************************************************
1125
1126
1127
1128
1129                    ;*****************************************************************************
1130                        TEST 10 - REG 12 READ/WRITE BIT TEST
1131                    *
1132                    * WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN D INTO REG 12 :
1133                    *       DATA PATTERN D = 000,040,000.
1134                    ;*****************************************************************************
1135
1136
1137
1138
1139
1140                    ;*****************************************************************************
1141                        TEST 11 - REG 13 READ/WRITE BIT TEST
1142                    *
1143                    * WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN E INTO REG 13 :
1144                    *       DATA PATTERN E = 000,120,020,100,120,000.
1145                    ;*****************************************************************************
1146
1147
1148
1149
1150
1151                    ;*****************************************************************************
1152                    ;   TEST 12 - REG 17 READ/WRITE BIT TEST
1153                    *
1154                    * WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN F INTO REG 17 :
1155                    *       DATA PATTERN F = 050,051,050.
1156                    ;*****************************************************************************
1157
1158
1159
1160
1161
1162                    ;*****************************************************************************
1163                        TEST 13 - MAINTENANCE CLOCK BIT TEST
1164                    *
1165                    * FIRST, A MASTER CLEAR IS ISSUED TO INIT ALL REGS. THEN, THE MICROPROCESSOR
1166                    * IS PLACED IN A LOOP ON AN INSTRUCTION, BY SETTING THE INSTRUCTION IN SEL6
1167                    * AND SETTING ROMI AND RUN IN BSEL1. THE INSTRUCTION IS ONE WHICH REPETITIVELY
1168                    * READS LINE UNIT REG 17 INTO BSEL2. THE PDP-11 CAN THEN SCAN BSEL2 TO MONITOR
1169                    * THE MAINTENANCE CLOCK BIT, MCLK. THE FOLLOWING SEQUENCE IS THEN PERFORMED
1170                    * TO MONITOR MCLK :
1171                    * - THE PROGRAM REPEATEDLY CHECKS THE MCLK BIT FOR THE 1 STATE, AND IF IT IS
1172                    *   NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC (DEPENDING ON THE PROCESSOR)
1173                    *   AN ERROR IS REPORTED. (THE MAINTENANCE CLOCK HAS A PERIOD OF 41.6 MICRO-
```

```
1174                    *   SEC).
1175                    * - THE PROGRAM NEXT REPEATEDLY CHECKS THE MCLK BIT FOR THE 0 STATE, AND IF
1176                    *   IT IS NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC AN ERROR IS REPORTED.
1177                    * - THE PROGRAM NEXT REPEATEDLY CHECKS MCLK BIT FOR THE 1 STATE AGAIN, AND
1178                    *   IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC, AN ERROR IS REPORTED.
1179                    *
1180                    * IF THE P-TABLE FOR THIS UNIT INDICATES THAT THE M8207 RUN SWITCH (E28 SW7)
1181                    * IS OFF, THE TEST WILL BE SKIPPED.
1182                    ;*****************************************************************************
1183
1184
1185
1186
1187
1188                    ;*****************************************************************************
1189                        TEST 14 - EXTENDED REGISTER MASTER CLEAR TEST
1190                    *
1191                    * FIRST, ALL READ/WRITE BITS OF EXTENDED REGS AX0-AX3 ARE SET BY LOADING
1192                    * A DIFFEPENT WORD OF PATTERN H INTO EACH REG. THEN, A MASTER CLEAR IS
1193                    * ISSUED AND EACH REG IS READ AND COMPARED TO A WORD OF PATTERN I, WHICH
1194                    * CONTAINS THE INITIALIZED STATES OF ALL THE EXTENDED REGS.
1195                    *    PATTERN H = 000,000,377,017,377,377,375,377
1196                    *    PATTERN I = 000,000,000,000,000,103,000,000
1197                    ;*****************************************************************************
1198
1199
1200
1201
1202
1203                    ;*****************************************************************************
1204                        TEST 15 - EXTENDED REGISTER ADDRESSING TEST
1205                    *
1206                    * FIRST, ISSUE A MASTER CLEAR TO PUT REGS INTO INITIALIZED STATES SHOWN IN
1207                    * PATTERN I. THEN, WRITE A DIFFERENT WORD OF PATTERN J INTO EACH EXTENDED (AX)
1208                    * REG, AND AFTER EACH WRITE, READ AND COMPARE ALL EXTENDED REGS TO EXPECTED
1209                    * VALUES.
1210                    *   PATTERN I = 000,000,000,000,000,103,000,000
1211                    *   PATTERN J = 000,000,001,002,004,103,040,100
1212                    ;*****************************************************************************
1213
1214
1215
1216
1217
1218                    ;*****************************************************************************
1219                        TEST 16 - REGS 15,16 / AX2-15,AX2-16 READ/WRITE BIT TEST
1220                    *
1221                    * USING REGS 15,16, THE INDIRECT REGS AX2-15,AX2-16 (USYRT REGS 4,5) ARE
1222                    * WRITTEN AND READ USING EACH WORD OF PATTERN K. AX2-15 IS COMPARED
1223                    * TO THE WORD WRITTEN, AND AX2-16 IS ALWAYS COMPARED TO 103. (AX2-16 IS NOT
1224                    * WRITEABLE).
1225                    *    PATTERN K =
1226                    *      FOR REG 15: 000,377,125,252,001,002,004,010,020,040,100,200,000,000,
1227                    *                  000,000,000,000,000,000,376,375,373,367,357,337,277,177,
1228                    *                  377,377,377,377,377,377,377,377
1229                    *      FOR REG 16: 000,377,125,252,000,000,000,000,000,000,000,000,001,002,
```

```
1230                    *                   004,010,020,040,100,200,377,377,377,377,377,377,377,377,
1231                    *                   376,375,373,367,357,337,277,177.
1232                    ;*****************************************************************************
1233
1234
1235
1236
1237
1238                    ;*****************************************************************************
1239                        TEST 17 - AX0-15,AX0-16 READ/WRITE BIT TEST
1240                    *
1241                    * IN THIS TEST, A MASTER CLEAR IS DONE, AND THEN A WRITE, READ, AND COMPARE
1242                    * ARE PERFORMED IN REGS AX0-15,AX0-16 USING EACH WORD OF PATTERN L.
1243                    * ANY BITS IN AX0-15,AX0-16 WHICH ARE NOT READ/WRITE ARE MASKED OFF (TO 0)
1244                    * IN THE EXPECTED VALUE BEFORE COMPARISON.
1245-                   *    PATTERN L =
1246                    *       FOR REG 15: 000,377,000
1247                    *       FOR REG 16: 000,377,000.
1248                    ;*****************************************************************************
1249
1250
1251
1252
1253
1254                    ;*****************************************************************************
1255                        TEST 18 - AX1-15,AX1-16 READ/WRITE BIT TEST
1256                    *
1257                    * IN THIS TEST, A MASTER CLEAR IS DONE, AND THEN A WRITE, READ, AND COMPARE
1258                    * ARE PERFORMED IN REGS AX1-15,AX1-16 USING EACH WORD OF PATTERN K.
1259                    * ANY BITS IN AX1-15,AX1-16 WHICH ARE NOT READ/WRITE ARE MASKED OFF (TO 0)
1260                    * IN THE EXPECTED VALUE BEFORE COMPARISON.
1261                    ;*****************************************************************************
1262
1263
1264
1265
1266
1267                    ;*****************************************************************************
1268                        TEST 19 - AX3-15,AX3-16 READ/WRITE BIT TEST
1269                    *
1270                    * IN THIS TEST A MASTER CLEAR IS DONE AND THEN A WRITE, READ, AND COMPARE ARE
1271                    * PERFORMED IN REGS AX3-15,AX3-16 USING EACH WORD OF PATTERN V FOR WPITING,
1272                    * AND PATTERN U FOR COMPARING.
1273                    * ANY BITS IN AX3-15,AX3-16 WHICH ARE NOT READ/WRITE ARE MASKED OFF (TO 0)
1274                    * IN THE EXPECTED VALUE BEFORE COMPARISON.
1275                    *   PATTERN V =
1276                    *       FOR REG 15 : 000,333,331,323,313,233,133,000,000,000,000,
1277                    *                    000,000,000,000,000,000,000,000
1278                    *       FOR REG 16 : 000,000,000,000,000,000,000,001,002,004,040,
1279                    *                    100,200,346,345,343,307,247,147
1280                    *   PATTERN U =
1281                    *       FOR REG 15 : 000,001,013,011,021,101,301,000,000,000,000,
1282                    *                    000,000,000,000,000,000,000,000
1283                    *       FOR REG 16 : 000,000,000,000,000,000,000,001,002,004,040,
1284                    *                    100,200,346,345,343,307,247,147
1285                    ;*****************************************************************************
```

```
1286
1287
1288
1289
1290
1291                    ;*****************************************************************************
1292                        TEST 20 - REG 17 - AX2-16 READ/WRITE, MASTER CLEAR TEST
1293                    *
1294                    * THIS TEST CONSISTS OF 2 SUBTESTS. IN THE FIRST SUBTEST, EACH BYTE OF PAT O
1295                    * IS WRITTEN INTO REG 17 AND AFTER EACH WRITE, AX2-16 IS READ AND COMPARED
1296                    * TO A BYTE OF PAT P.
1297                    *   PATTERN O = 000,041,004,010,020,040,100,101,200,201,300,111,301,375
1298                    *   PATTERN P = 000,113,200,040,020,010,001,104,007,105,007,144,107,157
1299                    * IN THE SECOND SUBTEST, REG 17 IS LOADED WITH 375, A MASTER CLEAR IS ISSUED,
1300                    * AND AX2-16 IS COMPARED TO ITS INITIALIZED STATE (103).
1301                    ;*****************************************************************************
1302
1303
1304
1305
1306
1307                    ;*****************************************************************************
1308                        TEST 21 - TRANSMITTER BUFFER DATA TEST
1309                    * A MASTER CLEAR IS DONE FIRST, AND THEN A BYTE OF PATTERN N IS LOADED INTO
1310                    * REG 11 AND THE NEXT BYTE IS LOADED TWICE INTO REG 10. THE PROGRAM THEN WAITS
1311                    * AT LEAST 50 MICRO-SEC, AND THEN IT READS AND COMPARES AX1-15 TO THE BYTE WHICH
1312                    * WAS LOADED INTO REG 10, AND IT READS AND COMPARES AX1-16 TO THE BYTE WHICH WAS
1313                    * LOADED INTO REG 11. THIS PROCESS IS REPEATED (INCLUDING THE MASTER CLEAR)
1314                    * FOR EACH PAIR OF BYTES IN PATTERN N.
1315                    *   PATTERN N =
1316                    *     FOR REG 10: 000,125,252,377,000,000,000
1317                    *     FOR REG 11: 000,000,000,000,005,012,017
1318                    ;*****************************************************************************
1319
1320
1321
1322
1323
1324                    ;*****************************************************************************
1325                        TEST 22 - TRANSMITTER BUFFER SEQUENCING TEST
1326                    *
1327                    * FIRST, A MASTER CLEAR IS DONE, AND THE PROGRAM CHECKS FOR ORDY=1, OCOR=0.
1328                    * THEN, 2 TSOM CHARS ARE LOADED INTO THE TX SILO, AND ALLOWED TO RIPPLE
1329                    * DOWN TO THE OUTPUT. THE PROGRAM CHECKS FOR ORDY=1, OCOR=1.
1330                    * NEXT, THE PROGRAM CYCLES THE STEPLU BIT UNTIL OCOR=0 AGAIN, AND CHECKS FOR
1331                    * THIS TO OCCUR WITHIN 3 CYCLES.
1332                    * THE SILO IS THEN FILLED WITH 64 BYTES OF A 256-BYTE BINARY COUNT PATTERN
1333                    * (000-377) AND THE PROGRAM CHECKS FOR ORDY=0 AFTER THE 64TH CHAR IS LOADED.
1334                    * THE PROGRAM CYCLES STEPLU FOR 8 CYCLES AND CHECKS THAT AFTER THE 8TH, ORDY=1.
1335                    * AX1-15 IS READ AND COMPARED TO EXPECTED DATA.
1336                    * THE REST OF THE BINARY COUNT DATA BYTES ARE LOADED, CYCLED 8 CLOCKS, READ AND
1337                    * COMPARED, A BYTE AT A TIME. UPON COMPLETION, THE SILO IS CHECKED TO BE EMPTY
1338                    * WITH ORDY=1, OCOR=0.
1339                    ;*****************************************************************************
1340
1341
```

```
1342
1343
1344
1345                ;*********************************************************************************
1346                    TEST 23 -  TX MSG TIMING TEST, CHAR MODE, WITH CRC
1347                *
1348                * IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
1349                * AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
1350                * (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
1351                * CHARACTER ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND CRC-16.
1352                * THE FOLLOWING STEPS ARE DONE:
1353                * A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN CHAR MODE.
1354                * SOM IS SET TWICE TO SEND 2 SYNCH CHARS. THEN, 2 000 CHARS ARE SENT, AND
1355                * THEN 2 TERMINATING SYNCHS ARE SENT.
1356                * THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
1357                * CLEARED STATE AFTER THE 3RD SYNCH COMPLETES.
1358                ;*********************************************************************************
1359
1360
1361
1362
1363
1364                ;*********************************************************************************
1365                    TEST 24 - TX MSG TIMING TEST, BIT MODE, WITH CRC
1366                *
1367                * IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
1368                * AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
1369                * (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
1370                * BIT ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND CRC-CCITT-1.
1371                * THE FOLLOWING STEPS ARE DONE:
1372                * A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN BIT MODE.
1373                * SOM IS SET TWICE TO SEND 2 FLAG CHARS. THEN, 2 000 CHARS ARE SENT, AND
1374                * THEN 2 TERMINATING FLAGS ARE SENT.
1375                * THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
1376                * CLEARED STATE.
1377                ;*********************************************************************************
1378
1379
1380
1381
1382
1383                ;*********************************************************************************
1384                    TEST 25 - TX MSG TIMING TEST, CHAR MODE, WITH NO CRC
1385                *
1386                * IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
1387                * AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
1388                * (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
1389                * CHARACTER ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND NO ERROR CHECKING.
1390                * THE FOLLOWING STEPS ARE DONE:
1391                * A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN CHAR MODE.
1392                * SOM IS SET TWICE TO SEND 2 SYNCH CHARS. THEN, 2 000 CHARS ARE SENT, AND
1393                * THEN 2 TERMINATING SYNCHS ARE SENT.
1394                * THE TEST IS PERFORMED WITH TXEN (REG 14, BIT6) SET, AND THE PROGRAM CHECKS
1395                * THAT THIS HOLDS RTS HIGH PAST THE END OF THE MESSAGE.
1396                * THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
1397                * CLEARED STATE.
```

```
1398                    ;*****************************************************************************
1399
1400
1401
1402
1403
1404                    ;*****************************************************************************
1405                        TEST 26 - TX UNDERRUN SET AND CLEAR TEST - CHAR MODE
1406                    *
1407                    * IN THIS TEST, A TX UNDERRUN ERROR IS FORCED IN EACH OF 2 SITUATIONS,
1408                    * AND THEN CLEARED DIFFERENTLY IN EACH.
1409                    * IN THE FIRST, A MESSAGE IS INITIATED, A 000 CHAR IS SENT, AND THE TX
1410                    * BUFFER IS NOT SERVICED IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG,
1411                    * WHICH CAUSES UNRR TO SET IN REG 11. THEN, SOM IS SET TO CLEAR THE ERROR,
1412                    * AND THIS IS VERIFIED.
1413                    * IN THE SECOND SITUATION, A MSG IS INITIATED, A 000 CHAR IS SENT, AND THE TX
1414                    * BUFFER IS NOT SERVICED IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG, WHICH
1415                    * AGAIN CAUSES UNRR TO SET. THEN, A MASTER CLEAR IS DONE, AND THE UNRR BIT
1416                    * IS CHECKED TO BE CLEARED.
1417                    ;*****************************************************************************
1418
1419
1420
1421
1422
1423                    ;*****************************************************************************
1424                        TEST 27 - TRANSMIT CHAR LENGTH TIMING TEST - CHAR MODE, CRC
1425                    *
1426                    * THE LINE UNIT IS PLACED IN CHAR MODE (DDCMP) AND A MESSAGE IS INITIATED
1427                    * WITH AN 8-BIT SYNCH AND A 5-BIT SYNCH CHAR. NEXT, A 000 CHAR IS SENT WITH
1428                    * EACH OF THE FOLLOWING TX CHAR LENGTHS : 5 BITS, 6 BITS, 7 BITS, 8 BITS.
1429                    * (FOR EXAMPLE, A 5-BIT CHAR REQUIRES 5 CLOCK CYCLES TO BE TRANSMITTED). TWO
1430                    * TERMINATING SYNCHS ARE SENT AFTER THE DATA.
1431                    ;*****************************************************************************
1432
1433
1434
1435
1436
1437                    ;*****************************************************************************
1438                        TEST 28 - TRANSMIT CHAR LENGTH TIMING TEST - BIT MODE, CRC
1439                    *
1440                    * THE LINE UNIT IS PLACED IN BIT MODE AND A MESSAGE IS INITIATED
1441                    * WITH 2 FLAG CHARS. NEXT, 2 8-BIT 000 CHARS ARE SENT, FOLLOWED BY 000 CHARS
1442                    * WITH EACH OF THE FOLLOWING TRANSMITTER CHAR LENGTHS:
1443                    *    1 BIT, 2 BITS, 3 BITS, 4 BITS, 5 BITS, 6 BITS, 7 BITS, AND 8 BITS.
1444                    * (FOR EXAMPLE, A 5-BIT CHAR REQUIRES 5 CLOCK CYCLES TO BE TRANSMITTED).
1445                    * TWO TERMINATING FLAGS ARE SENT AFTER THE DATA.
1446                    ;*****************************************************************************
1447
1448
1449
1450
1451
1452                    ;*****************************************************************************
1453                        TEST 29 - TXDATA BIT TEST - CHAR MODE, CRC
```

```
1454                    *
1455                    * THE LINE UNIT IS INITIALIZED AND A MSG IS INITIATED (USING STEPLU) WITH CRC-
1456                    * 16 SELECTED IN CHAR MODE. TWO SYNCHS, 000,125,252,377,000, AND 2 TERMINATING
1457                    * SYNCHS ARE THEN SENT. THE PROGRAM CHECKS EACH BIT OF THE TRANSMITTED
1458                    * DATA CHARS, BY MONITORING TXDATA (REG 17) AS THE DATA IS CLOCKED OUT OF
1459                    * THE USYRT TRANSMITTER.
1460                    ;*****************************************************************************
1461
1462
1463
1464
1465
1466                    ;*****************************************************************************
1467                        TEST 30 - USYRT RECEIVER MSG TEST - CHAR MODE, CRC
1468                    *
1469                    * THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
1470                    * LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH CRC-16
1471                    * SELECTED. TWO SYNCHS, 000,125,252,377,000, AND FOUR TERMINATING SYNCHS ARE
1472                    * SENT. THE PROGRAM MONITORS IACT, AND THE RCV'D CHARS AND CRC BYTES ARE READ
1473                    * FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR IACT
1474                    *  STILL SET AFTER THE MESSAGE.
1475                    ;*****************************************************************************
1476
1477
1478
1479
1480
1481                    ;*****************************************************************************
1482                        TEST 31 - USYRT RECEIVER MSG TEST - BIT MODE, CRC
1483                    *
1484                    * THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
1485                    * LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH CRC-
1486                    * CCITT-1. TWO FLAGS, 000,125,252,377,000, AND TWO TERMINATING FLAGS ARE THEN
1487                    * SENT. THE PROGRAM MONITORS IACT, RSOM, AND THE RCV'D CHARS ARE READ
1488                    * FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR
1489                    * IACT = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT STILL = 0.
1490                    ;*****************************************************************************
1491
1492
1493
1494
1495
1496                    ;*****************************************************************************
1497                        TEST 32 - USYRT RECEIVER MSG TEST - CHAR MODE, NO CRC
1498                    *
1499                    * THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
1500                    * LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH NO
1501                    * ERROR DETECTION. TWO SYNCHS, 000,125,252,377,000, AND TWO SYNCHS ARE
1502                    * THEN SENT. THE PROGRAM MONITORS IACT, AND THE RECEIVED CHARS ARE READ FROM
1503                    * AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR IACT
1504                    * STILL = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT = 0.
1505                    ;*****************************************************************************
1506
1507
1508
1509
```

```
1510
1511                    ;*****************************************************************************
1512                        TEST 33 - USYRT RECEIVER MSG TEST - BIT MODE, NO CRC
1513                    *
1514                    * THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
1515                    * LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH ERROR
1516                    * DETECTION INHIBITED. TWO FLAGS, 000,125,252,377,000, AND TWO TERMINATING FLAGS
1517                    * ARE THEN SENT. THE PROGRAM MONITORS IACT, RSOM, AND THE RCV'D CHARS ARE
1518                    * READ FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR
1519                    * IACT = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT STILL = 0.
1520                    ;*****************************************************************************
1521
1522
1523
1524
1525
1526                    ;*****************************************************************************
1527                        TEST 34 - SILO-DISABLED TRANSMITTER LOAD TEST
1528                    *
1529                    * THIS TEST DISABLES THE SILOS, LOADS A 125 CHARACTER INTO THE TX SILO, AND
1530                    * READS AX1-15 AND CHECKS THAT THE DATA DID NOT GET LOADED INTO THE USYRT TX
1531                    * BUFFER.
1532                    ;*****************************************************************************
1533
1534
1535
1536
1537
1538                    ;*****************************************************************************
1539                        TEST 35 - SILO-DISABLED MESSAGE TEST - BIT MODE, NO CRC
1540                    *
1541                    * THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU)
1542                    * WITH LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, WITH SILO
1543                    * DISABLE SET, AND WITH NO ERROR DETECTION. TWO FLAGS, 000,125,252, AND
1544                    * TERMINATING FLAGS ARE THEN SENT BY LOADING THE TRANSMITTED CHARS INTO
1545                    * REG AX1. THE PROGRAM MONITORS OACT, IACT, RSOM, REOM, ORDY, OCOR, ICIR,
1546                    * IRDY, AND THE RECEIVED CHARS ARE READ FROM AX0 AND COMPARED TO EXPECTED
1547                    * VALUES.
1548                    ;*****************************************************************************
1549
1550
1551
1552
1553
1554                    ;*****************************************************************************
1555                        TEST 36 - RECEIVER BUFFER TEST - CHAR MODE, CRC
1556                    *
1557                    * FIRST, A MASTER CLEAR IS DONE AND THE PROGRAM CHECKS FOR ICIR = 1 AND IRDY
1558                    * = 0. THEN, 2 SOM CHARS ARE LOADED AND CLOCKED INTO THE USYRT, AND 64
1559                    * BYTES OF A 256-BYTE BINARY COUNT DATA PATTERN (000-377) ARE LOADED INTO
1560                    * THE TX SILO.
1561                    * THE LINE UNIT IS THEN CLOCKED UNTIL IRDY = 1, AND THE PROGRAM CHECKS FOR
1562                    * THIS TO OCCUR WITHIN 40-43 CYCLES. THE PROGRAM READS THE RCV SILO, CHECKS THE
1563                    * CHAR FOR 000, AND CHECKS FOR IRDY = 0 AGAIN.
1564                    * THE LINE UNIT IS THEN CLOCKED IN GROUPS OF 8 CYCLES, AND AFTER EACH, THE
1565                    * PROGRAM CHECKS FOR ICIR = 1, IRDY = 1, UNTIL THE 64TH GROUP, AFTER WHICH
```

```
1566                    * IT CHECKS FOR ICIR = 0, IRDY = 1. THE SECOND DATA CHAR IS READ FROM THE
1567                    * RECEIVER SILO AND COMPARED TO 001. THEN, THE PROGRAM CHECKS FOR ICIR = 1,
1568                    * IRDY = 1 AGAIN.
1569                    * THE REST OF THE BINARY COUNT DATA BYTES ARE CYCLED 8 CLOCKS AND READ AND
1570                    * COMPARED A BYTE AT A TIME.
1571                    ;*****************************************************************************
1572
1573
1574
1575
1576
1577                    ;*****************************************************************************
1578                        TEST 37 - RECEIVER CHAR LENGTH TIMING TEST - CHAR MODE, NO CRC
1579                    *
1580                    * THE LINE UNIT IS PLACED IN CHAR MODE, WITH NO ERROR DETECTION, AND A MSG IS
1581                    * INITIATED WITH 2 SYNCH CHARS. NEXT, FIFTEEN 000 CHARS ARE LOADED INTO THE
1582                    * TRANSMITTER SILO. THE LINE UNIT IS THEN CLOCKED USING STEPLU WITH LULOOP
1583                    * SET, WHILE THE RECEIVER CHAR LENGTH IS SET TO THE FOLLOWING VALUES : 5,6,7,8.
1584                    * FOR EACH RCV CHAR LENGTH, THE PROGRAM CHECKS TO MAKE SURE THET USYRT RECEIVER
1585                    * FLAGS OCCUR THE PROPER NO. OF CYCLES APART, FOR EACH RCV CHAR LENGTH.
1586                    * (FOR EXAMPLE A 5-BIT CHAR TAKES 5 CLOCK CYCLES TO BE RECEIVED). A MASTER
1587                    * CLEAR IS THEN DONE TO TERMINATE THE OPERATION.
1588                    ;*****************************************************************************
1589
1590
1591
1592
1593
1594                    ;*****************************************************************************
1595                        TEST 38 - RECEIVER CHAR LENGTH TIMING TEST - BIT MODE, NO CRC
1596                    *
1597                    * THE LINE UNIT IS PLACED IN BIT MODE WITH NO ERROR DETECTION, AND A MESSAGE IS
1598                    * INITIATED  WITH 2 FLAG CHARS. NEXT, FIFTEEN 000 CHARS ARE LOADED INTO THE
1599                    * TRANSMITTER SILO. THE LINE UNIT IS THEN CLOCKED USING STEPLU WITH LULOOP
1600                    * SET, WHILE THE RCV CHAR LENGTH IS SET TO THE FOLLOWING VALUES : 8,8,8,7,6,5,
1601                    * 4,3,2,1. FOR EACH RCV CHAR LENGTH, THE PROGRAM CHECKS TO MAKE SURE THAT THE
1602                    * USYRT RECEIVER FLAGS OCCUR THE PROPER NO. OF CYCLES APART, FOR EACH RCV
1603                    * CHAR LENGTH. (FOR EXAMPLE, A 5 BIT CHAR TAKES 5 CLOCK CYCLES TO BE RECEIVED).
1604                    * A MASTER CLEAR IS THEN DONE TO TERMINATE THE OPERATION.
1605                    ;*****************************************************************************
1606
1607
1608
1609
1610
1611                    ;*****************************************************************************
1612                        TEST 39 - TRANSMITTER UNDERRUN ERROR, IDLE MARKING, CHAR MODE,NO CRC
1613
1614                    * THE LINE UNIT IS PLACED IN CHAR MODE, AND THE IDLE BIT IS SET. THEN, A
1615                    * MSG IS INITIATED, A 000 CHAR IS SENT, AND THE TX BUFFER IS NOT SERVICED
1616                    * IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG, WHICH CAUSES A TX UNDERRUN
1617                    * ERROR. THEN, THE RECEIVER IS CLOCKED AND CHECKED FOR TWO 377 CHARS TO BE
1618                    * RECEIVED (LINE MARKING) BEFORE SHUTTING DOWN WITH A MASTER CLEAR.
1619                    ;*****************************************************************************
1620
1621
```

```
1622
1623
1624
1625                    ;*************************************************************************
1626                        TEST 40 - MSG TERMINATION WITH GA CHARS - BIT MODE,NO CRC
1627                    *
1628                    * THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
1629                    * INITIATED IN BIT MODE.
1630                    * 2 FLAG CHARACTERS ARE SENT, FOLLOWED BY
1631                    * THE FOLLOWING DATA CHARACTERS : 000, 125, 252, 377, 000. THEN THE LOOP
1632                    * MODE BIT (STRIP) IS SET AND 2 TERMINATING GO-AHEAD CHARACTERS ARE
1633                    * SENT. EACH USYRT RCV FLAG IS TIMED AS IT IS RECEIVED, AND THE 5 DATA
1634                    * WORDS ARE READ AND COMPARED TO EXPECTED VALUES.
1635                    * ALSO, THE FIRST GA CHAR IS CHECKED BY SCANNING THE TXDATA BIT AS THE GA
1636                    * IS BEING TRANSMITTED (GA = 376 OCTAL).
1637                    * THE TEST ALSO CHECKS FOR SETTING OF RAB AND EBLK
1638                    * IN LOOP MODE.
1639                    ;*************************************************************************
1640
1641
1642
1643
1644
1645                    ;*************************************************************************
1646                        TEST 41 - IDLE SYNCHS TEST - CHAR MODE
1647                    *
1648                    * THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
1649                    * INITIATED IN CHAR MODE. 24(DEC) SYNCHS ARE SENT.
1650                    * EACH SYNCH IS TIMED AS IT IS RECEIVED, AND THE BITS ARE CHECKED
1651                    * FOR A VALID SYNCH CHAR FOR EACH OF THE 22 SYNCHS WHICH FOLLOW
1652                    * THE FIRST TWO (THESE PERFORM SYNCHRONIZATION, AND ARE NOT READ).
1653                    * WHILE THE LAST SYNCH IS BEING TRANSMITTED, OC IS SET, AND THE
1654                    * NEXT CHAR RCV'D AFTER THE SYNCH IS CHECKED TO BE 377 (LINE MARKING).
1655                    * THEN, A MASTER CLEAR IS ISSUED.
1656                    * THE SYNCH CHAR USED IS 226 (OCTAL).
1657                    ;*************************************************************************
1658
1659
1660
1661
1662
1663                    ;*************************************************************************
1664                        TEST 42 - STRIP SYNCH TEST
1665                    *
1666                    * THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
1667                    * INITIATED IN CHAR MODE AND WITH THE STRIP SYNCH
1668                    * BIT SET. THEN 24 (DEC) SYNCHS ARE SENT
1669                    * FOLLOWED BY THE FOLLOWING DATA CHARACTERS : 377, 000, 125, 252,
1670                    * AND THEN 2 TERMINATING SYNCHS.
1671                    * EACH OF THE 23 SYNCHS AFTER THE FIRST ARE CHECKED AT THE TRANSMITTER OUTPUT,
1672                    * BY SCANNING THE TXDATA BIT.
1673                    * EACH USYRT RCV FLAG IS TIMED AS IT IS RECEIVED, AND THE 4 DATA WORDS
1674                    * ARE READ AND COMPARED TO EXPECTED VALUES.
1675                    * FINALLY, THE LINE UNIT IS CLOCKED FOR SEVERAL CHAR TIMES, AND A CHECK
1676                    * IS MADE FOR OACT = 0 (TEOM SHOULD CAUSE TX ENABLE TO DROP).
1677                    * THE ABOVE TEST IS REPEATED FOR EACH OF THE FOLLOWING SYNCH CHAR DATA
```

```
1678                    * PATTERNS : 226,000,125,252,376,177.
1679                    ;*****************************************************************************
1680
1681
1682
1683
1684
1685                    8.1 DATA PATTERNS USED
1686
1687
1688
1689                    ***** DATA PATTERN A *****
1690                            .BYTE   125
1691                            .BYTE   252
1692                            .BYTE   000
1693                            .BYTE   377
1694                            .BYTE   001
1695                            .BYTE   002
1696                            .BYTE   004
1697                            .BYTE   010
1698                            .BYTE   020
1699                            .BYTE   040
1700                            .BYTE   100
1701                            .BYTE   200
1702                            .BYTE   376
1703                            .BYTE   375
1704                            .BYTE   373
1705                            .BYTE   367
1706                            .BYTE   357
1707                            .BYTE   337
1708                            .BYTE   277
1709                            .BYTE   177
1710
1711                    ***** DATA PATTERN B *****
1712                    PATB:
1713                            .BYTE   000
1714                            .BYTE   000
1715                            .BYTE   040
1716                            .BYTE   100
1717                            .BYTE   220
1718                            .BYTE   000
1719                            .BYTE   000
1720                            .BYTE   051
1721
1722                    ***** DATA PATTERN C *****
1723                    PATC:
1724                            .BYTE   020
1725                            .BYTE   020
1726                            .BYTE   020
1727
1728                    ***** DATA PATTERN D *****
1729                    PATD:
1730                            .BYTE   000
1731                            .BYTE   040
1732                            .BYTE   000
1733
```

```
1734                    ***** DATA PATTERN E *****
1735                    PATE:
1736                            .BYTE   000
1737                            .BYTE   120
1738                            .BYTE   020
1739                            .BYTE   100
1740                            .BYTE   120
1741                            .BYTE   000
1742
1743                    ***** DATA PATTERN F *****
1744                    PATF:
1745                            .BYTE   050
1746                            .BYTE   051
1747                            .BYTF   050
1748
1749                    ***** DATA PATTERN G *****
1750                    PATG:
1751                            .BYTE   000
1752                            .BYTE   000
1753                            .BYTE   240
1754                            .BYTE   120
1755                            .BYTE   177
1756                            .BYTE   000
1757                            .BYTE   000
1758                            .BYTE   001
1759
1760                    ***** DATA PATTERN H *****
1761                    PATH:
1762                            .BYTE   000
1763                            .BYTE   000
1764                            .BYTE   377
1765                            .BYTE   017
1766                            .BYTE   377
1767                            .BYTE   377
1768                            .BYTE   375
1769                            .BYTE   377
1770
1771                    ***** DATA PATTERN I *****
1772                    PATI:
1773                            .BYTE   000
1774                            .BYTE   000
1775                            .BYTE   000
1776                            .BYTE   000
1777                            .BYTE   000
1778                            .BYTE   103
1779                            .BYTE   000
1780                            .BYTE   000
1781
1782                    ***** DATA PATTERN J *****
1783                    PATJ:
1784                            .BYTE   000
1785                            .BYTE   000
1786                            .BYTE   010
1787                            .BYTE   002
1788                            .BYTE   004
1789                            .BYTE   103
```

```
1790                                    .BYTE   001
1791                                    .BYTE   100
1792
1793                            ***** DATA PATTERN K *****
1794                            PATK:   .BYTE   000
1795                                    .BYTE   000
1796                                    .BYTE   377
1797                                    .BYTE   377
1798                                    .BYTE   125
1799                                    .BYTE   125
1800                                    .BYTE   252
1801                                    .BYTE   252
1802                                    .BYTE   001
1803                                    .BYTE   000
1804                                    .BYTE   002
1805                                    .BYTE   000
1806                                    .BYTE   004
1807                                    .BYTE   000
1808                                    .BYTE   010
1809                                    .BYTE   000
1810                                    .BYTE   020
1811                                    .BYTE   000
1812                                    .BYTE   040
1813                                    .BYTE   000
1814                                    .BYTE   100
1815                                    .BYTE   000
1816                                    .BYTE   200
1817                                    .BYTE   000
1818                                    .BYTE   000
1819                                    .BYTE   001
1820                                    .BYTE   000
1821                                    .BYTE   002
1822                                    .BYTE   000
1823                                    .BYTE   004
1824                                    .BYTE   000
1825                                    .BYTE   010
1826                                    .BYTE   000
1827                                    .BYTE   020
1828                                    .BYTE   000
1829                                    .BYTE   040
1830                                    .BYTE   000
1831                                    .BYTE   100
1832                                    .BYTE   000
1833                                    .BYTE   200
1834                                    .BYTE   376
1835                                    .BYTE   377
1836                                    .BYTE   375
1837                                    .BYTE   377
1838                                    .BYTE   373
1839                                    .BYTE   377
1840                                    .BYTE   367
1841                                    .BYTE   377
1842                                    .BYTE   357
1843                                    .BYTE   377
1844                                    .BYTE   337
1845                                    .BYTE   377
```

```
1846                                    .BYTE   277
1847                                    .BYTE   377
1848                                    .BYTE   177
1849                                    .BYTE   377
1850                                    .BYTE   377
1851                                    .BYTE   376
1852                                    .BYTE   377
1853                                    .BYTE   375
1854                                    .BYTE   377
1855                                    .BYTE   373
1856                                    .BYTE   377
1857                                    .BYTE   367
1858                                    .BYTE   377
1859                                    .BYTE   357
1860                                    .BYTE   377
1861                                    .BYTE   337
1862                                    .BYTE   377
1863                                    .BYTE   277
1864                                    .BYTE   377
1865                                    .BYTE   177
1866
1867                                ***** DATA PATTERN L *****
1868                                PATL:
1869                                    .BYTE   000
1870                                    .BYTE   000
1871                                    .BYTE   377
1872                                    .BYTE   377
1873                                    .BYTE   000
1874                                    .BYTE   000
1875
1876                                ***** DATA PATTERN M *****
1877                                PATM:
1878                                    .BYTE   000
1879                                    .BYTE   020
1880                                    .BYTE   000
1881                                    .BYTE   000
1882                                    .BYTE   200
1883                                    .BYTE   000
1884                                    .BYTE   000
1885                                    .BYTE   051
1886
1887                                ***** DATA PATTERN N *****
1888                                PATN:
1889                                    .BYTE   000
1890                                    .BYTE   000
1891                                    .BYTE   000
1892                                    .BYTE   125
1893                                    .BYTE   000
1894                                    .BYTE   252
1895                                    .BYTE   000
1896                                    .BYTE   377
1897                                    .BYTE   005
1898                                    .BYTE   000
1899                                    .BYTE   012
1900                                    .BYTE   000
1901                                    .BYTE   017
```

```
1902                                    .BYTE   000
1903
1904                            ***** DATA PATTERN 0 *****
1905                            PAT0:
1906                                    .BYTE   000
1907                                    .BYTE   041
1908                                    .BYTE   004
1909                                    .BYTE   010
1910                                    .BYTE   020
1911                                    .BYTE   040
1912                                    .BYTE   100
1913                                    .BYTE   101
1914                                    .BYTE   200
1915                                    .BYTE   201
1916                                    .BYTE   300
1917                                    .BYTE   111
1918                                    .BYTE   301
1919                                    .BYTE   375
1920
1921                            ***** DATA PATTERN P *****
1922                            PATP:
1923                                    .BYTE   000
1924                                    .BYTE   113
1925                                    .BYTE   200
1926                                    .BYTE   040
1927                                    .BYTE   020
1928                                    .BYTE   010
1929                                    .BYTE   001
1930                                    .BYTE   104
1931                                    .BYTE   007
1932                                    .BYTE   105
1933                                    .BYTE   007
1934                                    .BYTE   144
1935                                    .BYTE   107
1936                                    .BYTE   157
1937
1938                            ***** DATA PATTERN U *****
1939                            PATU:   .BYTE   000
1940                                    .BYTE   000
1941                                    .BYTE   001
1942                                    .BYTE   000
1943                                    .BYTE   013
1944                                    .BYTE   000
1945                                    .BYTE   011
1946                                    .BYTE   000
1947                                    .BYTE   021
1948                                    .BYTE   000
1949                                    .BYTE   101
1950                                    .BYTE   000
1951                                    .BYTE   301
1952                                    .BYTE   000
1953                                    .BYTE   000
1954                                    .BYTE   001
1955                                    .BYTE   000
1956                                    .BYTE   002
1957                                    .BYTE   000
```

```
1958                                    .BYTE   004
1959                                    .BYTE   000
1960                                    .BYTE   040
1961                                    .BYTE   000
1962                                    .BYTE   100
1963                                    .BYTE   000
1964                                    .BYTE   200
1965                                    .BYTE   000
1966                                    .BYTE   346
1967                                    .BYTE   000
1968                                    .BYTE   345
1969                                    .BYTE   000
1970                                    .BYTE   343
1971                                    .BYTE   000
1972                                    .BYTE   307
1973                                    .BYTE   000
1974                                    .BYTE   247
1975                                    .BYTE   000
1976                                    .BYTE   147
1977
1978                            ***** DATA PATTERN V *****
1979                            PATV:   .BYTE   000
1980                                    .BYTE   000
1981                                    .BYTE   333
1982                                    .BYTE   000
1983                                    .BYTE   331
1984                                    .BYTE   000
1985                                    .BYTE   323
1986                                    .BYTE   000
1987                                    .BYTE   313
1988                                    .BYTE   000
1989                                    .BYTE   233
1990                                    .BYTE   000
1991                                    .BYTE   133
1992                                    .BYTE   000
1993                                    .BYTE   000
1994                                    .BYTE   001
1995                                    .BYTE   000
1996                                    .BYTE   002
1997                                    .BYTE   000
1998                                    .BYTE   004
1999                                    .BYTE   000
2000                                    .BYTE   040
2001                                    .BYTE   000
2002                                    .BYTE   100
2003                                    .BYTE   000
2004                                    .BYTE   200
2005                                    .BYTE   000
2006                                    .BYTE   346
2007                                    .BYTE   000
2008                                    .BYTE   345
2009                                    .BYTE   000
2010                                    .BYTE   343
2011                                    .BYTE   000
2012                                    .BYTE   307
2013                                    .BYTE   000
```

```
2014                                    .BYTE   247
2015                                    .BYTE   000
2016                                    .BYTE   147
2017
2018
2019
2020
2021
2022
```

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 :

```
  CZDMR 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 :

```
  CZDMR 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
```

```
2079
2080
2081
2082                                    a
2083
2084
2085
2086
2087
```

```
                                .TITLE CZDMRF M8203 STATIC DIAG #1
002000                                  .=2000
                                .MCALL  SVC
002000                          SVC                             ; INITIALIZE SUPERVISOR MACROS
002000                          BGNMOD  LU1MOD
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 BGNAU,BGNDU

                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        ;       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".
                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

002000                          HEADER  CZDMR,F,0,45.,0
002000                                                          L$NAME::
002000     103                                                          .ASCII /C/
002001     132                                                          .ASCII /Z/
002002     104                                                          .ASCII /D/
002003     115                                                          .ASCII /M/
002004     122                                                          .ASCII /R/
002005     000                                                          .BYTE   0
002006     000                                                          .BYTE   0
002007     000                                                          .BYTE   0
002010                                                          L$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  035500                                                          .WORD   L$HARD
002020                                                          L$SPCP::
002020  000000                                                          .WORD   0
002022                                                          L$HPTP::
002022  002252                                                          .WORD   L$HW
002024                                                          L$SPTP::
002024  000000                                                          .WORD   0
002026                                                          L$LADP::
002026  036054                                                          .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$DEVP::
002060  003440                  .WORD   L$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  021556                  .WORD   L$AU
002072                  L$DUT::
002072  021474                  .WORD   L$DU
002074                  L$LUN::
002074  000000                  .WORD   0
002076                  L$DESP::
002076  003446                  .WORD   L$DESC
002100                  L$LOAD::
002100  104035                  EMT     E$LOAD
002102                  L$ETP::
002102  000000                  .WORD   0
002104                  L$ICP::
002104  021074                  .WORD   L$INIT
002106                  L$CCP::
002106  021472                  .WORD   L$CLEAN
002110                  L$ACP::
002110  021412                  .WORD   L$AUTO
002112                  L$PRT::
002112  021066                  .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
```

```
                                .SBTTL  DISPATCH TABLE

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

002122                                  DISPATCH 42
002122  000052                                                          .WORD   42
002124                                                          L$DISPATCH::
002124  021560                                                          .WORD   T1
002126  021642                                                          .WORD   T2
002130  021726                                                          .WORD   T3
002132  022052                                                          .WORD   T4
002134  022154                                                          .WORD   T5
002136  022302                                                          .WORD   T6
002140  022422                                                          .WORD   T7
002142  022576                                                          .WORD   T8
002144  023032                                                          .WORD   T9
002146  023150                                                          .WORD   T10
002150  023266                                                          .WORD   T11
002152  023404                                                          .WORD   T12
002154  023522                                                          .WORD   T13
002156  024152                                                          .WORD   T14
002160  024504                                                          .WORD   T15
002162  024774                                                          .WORD   T16
002164  025204                                                          .WORD   T17
002166  025410                                                          .WORD   T18
002170  025614                                                          .WORD   T19
002172  026052                                                          .WORD   T20
002174  026304                                                          .WORD   T21
002176  026556                                                          .WORD   T22
002200  027234                                                          .WORD   T23
002202  027354                                                          .WORD   T24
002204  027464                                                          .WORD   T25
002206  027660                                                          .WORD   T26
002210  030224                                                          .WORD   T27
002212  030434                                                          .WORD   T28
002214  030764                                                          .WORD   T29
002216  031064                                                          .WORD   T30
002220  031434                                                          .WORD   T31
002222  031776                                                          .WORD   T32
002224  032246                                                          .WORD   T33
002226  032610                                                          .WORD   T34
002230  032740                                                          .WORD   T35
002232  033652                                                          .WORD   T36
002234  034134                                                          .WORD   T37
002236  034334                                                          .WORD   T38
002240  034634                                                          .WORD   T39
002242  034724                                                          .WORD   T40
002244  035066                                                          .WORD   T41
002246  035234                                                          .WORD   T42

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;       CHANGE THE ARGUMENT OF 'DISPATCH'' TO BE THE
                                ;       NUMBER OF HARDWARE TESTS IN YOUR PROGRAM.
```

CZDMRF.P11 03-NOV-81 10:29 DISPATCH TABLE

```
2288                    ;%%%
```

```
CZDMRF.P11    03-NOV-81 10:29          DEFAULT HARDWARE P-TABLE

  2294                                 .SBTTL  DEFAULT HARDWARE P-TABLE
  2295
  2296                                 ;////////////////////////////////////////////////////////////////////////////
  2297                                 ;/ THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
  2298                                 ;/ THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
  2299                                 ;/ IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
  2300                                 ;////////////////////////////////////////////////////////////////////////////
  2301
  2302  002250                                 BGNHW   DFPTBL
  2303  002250  000002                                                                  .WORD   L10000-L$HW/2
  2304  002252                                                                 L$HW::
  2305  002252                                                                 DFPTBL::
  2306
  2307  002252  160170                        .WORD   160170          ;M8207 CSR UNIBUS ADDRESS
  2308  002254  000001                        .WORD        1          ;M8207 RUN SWITCH (E28 SW7) IS ON
  2309
  2310  002256                                 ENDHW
  2311  002256                                                                 L10000:
  2312
  2313
  2314
  2315
  2316
```

```
CZDMRF.P11    03-NOV-81 10:29          SOFTWARE P-TABLE

  2317                                 .SBTTL  SOFTWARE P-TABLE
  2318
  2319                                 ;////////////////////////////////////////////////////////////////////////
  2320                                 ;/ THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
  2321                                 ;/ PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
  2322                                 ;////////////////////////////////////////////////////////////////////////
  2323
  2324  002256                             BGNSW   SFPTBL
  2325  002256  000000                                                          .WORD   L10001-L$SW/2
  2326  002260                                                         L$SW::
  2327  002260                                                         SFPTBL::
  2328
  2329
  2330  002260                             ENDSW
  2331  002260                                                         L10001:
  2332
  2333
  2334
  2335
  2336
  2337
```

```
                                .SBTTL  GLOBAL EQUATES SECTION

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

002260                                  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
```

```
2394    000036          EF.CONTINUE==   30.             ; CONTINUE COMMAND WAS ISSUED
2395    000035          EF.NEW==        29.             ; A NEW PASS HAS BEEN STARTED
2396    000034          EF.PWR==        28.             ; A POWER-FAIL/POWER-UP OCCURRED
2397                    ;
2398                    ;
2399                    ; PRIORITY LEVEL DEFINITIONS
2400                    ;
2401    000340          PRI07== 340
2402    000300          PRI06== 300
2403    000240          PRI05== 240
2404    000200          PRI04== 200
2405    000140          PRI03== 140
2406    000100          PRIC2== 100
2407    000040          PRI01== 40
2408    000000          PRI00== 0
2409                    ;
2410                    ;OPERATOR FLAG BITS
2411                    ;
2412    000004          EVL==       4
2413    000010          LOT==      10
2414    000020          ADR==      20
2415    000040          IDU==      40
2416    000100          ISR==     100
2417    000200          UAM==     200
2418    000400          BOE==     400
2419    001000          PNT==    1000
2420    002000          PRI==    2000
2421    004000          IXE==    4000
2422    010000          IBE==   10000
2423    020000          IER==   20000
2424    040000          LOE==   40000
2425    100000          HOE==  100000
2426
2427
2428
2429
2430                    ;;*************************************************************************
2431                    ;* PROGRAM EVENT FLAG DEFINITIONS
2432                    ;;*************************************************************************
2433
2434
2435
2436
2437
2438                    ;;*************************************************************************
2439                    ;* MAINTENANCE REGISTER - BSEL1
2440                    ;;*************************************************************************
2441    000200          RUN     = BIT7
2442    000100          MCLR    = BIT6
2443    000020          STEPLU  = BIT4
2444    000010          LULOOP  = BIT3
2445    000004          ROMO    = BIT2
2446    000002          ROMI    = BIT1
2447    000001          STEPMP  = BIT0
2448
2449                    ;;*************************************************************************
```

```
2450
2451                                    ;* OBJS REG 10 - TRANSMITTER BUFFER
2452                                    ;;*****************************************************************************
2453            000200                  TX7     = BIT7
2454            000100                  TX6     = BIT6
2455            000040                  TX5     = BIT5
2456            000020                  TX4     = BIT4
2457            000010                  TX3     = BIT3
2458            000004                  TX2     = BIT2
2459            000002                  TX1     = BIT1
2460            000001                  TX0     = BIT0
2461
2462                                    ;;*****************************************************************************
2463                                    ;* OBUS REG 11
2464                                    ;;*****************************************************************************
2465            000200                  OC      = BIT7
2466            000010                  GOAH    = BIT3
2467            000004                  ABORT   = BIT2
2468            000002                  EOM     = BIT1
2469            000001                  SOM     = BIT0
2470
2471                                    ;;*****************************************************************************
2472                                    ;* OBUS REG 12
2473                                    ;;*****************************************************************************
2474            000200                  IC      = BIT7
2475            000100                  BPOLL   = BIT6
2476            000040                  LULP    = BIT5
2477
2478                                    ;;*****************************************************************************
2479                                    ;* CBUS REG 13
2480                                    ;;*****************************************************************************
2481            000200                  POLL    = BIT7
2482            000100                  DTR     = BIT6
2483            000040                  SELFR   = BIT5
2484            000020                  HDX     = BIT4
2485            000010                  MAINT1  = BIT3
2486            000004                  MAINT2  = BIT2
2487            000002                  SELSBY  = BIT1
2488
2489                                    ;;*****************************************************************************
2490                                    ;* OBUS REG 14
2491                                    ;;*****************************************************************************
2492            000100                  TXEN    = BIT6
2493            000040                  DISSI   = BIT5
2494            000020                  RDAX    = BIT4
2495            000010                  WAX     = BIT3
2496            000004                  ENAX    = BIT2
2497            000002                  AX2     = BIT1
2498            000001                  AX1     = BIT0
2499
2500                                    ;;*****************************************************************************
2501                                    ;* OBUS REG 17
2502                                    ;;*****************************************************************************
2503            000200                  CRC2    = BIT7
2504            000100                  CRC1    = BIT6
2505            000040                  IDLE    = BIT5
2506            000020                  SECA    = BIT4
```

```
2506    000010          STRIP   = BIT3
2507    000004          RDALL   = BIT2
2508    000002          IERR    = BIT1
2509    000001          DDCMP   = BIT0
2510
2511                    ;;*****************************************************************************
2512                    ;* IBUS REG 10 - RECEIVER BUFFER
2513                    ;;*****************************************************************************
2514    000200          RX7     = BIT7
2515    000100          RX6     = BIT6
2516    000040          RX5     = BIT5
2517    000020          RX4     = BIT4
2518    000010          RX3     = BIT3
2519    000004          RX2     = BIT2
2520    000002          RX1     = BIT1
2521    000001          RX0     = BIT0
2522
2523                    ;;*****************************************************************************
2524                    ;* IBUS REG 11
2525                    ;;*****************************************************************************
2526    000200          OC      = BIT7
2527    000100          OACT    = BIT6
2528    000040          SW3     = BIT5
2529    000020          ORDY    = BIT4
2530    000010          SW2     = BIT3
2531    000004          SW1     = BIT2
2532    000002          SW0     = BIT1
2533    000001          UNRR    = BIT0
2534
2535                    ;;*****************************************************************************
2536                    ;* IBUS REG 12
2537                    ;;*****************************************************************************
2538    000200          IC      = BIT7
2539    000100          IACT    = BIT6
2540    000040          LULP    = BIT5
2541    000020          IRDY    = BIT4
2542    000010          OVRR    = BIT3
2543    000004          RAB     = BIT2
2544    000002          EBLK    = BIT1
2545    000001          BCC     = BIT0
2546
2547                    ;;*****************************************************************************
2548                    ;* IBUS REG 13
2549                    ;;*****************************************************************************
2550    000200          RING    = BIT7
2551    000100          DTR     = BIT6
2552    000040          RTS     = BIT5
2553    000020          HDX     = BIT4
2554    000010          MODR    = BIT3
2555    000004          CS      = BIT2
2556    000002          STBY    = BIT1
2557    000001          CARR    = BIT0
2558
2559                    ;;*****************************************************************************
2560                    ;* IBUS REG 14
2561                    ;;*****************************************************************************
```

```
2562    000200          READY   = BIT7
2563    000100          TXEN    = BIT6
2564    000040          DISSI   = BIT5
2565    000020          RDAX    = BIT4
2566    000010          WAX     = BIT3
2567    000004          ENAX    = BIT2
2568    000002          AX2     = BIT1
2569    000001          AX1     = BIT0
2570
2571                    ;;*****************************************************************************
2572                    ;* IBUS REG 17
2573                    ;;*****************************************************************************
2574    000200          SIGR    = BIT7
2575    000100          SIGQ    = BIT6
2576    000040          TXDATA  = BIT5
2577    000020          OCOR    = BIT4
2578    000010          ICIR    = BIT3
2579    000004          TESTMD  = BIT2
2580    000002          MCLK    = BIT1
2581    000001          DDCMP   = BIT0
2582
2583                    ;;*****************************************************************************
2584                    ;* AX0-15 - USYRT REG 0 (READ ONLY)
2585                    ;;*****************************************************************************
2586    000200          RX7     = BIT7
2587    000100          RX6     = BIT6
2588    000040          RX5     = BIT5
2589    000020          RX4     = BIT4
2590    000010          RX3     = BIT3
2591    000004          RX2     = BIT2
2592    000002          RX1     = BIT1
2593    000001          RX0     = BIT0
2594
2595                    ;;*****************************************************************************
2596                    ;* AX0-16 - USYRT REG 1 (READ ONLY)
2597                    ;;*****************************************************************************
2598    000200          RERR    = BIT7
2599    000100          ASBC2   = BIT6
2600    000040          ASBC1   = BIT5
2601    000020          ASBC0   = BIT4
2602    000010          ROR     = BIT3
2603    000004          RABT    = BIT2
2604    000002          REOM    = BIT1
2605    000001          RSOM    = BIT0
2606
2607                    ;;*****************************************************************************
2608                    ;* AX1-15 - USYRT REG 2
2609                    ;;*****************************************************************************
2610    000200          TX7     = BIT7
2611    000100          TX6     = BIT6
2612    000040          TX5     = BIT5
2613    000020          TX4     = BIT4
2614    000010          TX3     = BIT3
2615    000004          TX2     = BIT2
2616    000002          TX1     = BIT1
2617    000001          TX0     = BIT0
```

```
2618
2619                        ;;*****************************************************************************
2620                        ;* AX1-16 - USYRT REG 3
2621                        ;;*****************************************************************************
2622    000200              TERR    = BIT7
2623    000010              TXGA    = BIT3
2624    000004              TXAB    = BIT2
2625    000002              TEOM    = BIT1
2626    000001              TSOM    = BIT0
2627
2628                        ;;*****************************************************************************
2629                        ;* AX2-15 - USYRT REG 4
2630                        ;;*****************************************************************************
2631    000200              SYN7    = BIT7
2632    000100              SYN6    = BIT6
2633    000040              SYN5    = BIT5
2634    000020              SYN4    = BIT4
2635    000010              SYN3    = BIT3
2636    000004              SYN2    = BIT2
2637    000002              SYN1    = BIT1
2638    000001              SYN0    = BIT0
2639    000226              SYNCH   = 226
2640
2641                        ;;*****************************************************************************
2642                        ;* AX2-16 - USYRT REG 5
2643                        ;;*****************************************************************************
2644    000200              APA     = BIT7
2645    000100              DDC     = BIT6
2646    000040              STR     = BIT5
2647    000020              SEC     = BIT4
2648    000010              IDL     = BIT3
2649    000004              CRCTY2  = BIT2
2650    000002              CRCTY1  = BIT1
2651    000001              CRCTY0  = BIT0
2652
2653                        ;;*****************************************************************************
2654                        ;* AX3-15 - USYRT REG 6
2655                        ;;*****************************************************************************
2656    000200              I422    = BIT7
2657    000100              XYZ     = BIT6
2658    000040              C32BCC  = BIT5
2659    000020              V35     = BIT4
2660    000010              INTGRL  = BIT3
2661    000004              C32ENB  = BIT2
2662    000002              OP      = BIT1
2663    000001              TEST    = BIT0
2664    000372              AX315U  = I422!XYZ!C32BCC!V35!INTGRL!OP
2665
2666                        ;;*****************************************************************************
2667                        ;* AX3-16 - USYRT REG 7
2668                        ;;*****************************************************************************
2669    000200              TXLEN2  = BIT7
2670    000100              TXLEN1  = BIT6
2671    000040              TXLEN0  = BIT5
2672    000004              RXLEN2  = BIT2
2673    000002              RXLEN1  = BIT1
```

```
CZDMRF.P11    03-NOV-81 10:29          GLOBAL EQUATES SECTION

 2674           000001                RXLEN0  = BIT0
 2675
 2676
 2677
 2678
 2679
 2680                                 ;*****************************************************************************
 2681                                 ;* TX CONTROL BITS DEFINED ON WORD BASIS
 2682                                 ;*****************************************************************************
 2683           004000                TXGOA   = BIT11
 2684           002000                TXABT   = BIT10
 2685           001000                TXEOM   = BIT9
 2686           000400                TXSOM   = BIT8
 2687
 2688
 2689
 2690
 2691
 2692                                 ;*****************************************************************************
 2693                                 ;* RCV CONTROL BITS DEFINED ON WORD BASIS
 2694                                 ;*****************************************************************************
 2695           004000                RXOVR   = BIT11
 2696           002000                RXABT   = BIT10
 2697           001000                RXEBL   = BIT9
 2698           000400                RXBCC   = BIT8
 2699
 2700
 2701
 2702
 2703                                 ;*****************************************************************************
 2704                                 ;* ADDRESS EQUATES FOR REGISTER STORAGE TABLE (LUREG:)
 2705                                 ;*****************************************************************************
 2706           002260                LUR10   = LUREG+0       ;LINE UNIT IBUS REG 10
 2707           002262                LUR11   = LUREG+2       ;LINE UNIT IBUS REG 11
 2708           002264                LUR12   = LUREG+4       ;LINE UNIT IBUS REG 12
 2709           002266                LUR13   = LUREG+6       ;LINE UNIT IBUS REG 13
 2710           002270                LUR14   = LUREG+10      ;LINE UNIT IBUS REG 14
 2711           002272                LUR15   = LUREG+12      ;LINE UNIT IBUS REG 15
 2712           002274                LUR16   = LUREG+14      ;LINE UNIT IBUS REG 16
 2713           002276                LUR17   = LUREG+16      ;LINE UNIT IBUS REG 17
 2714           002300                AX0.15  = LUREG+20      ;USYRT REG 0
 2715           002302                AX0.16  = LUREG+22      ;USYRT REG 1
 2716           002304                AX1.15  = LUREG+24      ;USYRT REG 2
 2717           002306                AX1.16  = LUREG+26      ;USYRT REG 3
 2718           002310                AX2.15  = LUREG+30      ;USYRT REG 4
 2719           002312                AX2.16  = LUREG+32      ;USYRT REG 5
 2720           002314                AX3.15  = LUREG+34      ;USYRT REG 6
 2721           002316                AX3.16  = LUREG+36      ;USYRT REG 7
 2722
 2723
 2724
 2725
 2726
 2727           100000                CHPCHK  = BIT15
 2728
 2729           100000                BCCCHK  = BIT15
```

```
CZDMRF.P11    03-NOV-81 10:29          GLOBAL EQUATES SECTION

 2730           100000                CRCCHK  = BIT15
 2731
 2732
 2733
 2734
 2735
 2736                                 ;*************************************************************************************
 2737                                 ;* MICROINSTRUCTION DEFINITIONS
 2738                                 ;*************************************************************************************
 2739           021000                MVIOX   = 021000                ;MOVE IBUS TO OBUS*
 2740           122000                MVIXO   = 122000                ;MOVE IBUS* TO OBUS
 2741           121000                MVIXOX  = 121000                ;MOVE IBUS* TO OBUS*
 2742
 2743
 2744
 2745                                 ;***** ERROR1 BIT FLAG DEFINITIONS *****
 2746           000001                RRDYTO  = BIT0
 2747           000002                WRDYTO  = BIT1
 2748
 2749
 2750
 2751
 2752
```

```
2753                            .SBTTL  GLOBAL DATA SECTION
2754
2755                            ;////////////////////////////////////////////////////////////////////////////////
2756                            ;/      THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
2757                            ;/      IN MORE THAN ONE TEST.
2758                            ;////////////////////////////////////////////////////////////////////////////////
2759
2760                            ;;*****************************************************************************
2761                            ;* STORAGE FOR DEVICE REGISTERS
2762                            ;;*****************************************************************************
2763  002260  000020            LUREG:  .BLKW   16.
2764
2765                            ;*****************************************************************************
2766                            ;* MISCELLANEOUS STORAGE
2767                            ;*****************************************************************************
2768  002320  000000            SCRACH: .WORD   0               ;GEN'L PURPOSE SCRATCH WORD
2769  002322  000000            LOGDEV: .WORD   0               ;LOGICAL DEVICE NUMBER
2770  002324  000000            PSTACK: .WORD   0               ;CONTAINS BASE LEVEL PROGRAM STACK POINTER
2771  002326  000000            PRIOR:  .WORD   0               ;CPU PRIORITY FOR PRINTOUT
2772  002330  000000            SUBRPC: .WORD   0               ;PC OF SUBR CALL FOR ERROR REPORTS
2773  002332  000000            INTFLG: .WORD   0               ;INTERRUPT RECEIVED FLAGS
2774                                                            ;  BIT 0 FOR TX, BIT 1 FOR RCV
2775  002334  000000            ERRFLG: .WORD   0               ;SUBROUTINE ERROR FLAG
2776  002336  000000            TIMFLG: .WORD   0               ;EVENT TIME-OUT FLAG
2777  002340  000000            RETADR: .WORD   0               ;SUBR ERROR RETURN ADDRESS
2778  002342  000000            REDBYT: .WORD   0               ;LO BYTE CONTAINS BYTE READ FROM LU REG
2779  002344  000000            WRIBYT: .WORD   0               ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
2780  002346  000000            RAX15:  .WORD   0               ;LO BYTE CONTAINS BYTE READ FROM REG 15
2781  002350  000000            RAX16:  .WORD   0               ;LO BYTE CONTAINS BYTE READ FROM REG 16
2782  002352  000000            WAX15:  .WORD   0               ;LO BYTE CONTAINS BYTE TO LOAD INTO REG 15
2783  002354  000000            WAX16:  .WORD   0               ;LO BYTE CONTAINS BYTE TO LOAD INTO REG 16
2784  002356  000000            REGNUM: .WORD   0               ;NUMBER (10-17) OF LINE UNIT REG BEING TESTED
2785  002360  000000            AXNUM:  .WORD   0               ;NUMBER (0-7) OF EXTENDED REG BYTE BEING TESTED
2786  002362  000000            GOODAT: .WORD   0               ;STORAGE FOR EXPECTED DATA
2787  002364  000000            BADDAT: .WORD   0               ;STORAGE FOR ACTUAL DATA
2788  002366  000000            LOADAT: .WORD   0               ;CONTAINS TEST DATA LOADED INTO REG
2789  002370  000000            FRSTIM: .WORD   0               ;FLAG=0 IF PROGRAM JUST LOADED
2790  002372  000000            SAVE4:  .WORD   0               ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
2791  002374  000000            SAVE6:  .WORD   0               ;SAVE LOC 6 HERE (ERROR TRAP VECTOR)
2792  002376  000000            ERROR1: .WORD   0               ;SUBR ERROR BIT FLAGS (DEF'D IN GLOBAL EQUATES)
2793  002400  000000            TXWORD: .WORD   0               ;BITS 0-11 CONTAIN DATA TO LOAD INTO TX SILO
2794  002402  000000            RXWORD: .WORD   0               ;BITS 0-11 CONTAIN DATA READ FROM RCV SILO
2795  002404  000000            DISILO: .WORD   0               ;CONTAINS CURRENT STATE OF DISSI IN BIT 5
2796  002406  000000            CHPTYP: .WORD   0               ;USYRT CHIP TYPE, =0 FOR SIG, ELSE =1
2797  002410  000000            SAVLEN: .WORD   0               ;SAVED TX AND RCV CHAR LENGTHS
2798  002412  000000            DEVMAP: .WORD   0               ;BIT MAP OF ACTIVE DEVICES
2799  002414  000000            DEVPTR: .WORD   0               ;DEVICE MAP BIT POINTER
2800  002416  000000            UNIT:   .WORD   0               ;CONTAINS UNIT NO. (1 TO N)
2801  002420  000000            TSTNUM: .WORD   0               ;CONTAINS TEST NUMBER FOR SOME TESTS
2802  002422  000000            STARES: .WORD   0               ;FLAG=0 IF FIRST PASS AFTER STA OR RES
2803
2804                            ;;*********  CURRENT DEVICE PARAMETERS  *************
2805  002424  160170            MPCSR:  .WORD   160170          ;POINTER TO MICROPROCESSOR CSR'S
2806  002426  160171            BSEL1:  .WORD   160171          ;POINTER TO BSEL1
2807  002430  160172            BSEL2:  .WORD   160172          ;POINTER TO BSEL2
2808  002432                    BSEL4:
```

```
2809  002432  160174          SEL4:    .WORD   160174          ;POINTER TO SEL4
2810  002434  160176          SEL6:    .WORD   160176          ;POINTER TO SEL6
2811  002436  000300          MPIVEC:  .WORD      300          ;MICROPROCESSOR INPUT INTERRUPT VECTOR
2812  002440  000304          MPOVEC:  .WORD      304          ;MICROPROCESSOR OUTPUT INTERRUPT VECTOR
2813  002442  000240          MPRIOR:  .WORD      240          ;MICROPROCESSOR DEVICE PRIORITY
2814  002444  000000          LUSWI1:  .WORD        0          ;LINE UNIT SWITCH PACK #1
2815  002446  000000          LUSWI2:  .WORD        0          ;LINE UNIT SWITCH PACK #2
2816  002450  000000          LUSWI3:  .WORD        0          ;LINE UNIT SWITCH PACK #3
2817  002452  000000          TSTCON:  .WORD        0          ;TEST CONNECTOR INDICATOR
2818  002454  000000          RUNINH:  .WORD        0          ;RUN SWITCH INDICATOR
2819
2820                          ;***** STORAGE FOR DATA READ IN ADDRESS TESTS *****
2821  002456      000         REDDAT:  .BYTE    0
2822  002457      000                  .BYTE    0
2823  002460      000                  .BYTE    0
2824  002461      000                  .BYTE    0
2825  002462      000                  .BYTE    0
2826  002463      000                  .BYTE    0
2827  002464      000                  .BYTE    0
2828  002465      000                  .BYTE    0
2829
2830                          ;;****  GEN'L PURPOSE SCRATCH STORAGE  *****
2831  002466  000000          REG0:    .WORD    0
2832  002470  000000          REG1:    .WORD    0
2833  002472  000000          REG2:    .WORD    0
2834  002474  000000          REG3:    .WORD    0
2835  002476  000000          REG4:    .WORD    0
2836  002500  000000          REG5:    .WORD    0
2837  002502  000000          REG6:    .WORD    0
2838  002504  000000          REG7:    .WORD    0
2839
2840                          ;;****  SCRATCH STORAGE FOR MESSAGE REPORTING   *****
2841  002506  000000          TMP0:    .WORD    0
2842  002510  000000          TMP1:    .WORD    0
2843  002512  000000          TMP2:    .WORD    0
2844  002514  000000          TMP3:    .WORD    0
2845  002516  000000          TMP4:    .WORD    0
2846  002520  000000          TMP5:    .WORD    0
2847  002522  000000          TMP6:    .WORD    0
2848  002524  000000          TMP7:    .WORD    0
2849
2850                          ;***** INBUS LU REG BIT MASKS FOR UNPREDICTABLE BITS *****
2851  002526                  UPBITS:
2852  002526      377                  .BYTE    377             ;MASK FOR REG 10
2853  002527      056                  .BYTE    056             ;MASK FOR REG 11
2854  002530      017                  .BYTE    017             ;MASK FOR REG 12
2855  002531      257                  .BYTE    257             ;MASK FOR REG 13
2856  002532      100                  .BYTE    100             ;MASK FOR REG 14
2857  002533      377                  .BYTE    377             ;MASK FOR REG 15
2858  002534      377                  .BYTE    377             ;MASK FOR REG 16
2859  002535      306                  .BYTE    306             ;MASK FOR REG 17
2860
2861  002536      200         R14NRW:  .BYTE    200             ;REG 14 NON-R/W BITS
2862
2863                          ;***** MASKS FOR EXTENDED REGISTER NON-READ/WRITE BITS *****
2864  002537                  ANBITS:
```

```
2865  002537  377                .BYTE   377             ;MASK FOR AX0-15
2866  002540  377                .BYTE   377             ;MASK FOR AX0-16
2867  002541  000                .BYTE   000             ;MASK FOR AX1-15
2868  002542  360                .BYTE   360             ;MASK FOR AX1-16
2869  002543  000                .BYTE   000             ;MASK FOR AX2-15
2870  002544  000                .BYTE   000             ;MASK FOR AX2-16
2871  002545  004                .BYTE   004             ;MASK FOR AX3-15
2872  002546  030                .BYTE   030             ;MASK FOR AX3-16
2873
2874                             ;***** DATA PATTERN A *****
2875  002547                     PATA:
2876  002547  125                .BYTE   125
2877  002550  252                .BYTE   252
2878  002551  000                .BYTE   000
2879  002552  377                .BYTE   377
2880  002553  001                .BYTE   001
2881  002554  002                .BYTE   002
2882  002555  004                .BYTE   004
2883  002556  010                .BYTE   010
2884  002557  020                .BYTE   020
2885  002560  040                .BYTE   040
2886  002561  100                .BYTE   100
2887  002562  200                .BYTE   200
2888  002563  376                .BYTE   376
2889  002564  375                .BYTE   375
2890  002565  373                .BYTE   373
2891  002566  367                .BYTE   367
2892  002567  357                .BYTE   357
2893  002570  337                .BYTE   337
2894  002571  277                .BYTE   277
2895  002572  177                .BYTE   177
2896
2897                             ;***** DATA PATTERN B *****
2898  002573                     PATB:
2899  002573  000                .BYTE   000
2900  002574  000                .BYTE   000
2901  002575  040                .BYTE   040
2902  002576  100                .BYTE   100
2903  002577  220                .BYTE   220
2904  002600  000                .BYTE   000
2905  002601  000                .BYTE   000
2906  002602  051                .BYTE   051
2907
2908                             ;***** DATA PATTERN C *****
2909  002603                     PATC:
2910  002603  020                .BYTE   020
2911  002604  020                .BYTE   020
2912  002605  020                .BYTE   020
2913
2914                             ;***** DATA PATTERN D *****
2915  002606                     PATD:
2916  002606  000                .BYTE   000
2917  002607  040                .BYTE   040
2918  002610  000                .BYTE   000
2919
2920                             ;***** DATA PATTERN E *****
```

CZDMRF.P11 03-NOV-81 10:29 GLOBAL DATA SECTION

```
2921  002611                        PATE:
2922  002611  000                           .BYTE   000
2923  002612  120                           .BYTE   120
2924  002613  020                           .BYTE   020
2925  002614  100                           .BYTE   100
2926  002615  120                           .BYTE   120
2927  002616  000                           .BYTE   000
2928
2929                                ;***** DATA PATTERN F *****
2930  002617                        PATF:
2931  002617  050                           .BYTE   050
2932  002620  051                           .BYTE   051
2933  002621  050                           .BYTE   050
2934
2935                                ;***** DATA PATTERN G *****
2936  002622                        PATG:
2937  002622  000                           .BYTE   000
2938  002623  000                           .BYTE   000
2939  002624  240                           .BYTE   240
2940  002625  120                           .BYTE   120
2941  002626  177                           .BYTE   177
2942  002627  000                           .BYTE   000
2943  002630  000                           .BYTE   000
2944  002631  001                           .BYTE   001
2945
2946                                ;***** DATA PATTERN H *****
2947  002632                        PATH:
2948  002632  000                           .BYTE   000
2949  002633  000                           .BYTE   000
2950  002634  377                           .BYTE   377
2951  002635  017                           .BYTE   017
2952  002636  377                           .BYTE   377
2953  002637  377                           .BYTE   377
2954  002640  375                           .BYTE   375
2955  002641  377                           .BYTE   377
2956
2957                                ;***** DATA PATTERN I *****
2958  002642                        PATI:
2959  002642  000                           .BYTE   000
2960  002643  000                           .BYTE   000
2961  002644  000                           .BYTE   000
2962  002645  000                           .BYTE   000
2963  002646  000                           .BYTE   000
2964  002647  103                           .BYTE   103
2965  002650  000                           .BYTE   000
2966  002651  000                           .BYTE   000
2967
2968                                ;***** DATA PATTERN J *****
2969  002652                        PATJ:
2970  002652  000                           .BYTE   000
2971  002653  000                           .BYTE   000
2972  002654  010                           .BYTE   010
2973  002655  002                           .BYTE   002
2974  002656  004                           .BYTE   004
2975  002657  103                           .BYTE   103
2976  002660  001                           .BYTE   001
```

```
 2977  002661      100                        .BYTE   100
 2978
 2979                                         ;***** DATA PATTERN K *****
 2980  002662      000                  PATK:   .BYTE   000
 2981  002663      000                          .BYTE   000
 2982  002664      377                          .BYTE   377
 2983  002665      377                          .BYTE   377
 2984  002666      125                          .BYTE   125
 2985  002667      125                          .BYTE   125
 2986  002670      252                          .BYTE   252
 2987  002671      252                          .BYTE   252
 2988  002672      001                          .BYTE   001
 2989  002673      000                          .BYTE   000
 2990  002674      002                          .BYTE   002
 2991  002675      000                          .BYTE   000
 2992  002676      004                          .BYTE   004
 2993  002677      000                          .BYTE   000
 2994  002700      010                          .BYTE   010
 2995  002701      000                          .BYTE   000
 2996  002702      020                          .BYTE   020
 2997  002703      000                          .BYTE   000
 2998  002704      040                          .BYTE   040
 2999  002705      000                          .BYTE   000
 3000  002706      100                          .BYTE   100
 3001  002707      000                          .BYTE   000
 3002  002710      200                          .BYTE   200
 3003  002711      000                          .BYTE   000
 3004  002712      000                          .BYTE   000
 3005  002713      001                          .BYTE   001
 3006  002714      000                          .BYTE   000
 3007  002715      002                          .BYTE   002
 3008  002716      000                          .BYTE   000
 3009  002717      004                          .BYTE   004
 3010  002720      000                          .BYTE   000
 3011  002721      010                          .BYTE   010
 3012  002722      000                          .BYTE   000
 3013  002723      020                          .BYTE   020
 3014  002724      000                          .BYTE   000
 3015  002725      040                          .BYTE   040
 3016  002726      000                          .BYTE   000
 3017  002727      100                          .BYTE   100
 3018  002730      000                          .BYTE   000
 3019  002731      200                          .BYTE   200
 3020  002732      376                          .BYTE   376
 3021  002733      377                          .BYTE   377
 3022  002734      375                          .BYTE   375
 3023  002735      377                          .BYTE   377
 3024  002736      373                          .BYTE   373
 3025  002737      377                          .BYTE   377
 3026  002740      367                          .BYTE   367
 3027  002741      377                          .BYTE   377
 3028  002742      357                          .BYTE   357
 3029  002743      377                          .BYTE   377
 3030  002744      337                          .BYTE   337
 3031  002745      377                          .BYTE   377
 3032  002746      277                          .BYTE   277
```

```
3033  002747  377                    .BYTE   377
3034  002750  177                    .BYTE   177
3035  002751  377                    .BYTE   377
3036  002752  377                    .BYTE   377
3037  002753  376                    .BYTE   376
3038  002754  377                    .BYTE   377
3039  002755  375                    .BYTE   375
3040  002756  377                    .BYTE   377
3041  002757  373                    .BYTE   373
3042  002760  377                    .BYTE   377
3043  002761  367                    .BYTE   367
3044  002762  377                    .BYTE   377
3045  002763  357                    .BYTE   357
3046  002764  377                    .BYTE   377
3047  002765  337                    .BYTE   337
3048  002766  377                    .BYTE   377
3049  002767  277                    .BYTE   277
3050  002770  377                    .BYTE   377
3051  002771  177                    .BYTE   177
3052
3053                                 ;***** DATA PATTERN L *****
3054  002772                         PATL:
3055  002772  000                    .BYTE   000
3056  002773  000                    .BYTE   000
3057  002774  377                    .BYTE   377
3058  002775  377                    .BYTE   377
3059  002776  000                    .BYTE   000
3060  002777  000                    .BYTE   000
3061
3062                                 ;***** DATA PATTERN M *****
3063  003000                         PATM:
3064  003000  000                    .BYTE   000
3065  003001  020                    .BYTE   020
3066  003002  000                    .BYTE   000
3067  003003  000                    .BYTE   000
3068  003004  200                    .BYTE   200
3069  003005  000                    .BYTE   000
3070  003006  000                    .BYTE   000
3071  003007  051                    .BYTE   051
3072
3073                                 ;***** DATA PATTERN N *****
3074  003010                         PATN:
3075  003010  000                    .BYTE   000
3076  003011  000                    .BYTE   000
3077  003012  000                    .BYTE   000
3078  003013  125                    .BYTE   125
3079  003014  000                    .BYTE   000
3080  003015  252                    .BYTE   252
3081  003016  000                    .BYTE   000
3082  003017  377                    .BYTE   377
3083  003020  005                    .BYTE   005
3084  003021  000                    .BYTE   000
3085  003022  012                    .BYTE   012
3086  003023  000                    .BYTE   000
3087  003024  017                    .BYTE   017
3088  003025  000                    .BYTE   000
```

```
3089
3090
3091  003026
3092  003026      000          ;***** DATA PATTERN 0 *****
3093  003027      041          PAT0:
3094  003030      004                  .BYTE   000
3095  003031      010                  .BYTE   041
3096  003032      020                  .BYTE   004
3097  003033      040                  .BYTE   010
3098  003034      100                  .BYTE   020
3099  003035      101                  .BYTE   040
3100  003036      200                  .BYTE   100
3101  003037      201                  .BYTE   101
3102  003040      300                  .BYTE   200
3103  003041      111                  .BYTE   201
3104  003042      301                  .BYTE   300
3105  003043      375                  .BYTE   111
3106                                   .BYTE   301
3107                                   .BYTE   375
3108  003044                     ;***** DATA PATTERN P *****
3109  003044      000          PATP:
3110  003045      113                  .BYTE   000
3111  003046      200                  .BYTE   113
3112  003047      040                  .BYTE   200
3113  003050      020                  .BYTE   040
3114  003051      010                  .BYTE   020
3115  003052      001                  .BYTE   010
3116  003053      104                  .BYTE   001
3117  003054      007                  .BYTE   104
3118  003055      105                  .BYTE   007
3119  003056      007                  .BYTE   105
3120  003057      144                  .BYTE   007
3121  003060      107                  .BYTE   144
3122  003061      157                  .BYTE   107
3123                                   .BYTE   157
3124                           ;***** DATA PATTERN U *****
3125  003062      000          PATU:   .BYTE   000
3126  003063      000                  .BYTE   000
3127  003064      001                  .BYTE   001
3128  003065      000                  .BYTE   000
3129  003066      013                  .BYTE   013
3130  003067      000                  .BYTE   000
3131  003070      011                  .BYTE   011
3132  003071      000                  .BYTE   000
3133  003072      021                  .BYTE   021
3134  003073      000                  .BYTE   000
3135  003074      101                  .BYTE   101
3136  003075      000                  .BYTE   000
3137  003076      301                  .BYTE   301
3138  003077      000                  .BYTE   000
3139  003100      000                  .BYTE   000
3140  003101      001                  .BYTE   001
3141  003102      000                  .BYTE   000
3142  003103      002                  .BYTE   002
3143  003104      000                  .BYTE   000
3144  003105      004                  .BYTE   004
```

```
3145  003106  000                .BYTE   000
3146  003107  040                .BYTE   040
3147  003110  000                .BYTE   000
3148  003111  100                .BYTE   100
3149  003112  000                .BYTE   000
3150  003113  200                .BYTE   200
3151  003114  000                .BYTE   000
3152  003115  346                .BYTE   346
3153  003116  000                .BYTE   000
3154  003117  345                .BYTE   345
3155  003120  000                .BYTE   000
3156  003121  343                .BYTE   343
3157  003122  000                .BYTE   000
3158  003123  307                .BYTE   307
3159  003124  000                .BYTE   000
3160  003125  247                .BYTE   247
3161  003126  000                .BYTE   000
3162  003127  147                .BYTE   147
3163
3164                          ;***** PATTERN V *****
3165  003130  000           PATV: .BYTE   000
3166  003131  000                .BYTE   000
3167  003132  333                .BYTE   333
3168  003133  000                .BYTE   000
3169  003134  331                .BYTE   331
3170  003135  000                .BYTE   000
3171  003136  323                .BYTE   323
3172  003137  000                .BYTE   000
3173  003140  313                .BYTE   313
3174  003141  000                .BYTE   000
3175  003142  233                .BYTE   233
3176  003143  000                .BYTE   000
3177  003144  133                .BYTE   133
3178  003145  000                .BYTE   000
3179  003146  000                .BYTE   000
3180  003147  001                .BYTE   001
3181  003150  000                .BYTE   000
3182  003151  002                .BYTE   002
3183  003152  000                .BYTE   000
3184  003153  004                .BYTE   004
3185  003154  000                .BYTE   000
3186  003155  040                .BYTE   040
3187  003156  000                .BYTE   000
3188  003157  100                .BYTE   100
3189  003160  000                .BYTE   000
3190  003161  200                .BYTE   200
3191  003162  000                .BYTE   000
3192  003163  346                .BYTE   346
3193  003164  000                .BYTE   000
3194  003165  345                .BYTE   345
3195  003166  000                .BYTE   000
3196  003167  343                .BYTE   343
3197  003170  000                .BYTE   000
3198  003171  307                .BYTE   307
3199  003172  000                .BYTE   000
3200  003173  247                .BYTE   247
```

```
CZDMRF.P11    03-NOV-81 10:29          GLOBAL DATA SECTION

 3201  003174     000                          .BYTE   000
 3202  003175     147                          .BYTE   147
 3203
 3204  003176                          ENDPAT:
 3205                                  .EVEN
 3206
 3207
 3208
 3209
 3210
 3211                                  ;***  TEST MESSAGES TO BE TRANSMITTED  ***
 3212
 3213  003176  000400                  MSG1:   TXSOM
 3214  003200  000400                          TXSOM
 3215  003202  000000                          000
 3216  003204  000125                          125
 3217  003206  000252                          252
 3218  003210  000377                          377
 3219  003212  000000                          000
 3220  003214  001000                          TXEOM
 3221  003216  001000                          TXEOM
 3222  003220  001000                          TXEOM
 3223  003222  001000                          TXEOM
 3224
 3225  003224  000377                  MSG4:   377
 3226  003226  000000                          000
 3227  003230  000125                          125
 3228  003232  000252                          252
 3229  003234  001000                          TXEOM
 3230  003236  001000                          TXEOM
 3231
 3232
 3233
 3234
 3235
 3236                                  ;*** RLCEIVED DATA BUFFER (64. WORDS) ***
 3237  003240  000100                  RCVBUF: .BLKW   64.
 3238
 3239
 3240
 3241
 3242
 3243
 3244
 3245
 3246
 3247
```

```
                                        .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
                                        ;;*************************************************************************
 003440                                         DEVTYP  <M8203>
 003440                                                                                 L$DVTYP::
 003440  034115  030062  000063                                                                 .ASCIZ  /M8203/
                                                                                                .EVEN

                                        ;;*************************************************************************
                                        ;* TITLE OF PROGRAM
                                        ;;*************************************************************************
 003446                                         DESCRIPT        <M8203 STATIC LOGIC TESTS - PART 1 OF 2>
 003446                                                                                 L$DESC::
 003446  034115  030062  020063                                                                 .ASCIZ  /M8203 STATIC LO
 003454  052123  052101  041511
 003462  046040  043517  041511
 003470  052040  051505  051524
 003476  026440  050040  051101
 003504  020124  020061  043117
 003512  031040     000
         003516                                                                                 .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.
                                ;*******************************************************************************
003516                          STPCLK:
003516  152777  000006  176702          BISB    #ROMO!ROMI,@BSEL1       ;SET ROMO, ROMI BITS IN BSEL1
003524  017677  000000  176702          MOV     @(SP),@SEL6     ;PUT INSTRUCTION INTO SEL6
003532  152777  000007  176666          BISB    #ROMO!ROMI!STEPMP,@BSEL1 ;SET ROMO, ROMI, STEPMP IN BSEL1
003540  142777  000007  176660          BICB    #ROMO!ROMI!STEPMP,@BSEL1 ;CLEAR ROMO, ROMI, STEPMP IN BSEL1
003546  062716  000002                  ADD     #2,(SP)         ;FIX UP RETURN PC
003552  000207                          RTS     PC              ;RETURN



                                ;*******************************************************************************
                                ;* MSTCLR - THIS SUBROUTINE ISSUES A MASTER CLEAR AND SETS LULOOP
                                ;*******************************************************************************
003554                          MSTCLR:
003554  010146                          MOV     R1,-(SP)        ;SAVE R1
003556  013746  002356                  MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
003562  112777  000100  176636          MOVB    #MCLR,@BSEL1    ;SET MASTER CLEAR BIT
003570  142777  000300  176630          BICB    #RUN!MCLR,@BSEL1 ;CLEAR RUN AND MCLR BITS
003576  012701  000024                  MOV     #20.,R1         ;INITIALIZE STALL COUNTER
003602  000240                  2$:     NOP                     ;STALL IN LOOP FOR SEVERAL MICRO-SEC
003604  005301                          DEC     R1
003606  001375                          BNE     2$
003610  152777  000010  176610          BISB    #LULOOP,@BSEL1  ;SET LU LOOP
003616  012737  000013  002356          MOV     #13,REGNUM      ;SET LU REG NO. = 13
003624  005037  002344                  CLR     WRIBYT
003630  004737  003726                  JSR     PC,WRITLU       ;CLEAR REG 13
003634  012637  002356                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
003640  012601                          MOV     (SP)+,R1        ;RESTORE R1
003642  005037  002410                  CLR     SAVLEN          ;CLEAR CHAR LENGTH FROM SETUP
003646  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.
                                ;*******************************************************************************
```

```
3348  003650                          READLU:
3349  003650  010146                          MOV     R1,-(SP)        ;SAVE R1
3350  003652  013701  002356                  MOV     REGNUM,R1       ;GET LINE UNIT REG NUMBER
3351  003656  006301                          ASL     R1              ;SHIFT INTO SOURCE BITS 4-7
3352  003660  006301                          ASL     R1
3353  003662  006301                          ASL     R1
3354  003664  006301                          ASL     R1
3355  003666  052701  000004                  BIS     #4,R1           ;SET DESTINATION = BSEL4
3356  003672  052701  021000                  BIS     #MVIOX,R1       ;SET REST OF MOVE INSTRUCTION
3357  003676  010137  003706                  MOV     R1,2$           ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
3358  003702  004737  003516                  JSR     PC,STPCLK       ;EXECUTE MOVE INSTRUCTION
3359  003706  000000                  2$:     .WORD   0               ;INSTRUCTION GOES HERE
3360  003710  117737  176516  002342          MOVB    @#BSEL4,REDBYT  ;GET LU REG CONTENTS INTO REDBYT
3361  003716  105037  002343                  CLRB    REDBYT+1        ;CLR HI BYTE OF STORAGE
3362  003722  012601                          MOV     (SP)+,R1        ;RESTORE R1
3363  003724  000207                          RTS     PC              ;RETURN
3364
3365
3366
3367
3368
3369                                  ;*******************************************************************************
3370                                  ;* WRITLU - THIS SUBROUTINE FORCES THE DMC11 OR KMC11 MICROPROCESSOR TO
3371                                  ;*       EXECUTE AN INSTRUCTION WHICH LOADS THE BYTE CONTAINED IN WRIBYT
3372                                  ;*       INTO THE LU REG WHOSE NUMBER IS PASSED IN REGNUM.
3373                                  ;*******************************************************************************
3374  003726                          WRITLU:
3375  003726  010146                          MOV     R1,-(SP)        ;SAVE R1
3376  003730  013701  002356                  MOV     REGNUM,R1       ;GET LINE UNIT REG NUMBER
3377  003734  052701  000100                  BIS     #100,R1         ;SET SOURCE = BSEL4
3378  003740  052701  122000                  BIS     #MVIXO,R1       ;SET REST OF MOVE INSTRUCTION
3379  003744  010137  003766                  MOV     R1,2$           ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
3380  003750  105037  002345                  CLRB    WRIBYT+1        ;CLR HI BYTE OF STORAGE
3381  003754  113777  002344  176450          MOVB    WRIBYT,@#BSEL4  ;LOAD BYTE INTO BSEL4
3382  003762  004737  003516                  JSR     PC,STPCLK       ;EXECUTE MOVE INSTRUCTION
3383  003766  000000                  2$:     .WORD   0
3384  003770  012601                          MOV     (SP)+,R1        ;RESTORE R1
3385  003772  000207                          RTS     PC              ;RETURN
3386
3387
3388
3389
3390
3391                                  ;*******************************************************************************
3392                                  ;* GETREG - THIS SUBROUTINE READS THE LINE UNIT REGISTERS 10-17 INTO THE
3393                                  ;*       REGISTER STORAGE TABLE (LUREG:).
3394                                  ;*******************************************************************************
3395  003774  010146                  GETREG: MOV     R1,-(SP)        ;SAVE R1
3396  003776  013746  002356                  MOV     REGNUM,-(SP)    ;SAVE CURRENT REG NO.
3397  004002  012701  002260                  MOV     #LUR10,R1       ;INIT POINTER TO REG STORAGE TABLE
3398  004006  012737  000010  002356          MOV     #10,REGNUM      ;INIT LU REG NO. TO 10
3399  004014  004737  003650          3$:     JSR     PC,READLU       ;READ A LINE UNIT REG
3400  004020  113721  002342                  MOVB    REDBYT,(R1)+    ;PUT BYTE READ INTO TABLE
3401  004024  105021                          CLRB    (R1)+           ;CLEAR UPPER BYTE OF TABLE ENTRY
3402  004026  005237  002356                  INC     REGNUM          ;INCREMENT REG NO.
3403  004032  023727  002356  000020          CMP     REGNUM,#20      ;SEE IF ALL REGS READ YET
```

```
3404  004040  002765                        BLT     3$                  ;BR IF NOT
3405  004042  012637  002356                MOV     (SP)+,REGNUM        ;RESTORE CURRENT REG NO.
3406  004046  012601                        MOV     (SP)+,R1            ;RESTORE R1
3407  004050  000207                        RTS     PC                  ;RETURN
3408
3409
3410
3411
3412
3413                                  ;*************************************************************************************
3414                                  ;* LOOPIN - THIS SUBROUTINE PLACES THE MICROPROCESSOR IN A LOOP ON AN
3415                                  ;*      INSTRUCTION, BY MOVING THE INSTRUCTION FROM THE WORD FOLLOWING THE CALL
3416                                  ;*      INTO SEL6, AND SETTING RUN AND ROMI IN BSEL1. THE SUBROUTINE RETURNS
3417                                  ;*      WITH THE MICROPROCESSOR STUCK IN THE LOOP, AND IF IT IS DESIRED TO
3418                                  ;*      TERMINATE THE LOOP, THE PDP-11 PROGRAM MUST CLEAR THE RUN BIT IN
3419                                  ;*      BSEL1, OR CALL SUBROUTINE MSTCLR TO DO THIS.
3420                                  ;*************************************************************************************
3421  004052                          LOOPIN:
3422  004052  152777  000006  176346        BISB    #ROMO!ROMI,@BSEL1       ;SET ROMO, ROMI BITS IN BSEL1
3423  004060  017677  000000  176346        MOV     @(SP),@SEL6     ;PUT MICROINSTRUCTION INTO SEL6
3424  004066  152777  000206  176332        BISB    #RUN!ROMO!ROMI,@BSEL1 ;SET RUN, ROMO, ROMI IN BSEL1
3425  004074  062716  000002                ADD     #2,(SP)             ;FIX UP RETURN PC
3426  004100  000207                        RTS     PC                  ;RETURN WITH MICROPROCESSOR STUCK IN SINGLE
3427                                                                    ;    INSTRUCTION LOOP
3428
3429
3430
3431
3432
3433                                  ;*************************************************************************************
3434                                  ;* READAX - THIS SUBROUTINE READS THE USYRT REG PAIR WHOSE NUMBER (0-3)
3435                                  ;*      IS PASSED IN BITS 1,2 OF AXNUM ON ENTRY, AND RETURNS THE BYTES READ IN
3436                                  ;*      RAX15 AND RAX16. IF THE LINE UNIT DOES NOT RESPOND WITH READY IN REG 14,
3437                                  ;*      RRDYTO BIT IS SET IN ERROR1 ON RETURN.
3438                                  ;*************************************************************************************
3439  004102  010146                  READAX: MOV     R1,-(SP)            ;SAVE R1
3440  004104  013746  002356                  MOV     REGNUM,-(SP)        ;STORE CURRENT REG NO.
3441  004110  042737  000001  002376          BIC     #RRDYTO,ERROR1      ;CLEAR ERROR BIT
3442  004116  012737  000014  002356          MOV     #14,REGNUM          ;SET LU REG NO. = 14
3443  004124  113737  002360  002344          MOVB    AXNUM,WRIBYT        ;SET UP AX REG NO. BITS
3444  004132  006237  002344                  ASR     WRIBYT
3445  004136  152737  000024  002344          BISB    #RDAX!ENAX,WRIBYT ;SET UP BITS TO LOAD INTO REG 14
3446  004144  053737  002404  002344          BIS     DISILO,WRIBYT       ;SET PROPER STATE OF DISSI BIT
3447  004152  004737  003726                  JSR     PC,WRITLU           ;SET RDAX AND ENAX IN REG 14
3448  004156  005001                          CLR     R1                  ;INIT TIMER
3449  004160  004737  003650          6$:     JSR     PC,READLU           ;READ REG 14
3450  004164  132737  000200  002342          BITB    #READY,REDBYT       ;SEE IF READY BIT SET IN REG 14 YET
3451  004172  001006                          BNE     9$                  ;BR IF READY SET
3452  004174  005201                          INC     R1                  ;INCR TIMER
3453  004176  001370                          BNE     6$                  ;BR IF TIMER DIDN'T TIME OUT YET
3454  004200  052737  000001  002376          BIS     #RRDYTO,ERROR1      ;SET ERROR FLAG FOR TIME OUT ON READ RDY
3455  004206  000424                          BR      12$                 ;BR TO RETURN
3456  004210  012737  000015  002356  9$:     MOV     #15,REGNUM          ;SET REG NO. = 15
3457  004216  004737  003650                  JSR     PC,READLU           ;READ REG 15
3458  004222  113737  002342  002346          MOVB    REDBYT,RAX15        ;STORE REG AX-15
3459  004230  105037  002347                  CLRB    RAX15+1             ;CLR HI BYTE OF STORAGE
```

```
3460  004234  012737  000016  002356          MOV     #16,REGNUM      ;SET REG NO. = 16
3461  004242  004737  003650                  JSR     PC,READLU       ;READ REG 16
3462  004246  113737  002342  002350          MOVB    REDBYT,RAX16    ;STORE REG AX-16
3463  004254  105037  002351                  CLRB    RAX16+1         ;CLR HI BYTE OF STORAGE
3464  004260  012637  002356          12$:    MOV     (SP)+,REGNUM    ;RESTORE CURRENT REG NO.
3465  004264  012601                          MOV     (SP)+,R1        ;RESTORE R1
3466  004266  000207                          RTS     PC              ;RETURN
3467
3468
3469
3470
3471                                  ;*************************************************************************
3472                                  ;* WRITAX - THIS SUBROUTINE WRITES THE USYRT REG PAIR WHOSE NUMBER (0-3) IS
3473                                  ;*      PASSED IN BITS 1,2 OF AXNUM ON ENTRY, WITH THE DATA FROM WAX15 AND
3474                                  ;*      WAX16. IF LINE UNIT DOES NOT RESPOND WITH READY IN REG 14, WRDYTO BIT
3475                                  ;*      IS SET IN ERROR1 ON RETURN.
3476                                  ;*************************************************************************
3477  004270  010146                  WRITAX: MOV     R1,-(SP)        ;SAVE R1
3478  004272  013746  002356                  MOV     REGNUM,-(SP)    ;SAVE CURRENT REG NO.
3479  004276  042737  000002  002376          BIC     #WRDYTO,ERROR1  ;CLEAR ERROR BIT
3480  004304  012737  000014  002356          MOV     #14,REGNUM      ;SET LU REG NO. = 14
3481  004312  113737  002360  002344          MOVB    AXNUM,WRIBYT    ;SET AX REG NO. BITS
3482  004320  006237  002344                  ASR     WRIBYT
3483  004324  053737  002404  002344          BIS     DISILO,WRIBYT   ;SET PROPER STATE OF DISSI BIT
3484  004332  004737  003726                  JSR     PC,WRITLU       ;SET AX NO. BITS IN REG 14
3485  004336  012737  000015  002356          MOV     #15,REGNUM      ;SET REG NO. = 15
3486  004344  105037  002353                  CLRB    WAX15+1         ;CLR HI BYTE OF STORAGE
3487  004350  113737  002352  002344          MOVB    WAX15,WRIBYT    ;SET UP BYTE TO WRITE INTO REG 15
3488  004356  004737  003726                  JSR     PC,WRITLU       ;WRITE BYTE INTO REG 15
3489  004362  005237  002356                  INC     REGNUM          ;SET REG NO. = 16
3490  004366  105037  002355                  CLRB    WAX16+1         ;CLR HI BYTE OF STORAGE
3491  004372  113737  002354  002344          MOVB    WAX16,WRIBYT    ;SET UP BYTE TO WRITE INTO REG 16
3492  004400  004737  003726                  JSR     PC,WRITLU       ;WRITE BYTE INTO REG 16
3493  004404  012737  000014  002356          MOV     #14,REGNUM      ;SET REG NO. = 14
3494  004412  113737  002360  002344          MOVB    AXNUM,WRIBYT    ;SET AX REG NO. BITS
3495  004420  006237  002344                  ASR     WRIBYT
3496  004424  152737  000014  002344          BISB    #ENAX!WAX,WRIBYT ;SET UP BITS TO LOAD INTO REG 14
3497  004432  053737  002404  002344          BIS     DISILO,WRIBYT   ;SET PROPER STATE OF DISSI BIT
3498  004440  004737  003726                  JSR     PC,WRITLU       ;SET ENAX AND WAX IN REG 14
3499  004444  005001                          CLR     R1              ;INIT PROGRAM TIMER
3500  004446  004737  003650          6$:     JSR     PC,READLU       ;READ REG 14
3501  004452  132737  000200  002342          BITB    #READY,REDBYT   ;SEE IF READY BIT SET IN REG 14 YET
3502  004460  001005                          BNE     9$              ;BR IF READY SET
3503  004462  005201                          INC     R1              ;INCR TIMER
3504  004464  001370                          BNE     6$              ;BR IF TIMER DIDN'T TIME OUT YET
3505  004466  052737  000002  002376          BIS     #WRDYTO,ERROR1  ;SET ERROR FLAG BIT FOR TIME OUT ON WRITE RDY
3506  004474  012637  002356          9$:     MOV     (SP)+,REGNUM    ;RESTORE CURRENT REG NO.
3507  004500  012601                          MOV     (SP)+,R1        ;RESTORE R1
3508  004502  000207                          RTS     PC              ;RETURN
3509
3510
3511
3512
3513
3514                                  ;*************************************************************************
3515                                  ;* GETALL - THIS SUBROUTINE READS THE LINE UNIT REGS 10-17 AND THE EXTENDED
```

```
                        ;*      REGISTERS AX0-AX3 INTO REGISTER STORAGE TABLE (LUREG:).
                        ;*************************************************************************
004504  010146          GETALL: MOV     R1,-(SP)        ;SAVE R1
004506  013746  002360          MOV     AXNUM,-(SP)     ;SAVE CURRENT AX REG BYTE NO.
004512  012737  014413  002506  MOV     #DH5,TMP0       ;SET AX LO BYTE NO.
004520  032737  000001  002360  BIT     #BIT0,AXNUM     ;SEE IF LO OR HI BYTE
004526  001403                  BEQ     1$              ;BR IF LO BYTE
004530  012737  014416  002506  MOV     #DH6,TMP0       ;SET AX HI BYTE NO.
004536  004737  003774      1$: JSR     PC,GETREG       ;READ AND STORE REGS 10-17
004542  142777  000010  175656  BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
004550  012701  002300          MOV     #AX0.15,R1      ;INIT POINTER TO REG STORAGE TABLE
004554  005037  002360          CLR     AXNUM           ;INIT AX REG BYTE NO. TO 0
004560  004737  004102      3$: JSR     PC,READAX       ;READ 2 AX REG BYTES
004564  113721  002346          MOVB    RAX15,(R1)+     ;PUT LO BYTE READ INTO TABLE
004570  105021                  CLRB    (R1)+           ;CLEAR UPPER BYTE OF TABLE ENTRY
004572  113721  002350          MOVB    RAX16,(R1)+     ;PUT HI BYTE READ INTO TABLE
004576  105021                  CLRB    (R1)+           ;CLEAR UPPER BYTE OF TABLE ENTRY
004600  062737  000002  002360  ADD     #2,AXNUM        ;INCR AX REG BYTE NO.
004606  023727  002360  000010  CMP     AXNUM,#10       ;SEE IF ALL REGS READ YET
004614  002761                  BLT     3$              ;BR IF NOT
004616  012637  002360          MOV     (SP)+,AXNUM     ;RESTORE CURRENT AX REG BYTE NO.
004622  012601                  MOV     (SP)+,R1        ;RESTORE R1
004624  013737  002360  002510  MOV     AXNUM,TMP1
004632  006237  002510          ASR     TMP1            ;GET EXTENDED REG NO. FOR PRINTOUT
004636  000207                  RTS     PC              ;RETURN


                        ;*************************************************************************
                        ;* OSIRDY - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF ORDY (REG 11)
                        ;*      AND OCOR (REG 17) AND REPORTS AN ERROR IF EITHER IS NOT PROPERLY SET
                        ;*      AS PASSED IN BIT 0 (ORDY) AND BIT 1 (OCOR) OF THE WORD FOLLOWING THE
                        ;*      CALL.
                        ;*      IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST, AT THE ADDRESS IN
                        ;*      RETADR.
                        ;*************************************************************************
004640  013746  002356  OSIRDY: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
004644  013746  002330          MOV     SUBRPC,-(SP)
004650  005737  002330          TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
004654  001006                  BNE     1$              ;BR IF YES
004656  016637  000004  002330  MOV     4(SP),SUBRPC
004664  162737  000004  002330  SUB     #4,SUBRPC       ;GET PC OF SUBROUTINE CALL
004672  012737  000011  002356  1$:     MOV     #11,REGNUM      ;SET REG NO. TO 11
004700  004737  003650          JSR     PC,READLU       ;READ REG 11
004704  032776  000001  000004  BIT     #BIT0,@4(SP)    ;GET EXPECTED STATE OF ORDY
004712  001413                  BEQ     3$              ;BR IF EXPECTED ORDY = 0
004714  132737  000020  002342  BITB    #ORDY,REDBYT    ;SEE IF ORDY = 1
004722  001022                  BNE     9$              ;BR IF ORDY = 1
004724  004737  004504          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                        ;REPORT ORDY NOT SET
004730                  ERRDF   7,EM7,ERR4
004730  104455                                                  TRAP    C$ERDF
004732  000007                                                  .WORD   7
004734  012352                                                  .WORD   EM7
```

```
3572  004736  015600                                                                    .WORD   ERR4
3573  004740  000451                    BR      16$             ;TAKE ERROR RETURN
3574  004742  132737  000020  002342  3$:       BITB    #ORDY,REDBYT    ;SEE IF ORDY = 0
3575  004750  001407                    BEQ     9$              ;BR IF ORDY = 0
3576  004752  004737  004504            JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3577                                    ;REPORT ORDY NOT CLEARED
3578  004756                            ERRDF   8,EM8,ERR4
3579  004756  104455                                                                    TRAP    C$ERDF
3580  004760  000010                                                                    .WORD   8
3581  004762  012367                                                                    .WORD   EM8
3582  004764  015600                                                                    .WORD   ERR4
3583  004766  000436                    BR      16$             ;TAKE ERROR RETURN
3584  004770  012737  000017  002356  9$:       MOV     #17,REGNUM      ;SET REG NO. = 17
3585  004776  004737  003650            JSR     PC,READLU       ;READ LU REG 17
3586  005002  132776  000002  000004    BITB    #BIT1,@4(SP)    ;GET EXPECTED STATE OF OCOR
3587  005010  001413                    BEQ     12$             ;BR IF EXPECTED OCOR = 0
3588  005012  132737  000020  002342    BITB    #OCOR,REDBYT    ;SEE IF OCOR = 1
3589  005020  001031                    BNE     20$             ;BR IF OCOR = 1
3590  005022  004737  004504            JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3591                                    ;REPORT OCOR NOT SET
3592  005026                            ERRDF   9,EM9,ERR4
3593  005026  104455                                                                    TRAP    C$ERDF
3594  005030  000011                                                                    .WORD   9
3595  005032  012410                                                                    .WORD   EM9
3596  005034  015600                                                                    .WORD   ERR4
3597  005036  000412                    BR      16$             ;TAKE ERROR RETURN
3598  005040  132737  000020  002342  12$:      BITB    #OCOR,REDBYT    ;SEE IF OCOR = 0
3599  005046  001416                    BEQ     20$             ;BR IF OCOR = 0
3600  005050  004737  004504            JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3601                                    ;REPORT OCOR NOT CLEARED
3602  005054                            ERRDF   10,EM10,ERR4
3603  005054  104455                                                                    TRAP    C$ERDF
3604  005056  000012                                                                    .WORD   10
3605  005060  012425                                                                    .WORD   EM10
3606  005062  015600                                                                    .WORD   ERR4
3607  005064  016637  000002  002356  16$:      MOV     2(SP),REGNUM    ;RESTORE LU REG NO.
3608  005072  013706  002324            MOV     PSTACK,SP       ;RESTORE STACK POINTER TO BASE LEVEL
3609  005076  013746  002340            MOV     RETADR,-(SP)    ;FIX ERROR RETURN PC
3610  005102  000407                    BR      23$
3611  005104  062766  000002  000004  20$:      ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
3612  005112  012637  002330            MOV     (SP)+,SUBRPC
3613  005116  012637  002356            MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
3614  005122  000207                    23$:    RTS     PC              ;RETURN
3615
3616
3617
3618
3619
3620                                    ;*****************************************************************************
3621                                    ;* WAIT50 - THIS SUBROUTINE STALLS FOR AT LEAST 50 MICRO-SEC, AND THEN RETURNS.
3622                                    ;*****************************************************************************
3623  005124  010146                    WAIT50: MOV     R1,-(SP)        ;SAVE R1
3624  005126  012701  000310            MOV     #200.,R1        ;INIT COUNTER
3625  005132  005301                    3$:     DEC     R1              ;DECREMENT COUNTER
3626  005134  001376                    BNE     3$              ;BR IF NOT DONE YET
3627  005136  012601                    MOV     (SP)+,R1        ;RESTORE R1
```

```
3628  005140  000207                          RTS     PC              ;RETURN
3629
3630
3631
3632
3633
3634                                  ;*********************************************************************************
3635                                  ;* STALL - THIS SUBROUTINE STALLS FOR ABOUT A MICRO-SEC.
3636                                  ;*********************************************************************************
3637  005142  000240                  STALL:  NOP
3638  005144  000240                          NOP
3639  005146  000240                          NOP
3640  005150  000207                          RTS     PC
3641
3642
3643
3644
3645
3646                                  ;*********************************************************************************
3647                                  ;* LDTXSI - THIS SUBROUTINE LOADS THE TX SILO (REGS 10,11) WITH THE DATA PASSED
3648                                  ;*       IN BITS 0-11 OF TXWORD.
3649                                  ;*********************************************************************************
3650  005152  013746  002356          LDTXSI: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
3651  005156  042737  170000  002400          BIC     #170000,TXWORD  ;CLEAR UNUSED BITS
3652  005164  012737  000011  002356          MOV     #11,REGNUM      ;SET REG NO. = 11
3653  005172  113737  002401  002344          MOVB    TXWORD+1,WRIBYT ;SET DATA TO BE WRITTEN INTO REG 11
3654  005200  004737  003726                  JSR     PC,WRITLU       ;LOAD DATA INTO REG 11
3655  005204  012737  000010  002356          MOV     #10,REGNUM      ;SET REG NO. = 10
3656  005212  113737  002400  002344          MOVB    TXWORD,WRIBYT   ;SET DATA TO BE WRITTEN INTO REG 10
3657  005220  004737  003726                  JSR     PC,WRITLU       ;LOAD DATA INTO REG 10
3658  005224  012637  002356                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
3659  005230  000207                          RTS     PC              ;RETURN
3660
3661
3662
3663
3664
3665                                  ;*********************************************************************************
3666                                  ;* STPLU - THIS SUBROUTINE CLOCKS THE LINE UNIT FOR THE NO. OF CYCLES PASSED
3667                                  ;*       IN BITS 0-14 OF THE WORD FOLLOWING THE CALL.
3668                                  ;*       IF BIT 15 = 1, A CHECK IS MADE TO DETERMINE IF THE USYRT CHIP TYPE
3669                                  ;*       REQUIRES DECREMENTING THE NO. OF CYCLES BY 1.
3670                                  ;*********************************************************************************
3671  005232  010146                  STPLU:  MOV     R1,-(SP)        ;SAVE R1
3672  005234  017601  000002                  MOV     @2(SP),R1       ;GET DESIRED NO. OF CYCLES
3673  005240  001426                          BEQ     6$              ;IF DESIRED CYCLES = 0, RETURN
3674  005242  100006                          BPL     2$              ;BR IF CHIP TYPE CHECK NOT NECESSARY
3675  005244  042701  100000                  BIC     #BIT15,R1       ;CLEAR FLAG BIT
3676  005250  005737  002406                  TST     CHPTYP          ;SEE IF SIG USYRT
3677  005254  001401                          BEQ     2$              ;BR IF YES
3678  005256  005301                          DEC     R1              ;DECREMENT CYCLE COUNT
3679  005260  152777  000010  175140  2$:     BISB    #LULOOP,@BSEL1  ;SET LU LOOP BIT
3680  005266  152777  000020  175132  3$:     BISB    #STEPLU,@BSEL1  ;SET THE STEPLU BIT (CLOCK THE TRANSMITTER)
3681  005274  004737  005142                  JSR     PC,STALL        ;STALL
3682  005300  142777  000020  175120          BICB    #STEPLU,@BSEL1  ;CLEAR THE STEPLU BIT (CLOCK THE RECEIVER)
3683  005306  004737  005142                  JSR     PC,STALL        ;STALL
```

```
3684  005312  005301                          DEC     R1              ;DECREMENT CYCLE COUNTER
3685  005314  001364                          BNE     3$              ;BR IF NOT DONE YET
3686  005316  062766  000002  000002  6$:     ADD     #2,2(SP)        ;FIX UP RETURN PC
3687  005324  012601                          MOV     (SP)+,R1        ;RESTORE R1
3688  005326  000207                          RTS     PC              ;RETURN
3689
3690
3691
3692
3693
3694                                  ;*************************************************************************
3695                                  ;* OACTIV - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF OACT (REG 11) AND
3696                                  ;*      REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE STATE OF BIT 0 IN THE
3697                                  ;*      WORD FOLLOWING THE CALL.
3698                                  ;*************************************************************************
3699  005330  013746  002356          OACTIV: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
3700  005334  013746  002330                  MOV     SUBRPC,-(SP)
3701  005340  005737  002330                  TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
3702  005344  001006                          BNE     1$              ;BR IF YES
3703  005346  016637  000004  002330          MOV     4(SP),SUBRPC
3704  005354  162737  000004  002330          SUB     #4,SUBRPC       ;GET PC OF SUBROUTINE CALL
3705  005362  012737  000011  002356  1$:     MOV     #11,REGNUM      ;SET REG NO. = 11
3706  005370  004737  003650                  JSR     PC,READLU       ;READ REG 11
3707  005374  032776  000001  000004          BIT     #BIT0,@4(SP)    ;GET EXPECTED STATE OF OACT
3708  005402  001413                          BEQ     3$              ;BR IF EXPECTED OACT = 0
3709  005404  132737  000100  002342          BITB    #OACT,REDBYT    ;SEE IF OACT = 1
3710  005412  001031                          BNE     9$              ;BR IF OACT = 1
3711  005414  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3712                                  ;REPORT OACT NOT SET
3713  005420                                  ERRDF   11,EM11,ERR4
3714  005420  104455                                                                  TRAP    C$ERDF
3715  005422  000013                                                                  .WORD   11
3716  005424  012446                                                                  .WORD   EM11
3717  005426  015600                                                                  .WORD   ERR4
3718  005430  000412                          BR      6$              ;TAKE ERROR RETURN
3719  005432  132737  000100  002342  3$:     BITB    #OACT,REDBYT    ;SEE IF OACT = 0
3720  005440  001416                          BEQ     9$              ;BR IF OACT = 0
3721  005442  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3722                                  ;REPORT OACT NOT CLEARED
3723  005446                                  ERRDF   12,EM12,ERR4
3724  005446  104455                                                                  TRAP    C$ERDF
3725  005450  000014                                                                  .WORD   12
3726  005452  012463                                                                  .WORD   EM12
3727  005454  015600                                                                  .WORD   ERR4
3728  005456  016637  000002  002356  6$:     MOV     2(SP),REGNUM    ;RESTORE LU REG NO.
3729  005464  013706  002324                  MOV     PSTACK,SP       ;RESTORE PROGRAM STACK TO BASE LEVEL
3730  005470  013746  002340                  MOV     RETADR,-(SP)    ;FIX UP ERROR RETURN PC
3731  005474  000407                          BR      12$
3732  005476  062766  000002  000004  9$:     ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
3733  005504  012637  002330                  MOV     (SP)+,SUBRPC
3734  005510  012637  002356                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
3735  005514  000207                  12$:    RTS     PC              ;RETURN
3736
3737
3738
3739
```

```
3740
3741                                    ;*****************************************************************************
3742                                    ;* INITRN - THIS SUBROUTINE INITIATES TRANSMISSION OF A MESSAGE, BY DOING A
3743                                    ;*      MASTER CLEAR, LOADING AX2-15 AND REG 17 WITH THE DATA PASSED IN THE 2
3744                                    ;*      WORDS FOLLOWING THE CALL, LOADING 2 SOM CHARS INTO THE TX SILO, AND
3745                                    ;*      CLOCKING THE LINE UNIT UNTIL THE FIRST SYNCH OR FLAG HAS BEEN SERIALIZED
3746                                    ;*      IN THE USYRT. THE PROGRAM MONITORS ORDY,OCOR, AND OACT FOR VALID STATES,
3747                                    ;*      THROUGHOUT THE PROCESS.
3748                                    ;*      IF THE SUBROUTINE DETECTS AN ERROR, A RETURN IS MADE TO THE TEST, AT THE
3749                                    ;*      ADDRESS CONTAINED IN RETADR.
3750                                    ;*****************************************************************************
3751  005516  010146                    INITRN: MOV     R1,-(SP)        ;SAVE R1
3752  005520  013746  002356                    MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
3753  005524  013746  002360                    MOV     AXNUM,-(SP)     ;SAVE AX BYTE NO.
3754  005530  016637  000006  002330            MOV     6(SP),SUBRPC
3755  005536  162737  000004  002330            SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
3756  005544  004737  003554                    JSR     PC,MSTCLR       ;ISSUE A MASTER CLEAR
3757  005550  004737  004640                    JSR     PC,OSIRDY       ;CHECK ORDY=1, OCOR=0
3758  005554  000001                            1
3759  005556  004737  005330                    JSR     PC,OACTIV       ;CHK OACT=0
3760  005562  000000                            0
3761  005564  012737  000004  002360            MOV     #4,AXNUM        ;SET AX BYTE NO. = 4 FOR AX2
3762  005572  117637  000006  002352            MOVB    @6(SP),WAX15    ;SET DATA BYTE TO LOAD INTO AX2-15
3763  005600  012737  000400  002400            MOV     #TXSOM,TXWORD   ;SET TSOM BIT
3764  005606  113737  002352  002400            MOVB    WAX15,TXWORD    ;SET SYNCH CHAR
3765  005614  005037  002354                    CLR     WAX16
3766  005620  004737  004270                    JSR     PC,WRITAX       ;LOAD AX2
3767  005624  012737  000017  002356            MOV     #17,REGNUM      ;SET REG NO. = 17
3768  005632  062766  000002  000006            ADD     #2,6(SP)        ;INCR POINTER TO NEXT DATA BYTE
3769  005640  117637  000006  002344            MOVB    @6(SP),WRIBYT   ;SET DATA BYTE TO LOAD INTO REG 17
3770  005646  004737  003726                    JSR     PC,WRITLU       ;LOAD REG 17
3771  005652  004737  005152                    JSR     PC,LDTXSI       ;LOAD THE SILO WITH SOM CHAR
3772  005656  004737  005152                    JSR     PC,LDTXSI       ;LOAD ANOTHER SOM INTO SILO
3773  005662  004737  005124                    JSR     PC,WAIT50       ;WAIT FOR DATA TO RIPPLE
3774  005666  004737  004640                    JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=1
3775  005672  000003                            3
3776  005674  004737  005330                    JSR     PC,OACTIV       ;CHK FOR OACT = 0
3777  005700  000000                            0
3778  005702  005001                            CLR     R1              ;INIT CYCLE COUNTER
3779  005704  012737  000011  002356            MOV     #11,REGNUM      ;SET LU REG NO. = 11
3780  005712  152777  000010  174506  6$:       BISB    #LULOOP,@BSEL1  ;SET LINE UNIT LOOP BIT
3781  005720  152777  000020  174500            BISB    #STEPLU,@BSEL1  ;SET CLOCK BIT
3782  005726  004737  005142                    JSR     PC,STALL        ;STALL FOR MICRO-SEC
3783  005732  004737  003650                    JSR     PC,READLU       ;READ REG 11
3784  005736  132737  000100  002342            BITB    #OACT,REDBYT    ;SEE IF OACT = 1 YET
3785  005744  001014                            BNE     9$              ;BR IF OACT = 1
3786  005746  142777  000020  174452            BICB    #STEPLU,@BSEL1  ;CLEAR CLOCK BIT
3787  005754  004737  005142                    JSR     PC,STALL        ;STALL FOR A MICRO-SEC
3788  005760  005201                            INC     R1              ;INCR CYCLE COUNT
3789  005762  020127  000003                    CMP     R1,#3           ;SEE IF 3 CYCLES DONE YET
3790  005766  002751                            BLT     6$              ;BR IF NOT
3791  005770  004737  005330                    JSR     PC,OACTIV       ;CHK FOR OACT = 1
3792  005774  000001                            1
3793  005776  012737  000017  002356  9$:       MOV     #17,REGNUM      ;SET REG NO. = 17
3794  006004  005037  002406                    CLR     CHPTYP          ;CLEAR USYRT CHIP INDICATOR
3795  006010  004737  003650                    JSR     PC,READLU       ;READ REG 17
```

```
3796  006014  132737  000020  002342          BITB    #OCOR,REDBYT    ;CHK FOR OCOR CLEARED YET
3797  006022  001403                          BEQ     12$             ;BR IF YES - IT IS SIG CHIP
3798  006024  012737  000001  002406          MOV     #1,CHPTYP       ;SET INDICATOR FOR OTHER CHIP TYPE
3799  006032  142777  000020  174366  12$:    BICB    #STEPLU,@BSEL1  ;CLEAR CLOCK BIT
3800  006040  004737  005142                  JSR     PC,STALL        ;STALL FOR MICRO-SEC
3801  006044  0C4737  004640                  JSR     PC,OSIRDY       ;CHK FOR ORDY = 1, OCOR = 0
3802  006050  000001                          1
3803  006052  062766  000002  000006          ADD     #2,6(SP)        ;FIX UP RETURN PC
3804  006060  012637  002360                  MOV     (SP)+,AXNUM     ;RESTORE AX BYTE NO.
3805  006064  012637  002356                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
3806  006070  012601                          MOV     (SP)+,R1        ;RESTORE R1
3807  006072  005037  002330                  CLR     SUBRPC          ;CLEAR SUBR CALL PC
3808  006076  000207                          RTS     PC              ;RETURN
3809
3810
3811
3812
3813
3814                                  ;*****************************************************************************
3815                                  ;* TXCHAR - THIS SUBROUTINE INITIATES TRANSMISSION OF A CHARACTER, BY LOADING
3816                                  ;*       THE TX SILO WITH DATA PASSED IN BITS 0-11 OF THE WORD FOLLOWING THE CALL
3817                                  ;*       AND CLOCKS THE LINE UNIT WITH THE NUMBER OF CYCLES PASSED IN BITS 0-14
3818                                  ;*       OF THE SECOND WORD FOLLOWING THE CALL. IF BIT 15 = 1, A CHK IS MADE TO
3819                                  ;*       DETERMINE IF THE USYRT CHIP TYPE REQUIRES DECREMENTING THE NO. OF CYCLES
3820                                  ;*       BY 1. THE PROGRAM CHECKS FOR VALID STATES OF ORDY,
3821                                  ;*       OCOR, AND OACT THROUGHOUT THE PROCESS.
3822                                  ;*       IF AN ERROR IS DETECTED, A RETURN IS MADE TO THE TEST, AT THE ADDRESS
3823                                  ;*       CONTAINED IN RETADR.
3824                                  ;*****************************************************************************
3825  006100  010146                  TXCHAR: MOV     R1,-(SP)        ;SAVE R1
3826  006102  010246                          MOV     R2,-(SP)        ;SAVE R2
3827  006104  016637  000004  002330          MOV     4(SP),SUBRPC
3828  006112  162737  000004  002330          SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
3829  006120  017637  000004  002400          MOV     @4(SP),TXWORD   ;GET DATA TO BE TRANSMITTED
3830  006126  004737  005152                  JSR     PC,LDTXSI       ;LOAD THE TX SILO WITH THE DATA
3831  006132  004737  005124                  JSR     PC,WAIT50       ;WAIT FOR DATA TO RIPPLE DOWN SILO
3832  006136  062766  000002  000004          ADD     #2,4(SP)        ;INCR POINTER
3833  006144  005001                          CLR     R1              ;INIT CYCLE COUNT
3834  006146  017602  000004                  MOV     @4(SP),R2       ;GET DESIRED NO. OF CYCLES
3835  006152  005702                          TST     R2              ;SEE IF CHIP TYPE CHK SHOULD BE MADE
3836  006154  100006                          BPL     9$              ;BR IF NOT
3837  006156  042702  100000                  BIC     #BIT15,R2       ;CLEAR FLAG BIT
3838  006162  005737  002406                  TST     CHPTYP          ;SEE IF SIG USYRT
3839  006166  001401                          BEQ     9$              ;BR IF YES
3840  006170  005302                          DEC     R2              ;DECREMENT NO. OF CYCLES
3841  006172  004737  005330          9$:     JSR     PC,OACTIV       ;CHK OACT = 1
3842  006176  000001                          1
3843  006200  020102                          CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
3844  006202  001410                          BEQ     12$             ;BR IF YES
3845  006204  004737  004640                  JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=1
3846  006210  000003                          3
3847  006212  004737  005232                  JSR     PC,STPLU        ;STEP LU ONE CYCLE
3848  006216  000001                          1
3849  006220  005201                          INC     R1              ;INCR CYCLE COUNT
3850  006222  000763                          BR      9$
3851  006224  004737  004640          12$:    JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=0
```

```
3852  006230  000001                          1
3853  006232  062766  000002  000004          ADD     #2,4(SP)        ;FIX UP RETURN PC
3854  006240  005037  002330                  CLR     SUBRPC          ;CLEAR SUBR CALL PC
3855  006244  012602                          MOV     (SP)+,R2        ;RESTORE R2
3856  006246  012601                          MOV     (SP)+,R1        ;RESTORE R1
3857  006250  000207                          RTS     PC              ;RETURN
3858
3859
3860
3861
3862
3863                                  ;*************************************************************************
3864                                  ;* ENDTRN - THIS SUBROUTINE CLEARS THE TRANSMITTER BY SETTING OC. THE PROGRAM
3865                                  ;*      WAITS FOR 50 US, AND CHECKS FOR ORDY=1, OCOR=0, OACT=0, RTS=0.
3866                                  ;*************************************************************************
3867  006252  010146                  ENDTRN: MOV     R1,-(SP)        ;SAVE R1
3868  006254  013746  002356                  MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
3869  006260  016637  000004  002330          MOV     4(SP),SUBRPC
3870  006266  162737  000004  002330          SUB     #4,SUBRPC       ;GET PC OF SUBROUTINE CALL
3871  006274  012737  000011  002356          MOV     #11,REGNUM      ;SET LU REG NO. = 11
3872  006302  012737  000200  002344          MOV     #OC,WRIBYT      ;SET OC IN DATA
3873  006310  004737  003726                  JSR     PC,WRITLU       ;SET OC IN REG 11
3874  006314  004737  005124                  JSR     PC,WAIT50       ;STALL FOR >50 US.
3875  006320  004737  004640                  JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=0
3876  006324  000001                          1
3877  006326  004737  005330                  JSR     PC,OACTIV       ;CHK OACT = 0
3878  006332  000000                          0
3879  006334  012737  000013  002356          MOV     #13,REGNUM      ;SET REG NO. = 13
3880  006342  004737  003650                  JSR     PC,READLU       ;READ REG 13
3881  006346  032737  000040  002342          BIT     #RTS,REDBYT     ;CHK FOR RTS = 0
3882  006354  001406                          BEQ     3$              ;BR IF RTS = 0
3883  006356  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3884                                  ;REPORT RTS NOT CLEARED
3885  006362                                  ERRDF   65,EM65,ERR4
3886  006362  104455                                                                  TRAP    C$ERDF
3887  006364  000101                                                                  .WORD   65
3888  006366  014222                                                                  .WORD   EM65
3889  006370  015600                                                                  .WORD   ERR4
3890  006372  005037  002330          3$:     CLR     SUBRPC          ;CLEAR SUBR CALL PC
3891  006376  012637  002356                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
3892  006402  012601                          MOV     (SP)+,R1        ;RESTORE R1
3893  006404  000207                          RTS     PC              ;RETURN
3894
3895
3896
3897
3898
3899                                  ;*************************************************************************
3900                                  ;* ISIRDY - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF ICIR (REG 17)
3901                                  ;*      AND IRDY (REG 12) AND REPORTS AN ERROR IF EITHER IS NOT PROPERLY SET
3902                                  ;*      AS PASSED IN BIT 0 (ICIR) AND BIT 1 (IRDY) OF THE WORD FOLLOWING THE
3903                                  ;*      CALL.
3904                                  ;*      IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS
3905                                  ;*      IN RETADR.
3906                                  ;*************************************************************************
3907  006406  013746  002356          ISIRDY: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
```

```
3908  006412  013746  002330                 MOV     SUBRPC,-(SP)
3909  006416  005737  002330                 TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
3910  006422  001006                         BNE     1$              ;BR IF YES
3911  006424  016637  000004  002330         MOV     4(SP),SUBRPC
3912  006432  162737  000004  002330         SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
3913  006440  012737  000012  002356  1$:    MOV     #12,REGNUM      ;SET REG NO. TO 12
3914  006446  004737  003650                 JSR     PC,READLU       ;READ REG 12
3915  006452  032776  000002  000004         BIT     #BIT1,@4(SP)    ;GET EXPECTED STATE OF IRDY
3916  006460  001413                         BEQ     3$              ;BR IF EXPECTED IRDY = 0
3917  006462  132737  000020  002342         BITB    #IRDY,REDBYT    ;SEE IF IRDY = 1
3918  006470  001022                         BNE     9$              ;BR IF IRDY = 1
3919  006472  004737  004504                 JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3920                                 ;REPORT IRDY NOT SET
3921  006476                                 ERRDF   17,EM17,ERR4
3922  006476  104455                                                                TRAP    C$ERDF
3923  006500  000021                                                                .WORD   17
3924  006502  012621                                                                .WORD   EM17
3925  006504  015600                                                                .WORD   ERR4
3926  006506  000451                         BR      16$             ;TAKE ERROR EXIT
3927  006510  132737  000020  002342  3$:    BITB    #IRDY,REDBYT    ;SEE IF IRDY = 0
3928  006516  001407                         BEQ     9$              ;BR IF IRDY = 0
3929  006520  004737  004504                 JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3930                                 ;REPORT IRDY NOT CLEARED
3931  006524                                 ERRDF   18,EM18,ERR4
3932  006524  104455                                                                TRAP    C$ERDF
3933  006526  000022                                                                .WORD   18
3934  006530  012636                                                                .WORD   EM18
3935  006532  015600                                                                .WORD   ERR4
3936  006534  000436                         BR      16$             ;TAKE ERROR RETURN
3937  006536  012737  000017  002356  9$:    MOV     #17,REGNUM      ;SET REG NO. = 17
3938  006544  004737  003650                 JSR     PC,READLU       ;READ REG 17
3939  006550  132776  000001  000004         BITB    #BIT0,@4(SP)    ;GET EXPECTED STATE OF ICIR
3940  006556  001413                         BEQ     12$             ;BR IF EXPECTED ICIR = 0
3941  006560  132737  000010  002342         BITB    #ICIR,REDBYT    ;SEE IF ICIR = 1
3942  006566  001031                         BNE     20$             ;BR IF ICIR = 1
3943  006570  004737  004504                 JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3944                                 ;REPORT ICIR NOT SET
3945  006574                                 ERRDF   19,EM19,ERR4
3946  006574  104455                                                                TRAP    C$ERDF
3947  006576  000023                                                                .WORD   19
3948  006600  012657                                                                .WORD   EM19
3949  006602  015600                                                                .WORD   ERR4
3950  006604  000412                         BR      16$             ;TAKE ERROR RETURN
3951  006606  132737  000010  002342  12$:    BITB    #ICIR,REDBYT    ;SEE IF ICIR = 0
3952  006614  001416                         BEQ     20$             ;BR IF ICIR = 0
3953  006616  004737  004504                 JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3954                                 ;REPORT ICIR NOT CLEARED
3955  006622                                 ERRDF   20,EM20,ERR4
3956  006622  104455                                                                TRAP    C$ERDF
3957  006624  000024                                                                .WORD   20
3958  006626  012674                                                                .WORD   EM20
3959  006630  015600                                                                .WORD   ERR4
3960  006632  016637  000002  002356  16$:    MOV     2(SP),REGNUM    ;RESTORE LU REG NO.
3961  006640  013706  002324                 MOV     PSTACK,SP       ;RESTORE STACK POINTER TO BASE LEVEL
3962  006644  013746  002340                 MOV     RETADR,-(SP)    ;FIX ERROR RETURN PC
3963  006650  000407                         BR      23$
```

```
3964  006652  062766  000002  000004  20$:    ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
3965  006660  012637  002330                  MOV     (SP)+,SUBRPC
3966  006664  012637  002356                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
3967  006670  000207                  23$:    RTS     PC              ;RETURN
3968
3969
3970
3971
3972
3973                                  ;*************************************************************************
3974                                  ;* IACTIV - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF IACT (REG 12) AND
3975                                  ;*      REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE STATE OF BIT 0 IN THE
3976                                  ;*      WORD FOLLOWING THE CALL.
3977                                  ;*      IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS IN
3978                                  ;*      RETADR.
3979                                  ;*************************************************************************
3980  006672  013746  002356          IACTIV: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
3981  006676  013746  002330                  MOV     SUBRPC,-(SP)
3982  006702  005737  002330                  TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
3983  006706  001006                          BNE     1$              ;BR IF YES
3984  006710  016637  000004  002330          MOV     4(SP),SUBRPC
3985  006716  162737  000004  002330          SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
3986  006724  012737  000012  002356  1$:     MOV     #12,REGNUM      ;SET REG NO. = 12
3987  006732  004737  003650                  JSR     PC,READLU       ;READ REG 12
3988  006736  032776  000001  000004          BIT     #BIT0,@4(SP)    ;GET EXPECTED STATE OF IACT
3989  006744  001413                          BEQ     3$              ;BR IF EXPECTED IACT = 0
3990  006746  132737  000100  002342          BITB    #IACT,REDBYT    ;SEE IF IACT = 1
3991  006754  001031                          BNE     9$              ;BR IF IACT = 1
3992  006756  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
3993                                  ;REPORT IACT NOT SET
3994  006762                                  ERRDF   21,EM21,ERR4
3995  006762  104455                                                                  TRAP    C$ERDF
3996  006764  000025                                                                  .WORD   21
3997  006766  012715                                                                  .WORD   EM21
3998  006770  015600                                                                  .WORD   ERR4
3999  006772  000412                          BR      6$              ;TAKE ERROR EXIT
4000  006774  132737  000100  002342  3$:     BITB    #IACT,REDBYT    ;SEE IF IACT = 0
4001  007002  001416                          BEQ     9$              ;BR IF IACT = 0
4002  007004  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4003                                  ;REPORT IACT NOT CLEARED
4004  007010                                  ERRDF   22,EM22,ERR4
4005  007010  104455                                                                  TRAP    C$ERDF
4006  007012  000026                                                                  .WORD   22
4007  007014  012732                                                                  .WORD   EM22
4008  007016  015600                                                                  .WORD   ERR4
4009  007020  016637  000002  002356  6$:     MOV     2(SP),REGNUM    ;RESTORE LU REG NO.
4010  007026  013706  002324                  MOV     PSTACK,SP       ;RESTORE PROGRAM STACK TO BASE LEVEL
4011  007032  013746  002340                  MOV     RETADR,-(SP)    ;FIX UP ERROR RETURN PC
4012  007036  000407                          BR      12$
4013  007040  062766  000002  000004  9$:     ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
4014  007046  012637  002330                  MOV     (SP)+,SUBRPC
4015  007052  012637  002356                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
4016  007056  000207                  12$:    RTS     PC              ;RETURN
4017
4018
4019
```

```
4020
4021
4022                                    ;*****************************************************************************
4023                                    ;* RSEOM - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF RSOM AND REOM IN
4024                                    ;* AX0-16, AND REPORTS AN ERROR IF EITHER IS NOT SET TO THE STATE PASSED IN BITS
4025                                    ;* 0,1, RESPECTIVELY, OF THE WORD FOLLOWING THE CALL.
4026                                    ;* IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS IN RETADR.
4027                                    ;*****************************************************************************
4028  007060  013746  002360           RSEOM:  MOV     AXNUM,-(SP)     ;SAVE AX BYTE NO.
4029  007064  013746  002330                   MOV     SUBRPC,-(SP)
4030  007070  005737  002330                   TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
4031  007074  001006                           BNE     1$              ;BR IF YES
4032  007076  016637  000004  002330           MOV     4(SP),SUBRPC
4033  007104  162737  000004  002330           SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
4034  007112  012737  000001  002360  1$:      MOV     #1,AXNUM        ;SET AX BYTE NO. FOR AX0-16
4035  007120  004737  004102                   JSR     PC,READAX       ;READ AX0
4036  007124  032776  000001  000004           BIT     #BIT0,@4(SP)    ;GET EXPECTED STATE OF RSOM
4037  007132  001413                           BEQ     3$              ;BR IF EXPECTED RSOM = 0
4038  007134  132737  000001  002350           BITB    #RSOM,RAX16     ;SEE IF RSOM = 1
4039  007142  001022                           BNE     9$              ;BR IF RSOM = 1
4040  007144  004737  004504                   JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4041                                    ;REPORT RSOM NOT SET
4042  007150                                    ERRDF   29,EM29,ERR6
4043  007150  104455                                                                   TRAP    C$ERDF
4044  007152  000035                                                                   .WORD   29
4045  007154  013151                                                                   .WORD   EM29
4046  007156  016770                                                                   .WORD   ERR6
4047  007160  000444                           BR      16$             ;TAKE ERROR EXIT
4048  007162  132737  000001  002350  3$:      BITB    #RSOM,RAX16     ;SEE IF RSOM = 0
4049  007170  001407                           BEQ     9$              ;BR IF RSOM = 0
4050  007172  004737  004504                   JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4051                                    ;REPORT RSOM NOT CLEARED
4052  007176                                    ERRDF   28,EM28,ERR6
4053  007176  104455                                                                   TRAP    C$ERDF
4054  007200  000034                                                                   .WORD   28
4055  007202  013130                                                                   .WORD   EM28
4056  007204  016770                                                                   .WORD   ERR6
4057  007206  000431                           BR      16$             ;TAKE ERROR RETURN
4058  007210  132776  000002  000004  9$:      BITB    #BIT1,@4(SP)    ;GET EXPECTED STATE OF REOM
4059  007216  001413                           BEQ     12$             ;BR IF EXPECTED REOM = 0
4060  007220  132737  000002  002350           BITB    #REOM,RAX16     ;SEE IF REOM = 1
4061  007226  001031                           BNE     20$             ;BR IF REOM = 1
4062  007230  004737  004504                   JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4063                                    ;REPORT REOM NOT SET
4064  007234                                    ERRDF   31,EM31,ERR6
4065  007234  104455                                                                   TRAP    C$ERDF
4066  007236  000037                                                                   .WORD   31
4067  007240  013207                                                                   .WORD   EM31
4068  007242  016770                                                                   .WORD   ERR6
4069  007244  000412                           BR      16$             ;TAKE ERROR RETURN
4070  007246  132737  000002  002350  12$:     BITB    #REOM,RAX16     ;SEE IF REOM = 0
4071  007254  001416                           BEQ     20$             ;BR IF REOM = 0
4072  007256  004737  004504                   JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4073                                    ;REPORT REOM NOT CLEARED
4074  007262                                    ERRDF   30,EM30,ERR6
4075  007262  104455                                                                   TRAP    C$ERDF
```

```
4076  007264  000036                                                          .WORD   30
4077  007266  013166                                                          .WORD   EM30
4078  007270  016770                                                          .WORD   ERR6
4079  007272  016637  000002  002360  16$:    MOV     2(SP),AXNUM     ,RESTORE AX BYTE NO.
4080  007300  013706  002324                  MOV     PSTACK,SP       ;RESTORE STACK POINTER TO BASE LEVEL
4081  007304  013746  002340                  MOV     RETADR,-(SP)    ;FIX ERROR RETURN PC
4082  007310  000407                          BR      23$
4083  007312  062766  000002  000004  20$:    ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
4084  007320  012637  002330                  MOV     (SP)+,SUBRPC
4085  007324  012637  002360                  MOV     (SP)+,AXNUM     ;RESTORE AX BYTE NO.
4086  007330  000207                  23$:    RTS     PC              ;RETURN
4087
4088
4089
4090
4091
4092                                  ;*************************************************************************
4093                                  ;* RDRXSI - THIS SUBROUTINE READS THE RCV SILO (REGS 10,12) AND RETURNS THE
4094                                  ;*      SILO ENTRY IN BITS 0-11 OF RXWORD.
4095                                  ;*************************************************************************
4096  007332  013746  002356          RDRXSI: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
4097  007336  012737  000012  002356          MOV     #12,REGNUM      ;SET REG NO. = 12
4098  007344  004737  003650                  JSR     PC,READLU       ;READ LU REG 12
4099  007350  113737  002342  002403          MOVB    REDBYT,RXWORD+1 ;GET HI BITS OF SILO ENTRY
4100  007356  042737  170000  002402          BIC     #170000,RXWORD  ;CLEAR UNUSED BITS
4101  007364  012737  000010  002356          MOV     #10,REGNUM      ;SET REG NO. = 10
4102  007372  004737  003650                  JSR     PC,READLU       ;READ REG 10
4103  007376  113737  002342  002402          MOVB    REDBYT,RXWORD   ;GET LOW BITS OF SILO ENTRY
4104  007404  012637  002356                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
4105  007410  000207                          RTS     PC              ;RETURN
4106
4107
4108
4109
4110
4111                                  ;*************************************************************************
4112                                  ;* RCV1ST - THIS SUBROUTINE RECEIVES THE FIRST CHAR OF A MESSAGE, AND MONITORS
4113                                  ;*      STATUS OF THE RECEIVER. FIRST, A CHECK IS MADE FOR IACT = 0, IRDY = 0,
4114                                  ;*      ICIR = 1, AND RSOM = 0. THEN, THE LINE UNIT IS CLOCKED USING
4115                                  ;*      STEPLU UNTIL IRDY = 1. THE PROGRAM CHECKS FOR THIS TO OCCUR WITHIN 3
4116                                  ;*      CYCLES AFTER THE NO. OF CYCLES PASSED IN THE WORD FOLLOWING THE CALL.
4117                                  ;*      IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST, AT THE ADDRESS
4118                                  ;*      CONTAINED IN RETADR.
4119                                  ;*************************************************************************
4120  007412  010146          RCV1ST: MOV     R1,-(SP)        ;SAVE R1
4121  007414  010346                  MOV     R3,-(SP)        ;SAVE R3
4122  007416  013746  002356          MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
4123  007422  016637  000006  002330  MOV     6(SP),SUBRPC
4124  007430  162737  000004  002330  SUB     #4,SUBRPC       ;GET PC OF SUBROUTINE CALL
4125  007436  012737  000012  002356  MOV     #12,REGNUM      ;SET LU REG NO. = 12
4126  007444  005001                  CLR     R1              ;INIT CYCLE COUNT TO 0
4127  007446  017603  000006          MOV     @6(SP),R3       ;GET CYCLE COUNT LIMIT
4128  007452  062703  000003          ADD     #3,R3
4129  007456  005776  000006          TST     @6(SP)          ;SEE IF DESIRED CYCLES = 0
4130  007462  001414                  BEQ     8$              ;BR IF YES
4131  007464  004737  006672          JSR     PC,IACTIV       ;CHK FOR IACT = 0
```

```
4132  007470  000000                          0
4133  007472  004737  006406                  JSR     PC,ISIRDY       ;CHK FOR ICIR = 1, IRDY = 0
4134  007476  000001                          1
4135  007500  004737  007060                  JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0 IN AX0-16
4136  007504  000000                          0
4137  007506  004737  005232          6$:     JSR     PC,STPLU        ;CLOCK LU FOR 1 CYCLE
4138  007512  000001                          1
4139  007514  004737  005124          8$:     JSR     PC,WAIT50       ;ALLOW SILO DATA TO RIPPLE
4140  007520  005201                          INC     R1              ;INCREMENT CYCLE COUNT
4141  007522  004737  003650                  JSR     PC,READLU       ;READ REG 12
4142  007526  132737  000020  002342          BITB    #IRDY,REDBYT    ;SEE IF IRDY = 1 YET
4143  007534  001005                          BNE     9$              ;BR IF IRDY = 1
4144  007536  020103                          CMP     R1,R3           ;SEE IF LIMIT EXCEEDED
4145  007540  002762                          BLT     6$              ;BR IF NOT YET
4146  007542  004737  006406                  JSR     PC,ISIRDY       ;CHK FOR ICIR = 1, IRDY = 1
4147  007546  000003                          3
4148  007550  020176  000006          9$:     CMP     R1,@6(SP)       ;SEE IF LESS THAN REQUIRED CYCLES
4149  007554  002003                          BGE     12$             ;BR IF NOT
4150  007556  004737  006406                  JSR     PC,ISIRDY       ;CHK FOR ICIR = 1, IRDY = 0
4151  007562  000001                          1
4152  007564  004737  006672          12$:    JSR     PC,IACTIV       ;CHK FOR IACT = 1
4153  007570  000001                          1
4154  007572  004737  006406                  JSR     PC,ISIRDY       ;CHK FOR ICIR = 1, IRDY = 1
4155  007576  000003                          3
4156  007600  062766  000002  000006          ADD     #2,6(SP)        ;FIX UP RETURN PC
4157  007606  012637  002356                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
4158  007612  012603                          MOV     (SP)+,R3        ;RESTORE R3
4159  007614  012601                          MOV     (SP)+,R1        ;RESTORE R1
4160  007616  005037  002330                  CLR     SUBRPC          ;CLEAR SUBR CALL PC
4161  007622  000207                          RTS     PC              ;RETURN
4162
4163
4164
4165
4166
4167                                  ;*****************************************************************************
4168                                  ;* STPERR - THIS SUBROUTINE LOADS THE CONTENTS OF THE FIRST WORD FOLLOWING THE
4169                                  ;*      CALL INTO REG 17, AND SETS THE IERR BIT, AND CLOCKS THE LINE UNIT
4170                                  ;*      FOR THE NO. OF CYCLES PASSED IN THE 2ND WORD FOLLOWING THE CALL. THEN,
4171                                  ;*      IT RESTORES REG 17 TO ITS ORIGINAL CONTENTS, CLEARING THE IERR BIT.
4172                                  ;*****************************************************************************
4173  007624  013746  002356          STPERR: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
4174  007630  012737  000017  002356          MOV     #17,REGNUM      ;SET LU REG NO. = 17
4175  007636  017637  000002  002344          MOV     @2(SP),WRIBYT
4176  007644  152737  000002  002344          BISB    #IERR,WRIBYT
4177  007652  004737  003726                  JSR     PC,WRITLU       ;SET IERR BIT IN REG 17
4178  007656  062766  000002  000002          ADD     #2,2(SP)        ;INCREMENT SUBR ARGUMENT POINTER
4179  007664  017637  000002  007676          MOV     @2(SP),3$       ;GET DESIRED NO. OF CYCLES
4180  007672  004737  005232                  JSR     PC,STPLU        ;CLOCK LU FOR DESIRED NO. OF CYCLES
4181  007676  000000                  3$:     .WORD   0               ;NO. OF CYCLES GOES HERE
4182  007700  142737  000002  002344          BICB    #IERR,WRIBYT
4183  007706  004737  003726                  JSR     PC,WRITLU       ;CLEAR IERR BIT IN REG 17
4184  007712  062766  000002  000002          ADD     #2,2(SP)        ;FIX UP RETURN PC
4185  007720  012637  002356                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
4186  007724  000207                          RTS     PC              ;RETURN
4187
```

```
4188
4189
4190
4191
4192                                    ;*****************************************************************************
4193                                    ;* CKDATA - THIS SUBROUTINE READS THE RCV SILO AND COMPARES THE SILO ENTRY
4194                                    ;*      TO BITS 0-11 OF THE FIRST WORD FOLLOWING THE CALL. IF THERE IS A
4195                                    ;*      MISMATCH, THE ERROR IS REPORTED AND A RETURN IS MADE TO THE TEST AT THE
4196                                    ;*      ADDRESS CONTAINED IN RETADR. IF BIT 15 = 0 IN THE FIRST WORD
4197                                    ;*      FOLLOWING THE CALL, THE SUBROUTINE WILL NOT CHECK THE BCC BIT (SILO
4198                                    ;*      BIT 8). IF THERE ARE NO ERRORS, THE LINE UNIT IS CLOCKED FOR THE
4199                                    ;*      NUMBER OF CYCLES PASSED IN THE SECOND WORD FOLLOWING THE CALL.
4200                                    ;*****************************************************************************
4201  007726  010146                    CKDATA: MOV     R1,-(SP)        ;SAVE R1
4202  007730  013746  002356                    MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
4203  007734  016637  000004  002330            MOV     4(SP),SUBRPC
4204  007742  162737  000004  002330            SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
4205  007750  017601  000004                    MOV     @4(SP),R1       ;GET EXPECTED SILO ENTRY
4206  007754  042701  170000                    BIC     #170000,R1      ;CLEAR UNUSED BITS FOR COMPARE
4207  007760  004737  007332                    JSR     PC,RDRXSI       ;READ RCV SILO
4208  007764  023727  002410  000347            CMP     SAVLEN,#TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0
4209  007772  001005                            BNE     4$              ;BR IF CHAR LENGTH NOT = 7
4210  007774  042701  000200                    BIC     #BIT7,R1        ;MASK OFF BIT 8TH BIT
4211  010000  042737  000200  002402            BIC     #BIT7,RXWORD
4212  010006  120137  002402          4$:       CMPB    R1,RXWORD       ;COMPARE EXPECTED BITS 0-7 TO ACTUAL
4213  010012  001445                            BEQ     6$              ;BR IF MATCH
4214  010014  005037  002362                    CLR     GOODAT
4215  010020  110137  002362                    MOVB    R1,GOODAT       ;GET EXPECTED DATA
4216  010024  005037  002364                    CLR     BADDAT
4217  010030  113737  002402  002364            MOVB    RXWORD,BADDAT   ;GET ACTUAL DATA
4218  010036  012737  000011  002356            MOV     #11,REGNUM      ;SET REG NO. = 11
4219  010044  004737  003650                    JSR     PC,READLU       ;READ REG 11
4220  010050  132737  000001  002342            BITB    #UNRR,REDBYT    ;SEE IF TX UNDERRUN ERROR
4221  010056  001410                            BEQ     5$              ;BR IF NOT
4222  010060  004737  004504                    JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4223                                  ;REPORT TX UNDERRUN ERROR
4224  010064                                    ERRDF   54,EM54,ERR4
4225  010064  104455                                                                    TRAP    C$ERDF
4226  010066  000066                                                                    .WORD   54
4227  010070  014164                                                                    .WORD   EM54
4228  010072  015600                                                                    .WORD   ERR4
4229  010074  000137  010556                    JMP     36$             ;TAKE ERROR EXIT
4230  010100  012737  000010  002356  5$:       MOV     #10,REGNUM      ;SET REG NO. = 10
4231  010106  004737  004504                    JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4232                                  ;REPORT RCV'D DATA MISCOMPARE
4233  010112                                    ERRDF   34,EM34,ERR8
4234  010112  104455                                                                    TRAP    C$ERDF
4235  010114  000042                                                                    .WORD   34
4236  010116  013276                                                                    .WORD   EM34
4237  010120  020100                                                                    .WORD   ERR8
4238  010122  000137  010556                    JMP     36$             ;TAKE ERROR EXIT
4239  010126  000301                    6$:     SWAB    R1
4240  010130  012737  000012  002356            MOV     #12,REGNUM      ;SET LU REG NO. FOR ERROR REPORTS
4241  010136  120137  002403                    CMPB    R1,RXWORD+1     ;COMPARE EXPECTED SILO BITS 8-11 TO ACTUAL
4242  010142  001002                            BNE     7$              ;BR IF MISMATCH
4243  010144  000137  010532                    JMP     22$             ;CONTINUE
```

```
4244  010150  005037  002362          7$:     CLR     GOODAT
4245  010154  110137  002362                  MOVB    R1,GOODAT       ;SET EXPECTED DATA
4246  010160  005037  002364                  CLR     BADDAT
4247  010164  113737  002403  002364          MOVB    RXWORD+1,BADDAT ;SET ACTUAL DATA '
4248  010172  032776  100000  000004          BIT     #BCCCHK,@4(SP)  ;SEE IF BCC SHOULD BE IGNORED
4249  010200  001433                          BEQ     10$             ;BR IF YES
4250  010202  132701  000001                  BITB    #BCC,R1         ;SEE IF EXPECTED BIT = 1
4251  010206  001014                          BNE     8$              ;BR IF YES
4252  010210  132737  000001  002403          BITB    #BCC,RXWORD+1   ;SEE IF ACTUAL BIT = 0
4253  010216  001424                          BEQ     10$             ;BR IF YES
4254  010220  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4255                                  ;REPORT BCC NOT CLEARED
4256  010224                                  ERRDF   35,EM35,ERR8
4257  010224  104455                                                                  TRAP    C$ERDF
4258  010226  000043                                                                  .WORD   35
4259  010230  013324                                                                  .WORD   EM35
4260  010232  020100                                                                  .WORD   ERR8
4261  010234  000137  010556                  JMP     36$             ;TAKE ERROR EXIT
4262  010240  132737  000001  002403  8$:     BITB    #BCC,RXWORD+1   ;SEE IF ACTUAL BIT = 1
4263  010246  001010                          BNE     10$             ;BR IF YES
4264  010250  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4265                                  ;REPORT BCC NOT SET
4266  010254                                  ERRDF   36,EM36,ERR8
4267  010254  104455                                                                  TRAP    C$ERDF
4268  010256  000044                                                                  .WORD   36
4269  010260  013344                                                                  .WORD   EM36
4270  010262  020100                                                                  .WORD   ERR8
4271  010264  000137  010556                  JMP     36$             ;TAKE ERROR EXIT
4272  010270                          10$:
4273  010270  132701  000002                  BITB    #EBLK,R1        ;SEE IF EXPECTED BIT = 1
4274  010274  001014                          BNE     12$             ;BR IF YES
4275  010276  132737  000002  002403          BITB    #EBLK,RXWORD+1  ;SEE IF ACTUAL BIT = 0
4276  010304  001424                          BEQ     14$             ;BR IF YES
4277  010306  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4278                                  ;REPORT EBLK NOT CLEARED
4279  010312                                  ERRDF   37,EM37,ERR8
4280  010312  104455                                                                  TRAP    C$ERDF
4281  010314  000045                                                                  .WORD   37
4282  010316  013360                                                                  .WORD   EM37
4283  010320  020100                                                                  .WORD   ERR8
4284  010322  000137  010556                  JMP     36$             ;TAKE ERROR EXIT
4285  010326  132737  000002  002403  12$:     BITB    #EBLK,RXWORD+1  ;SEE IF ACTUAL BIT = 1
4286  010334  001010                          BNE     14$             ;BR IF YES
4287  010336  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4288                                  ;REPORT EBLK NOT SET
4289  010342                                  ERRDF   38,EM38,ERR8
4290  010342  104455                                                                  TRAP    C$ERDF
4291  010344  000046                                                                  .WORD   38
4292  010346  013401                                                                  .WORD   EM38
4293  010350  020100                                                                  .WORD   ERR8
4294  010352  000137  010556                  JMP     36$             ;TAKE ERROR EXIT
4295  010356                          14$:
4296  010356  132701  000004                  BITB    #RAB,R1         ;SEE IF EXPECTED BIT = 1
4297  010362  001014                          BNE     16$             ;BR IF YES
4298  010364  132737  000004  002403          BITB    #RAB,RXWORD+1   ;SEE IF ACTUAL BIT = 0
4299  010372  001424                          BEQ     18$             ;BR IF YES
```

CZDMRF.P11 03-NOV-81 10:29 GLOBAL SUBROUTINES

```
4300  010374  004737  004504                    JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4301                                    ;REPORT RAB NOT CLEARED
4302  010400                                    ERRDF   39,EM39,ERR8
4303  010400  104455                                                                    TRAP    C$ERDF
4304  010402  000047                                                                    .WORD   39
4305  010404  013416                                                                    .WORD   EM39
4306  010406  020100                                                                    .WORD   ERR8
4307  010410  000137  010556                    JMP     36$             ;TAKE ERROR EXIT
4308  010414  132737  000004  002403  16$:     BITB    #RAB,RXWORD+1   ;SEE IF ACTUAL BIT = 1
4309  010422  001010                            BNE     18$             ;BR IF YES
4310  010424  004737  004504                    JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4311                                    ;REPORT RAB NOT SET
4312  010430                                    ERRDF   40,EM40,ERR8
4313  010430  104455                                                                    TRAP    C$ERDF
4314  010432  000050                                                                    .WORD   40
4315  010434  013436                                                                    .WORD   EM40
4316  010436  020100                                                                    .WORD   ERR8
4317  010440  000137  010556                    JMP     36$             ;TAKE ERROR EXIT
4318  010444                            18$:
4319  010444  132701  000010                    BITB    #OVRR,R1        ;SEE IF EXPECTED BIT = 1
4320  010450  001014                            BNE     20$             ;BR IF YES
4321  010452  132737  000010  002403            BITB    #OVRR,RXWORD+1  ;SEE IF ACTUAL BIT = 0
4322  010460  001424                            BEQ     22$             ;BR IF YES
4323  010462  004737  004504                    JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4324                                    ;REPORT OVRR NOT CLEARED
4325  010466                                    ERRDF   41,EM41,ERR8
4326  010466  104455                                                                    TRAP    C$ERDF
4327  010470  000051                                                                    .WORD   41
4328  010472  013452                                                                    .WORD   EM41
4329  010474  020100                                                                    .WORD   ERR8
4330  010476  000137  010556                    JMP     36$             ;TAKE ERROR EXIT
4331  010502  132737  000010  002403  20$:     BITB    #OVRR,RXWORD+1  ;SEE IF ACTUAL BIT = 1
4332  010510  001010                            BNE     22$             ;BR IF YES
4333  010512  004737  004504                    JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4334                                    ;REPORT OVRR NOT SET
4335  010516                                    ERRDF   42,EM42,ERR8
4336  010516  104455                                                                    TRAP    C$ERDF
4337  010520  000052                                                                    .WORD   42
4338  010522  013473                                                                    .WORD   EM42
4339  010524  020100                                                                    .WORD   ERR8
4340  010526  000137  010556                    JMP     36$             ;TAKE ERROR EXIT
4341  010532                            22$:
4342  010532  062766  000002  000004            ADD     #2,4(SP)        ;INCR SUBROUTINE ARGUMENT POINTER
4343  010540  017637  000004  010552            MOV     @4(SP),24$      ;GET DESIRED CYCLE COUNT
4344  010546  004737  005232                    JSR     PC,STPLU        ;CLOCK LU FOR DESIRED CYCLES
4345  010552  000000                    24$:    .WORD   0
4346  010554  000407                            BR      38$             ;TAKE ERROR-FREE EXIT
4347  010556  011637  002356            36$:    MOV     (SP),REGNUM     ;RESTORE LU REG NO.
4348  010562  013706  002324                    MOV     PSTACK,SP       ;RESTORE PROGRAM STACK TO BASE LEVEL
4349  010566  013746  002340                    MOV     RETADR,-(SP)    ;FIX UP ERROR RETURN PC
4350  010572  000406                            BR      40$
4351  010574  062766  000002  000004  38$:     ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
4352  010602  012637  002356                    MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
4353  010606  012601                            MOV     (SP)+,R1        ;RESTORE R1
4354  010610  005037  002330            40$:    CLR     SUBRPC          ;CLEAR SUBROUTINE PC
4355  010614  000207                            RTS     PC              ;RETURN
```

```
                                        ;*****************************************************************************
                                        ;* SETUP - THIS SUBROUTINE LOADS THE FIRST WORD AFTER THE CALL INTO AX2-15
                                        ;*       (SYNCH CHAR), LOADS THE SECOND WORD AFTER THE CALL INTO REG 17
                                        ;*       LOADS THE THIRD WORD INTO AX3-15, AND LOADS THE FOURTH INTO AX3-16.
                                        ;*****************************************************************************
010616  013746  002360          SETUP:  MOV     AXNUM,-(SP)     ;SAVE AX BYTE NO.
010622  013746  002356                  MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
010626  012737  000004  002360          MOV     #4,AXNUM        ;SET AX BYTE NO. FOR AX2
010634  017637  000004  002352          MOV     @4(SP),WAX15
010642  005037  002354                  CLR     WAX16
010646  004737  004270                  JSR     PC,WRITAX       ;SET SYNCH CHAR IN AX2-15, CLEAR AX2-16
010652  012737  000017  002356          MOV     #17,REGNUM      ;SET LU REG NO. = 17
010660  062766  000002  000004          ADD     #2,4(SP)        ;INCREMENT ARGUMENT POINTER
010666  017637  000004  002344          MOV     @4(SP),WRIBYT
010674  004737  003726                  JSR     PC,WRITLU       ;LOAD REG 17
010700  012737  000006  002360          MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX3
010706  062766  000002  000004          ADD     #2,4(SP)        ;INCREMENT ARGUMENT POINTER
010714  017637  000004  002352          MOV     @4(SP),WAX15
010722  062766  000002  000004          ADD     #2,4(SP)        ;INCR ARGUMENT POINTER
010730  017637  000004  002354          MOV     @4(SP),WAX16
010736  013737  002354  002410          MOV     WAX16,SAVLEN    ;STORE TX AND RCV CHAR LENGTH
010744  004737  004270                  JSR     PC,WRITAX       ;LOAD AX3-15, AX3-16
010750  062766  000002  000004          ADD     #2,4(SP)        ;FIX RETURN PC
010756  012637  002356                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
010762  012637  002360                  MOV     (SP)+,AXNUM     ;RESTORE AX BYTE NO.
010766  005037  002330                  CLR     SUBRPC          ;CLEAR SUBROUTINE PC STORAGE
010772  000207                          RTS     PC              ;RETURN


                                        ;*****************************************************************************
                                        ;* LODMSG - THIS SUBROUTINE LOADS THE NO. OF WORDS PASSED IN THE SECOND WORD
                                        ;*       FOLLOWING THE CALL FROM THE MSG BUFFER WHOSE ADDRESS IS IN THE FIRST
                                        ;*       WORD FOLLOWING THE CALL, INTO THE TRANSMITTER SILO.
                                        ;*****************************************************************************
010774  010146          LODMSG: MOV     R1,-(SP)        ;SAVE R1
010776  010246                  MOV     R2,-(SP)        ;SAVE R2
011000  017601  000004          MOV     @4(SP),R1       ;GET MSG POINTER INTO R1
011004  062766  000002  000004  ADD     #2,4(SP)        ;INCR ARG POINTER
011012  017602  000004          MOV     @4(SP),R2       ;GET WORD COUNT INTO R2
011016  062766  000002  000004  ADD     #2,4(SP)        ;FIX UP RETURN PC
011024  012137  002400  6$:     MOV     (R1)+,TXWORD    ;GET NEXT MSG WORD
011030  004737  005152          JSR     PC,LDTXSI       ;LOAD A WORD INTO TX SILO
011034  005302                  DEC     R2              ;DECR COUNT
011036  001372                  BNE     6$              ;BR IF NOT DONE YET
011040  004737  005124          JSR     PC,WAIT50       ;WAIT FOR SILO TO RIPPLE
011044  012602                  MOV     (SP)+,R2        ;RESTORE R2
011046  012601                  MOV     (SP)+,R1        ;RESTORE R1
011050  000207                  RTS     PC              ;RETURN
```

```
                                        ;*************************************************************************
                                        ;* RXCHAR - THIS SUBROUTINE READS THE RCV SILO AND CLOCKS THE LINE UNIT.
                                        ;*      FIRST, IT READS THE CHAR FROM THE RCV SILO, AND CHECKS FOR ICIR
                                        ;*      = 1, IRDY = 0. IT THEN CLOCKS THE LINE UNIT FOR THE NO. OF CYCLES
                                        ;*      PASSED IN THE WORD FOLLOWING THE CALL, AND THEN CHECKS FOR ICIR
                                        ;*      = 1, IRDY = 1.
                                        ;*************************************************************************
011052  010146                  RXCHAR: MOV     R1,-(SP)        ;SAVE R1
011054  016637  000002  002330          MOV     2(SP),SUBRPC
011062  162737  000004  002330          SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
011070  004737  007332                  JSR     PC,RDRXSI       ;READ RCV SILO
011074  004737  005124                  JSR     PC,WAIT50       ;ALLOW SILO TO RIPPLE
011100  005001                          CLR     R1              ;INIT CYCLE COUNT
011102  020176  000002          9$:     CMP     R1,@2(SP)       ;SEE IF REQUIRED CYCLES DONE YET
011106  001410                          BEQ     12$             ;BR IF YES
011110  004737  006406                  JSR     PC,ISIRDY       ;CHK ICIR = 1, IRDY = 0
011114  000001                          1
011116  004737  005232                  JSR     PC,STPLU        ;CLK LU 1 CYCLE
011122  000001                          1
011124  005201                          INC     R1              ;INCR CYCLE COUNT
011126  000765                          BR      9$
011130  004737  006406          12$:    JSR     PC,ISIRDY       ;CHK ICIR = 1 IRDY = 1
011134  000003                          3
011136  062766  000002  000002          ADD     #2,2(SP)        ;FIX RETURN PC
011144  005037  002330                  CLR     SUBRPC          ;CLEAR SUBR CALL PC
011150  012601                          MOV     (SP)+,R1        ;RESTORE R1
011152  000207                          RTS     PC              ;RETURN
                                        ;*************************************************************************
                                        ;* CKTBIT - THIS SUBROUTINE CLOCKS THE LINE UNIT FOR 8 CYCLES, AND AFTER EACH
                                        ;*      CYCLE, THE TXDATA BIT IN REG 17 IS EXAMINED, AND COMPARED TO A BIT OF
                                        ;*      THE CHAR PASSED IN THE WORD FOLLOWING THE CALL.
                                        ;*      IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS IN
                                        ;*      RETADR.
                                        ;*************************************************************************
011154  013746  002356          CKTBIT: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
011160  010146                          MOV     R1,-(SP)        ;SAVE R1
011162  016637  000004  002330          MOV     4(SP),SUBRPC
011170  162737  000004  002330          SUB     #4,SUBRPC       ;GET PC OF SUBROUTINE CALL
011176  012701  000001                  MOV     #BIT0,R1        ;INIT BIT POINTER
011202  012737  000017  002356          MOV     #17,REGNUM      ;SET REG NO. = 17
011210  004737  005232          4$:     JSR     PC,STPLU        ;CLOCK LINE UNIT 1 CYCLE
011214  000001                          1
011216  004737  003650                  JSR     PC,READLU       ;READ REG 17
011222  030176  000004                  BIT     R1,@4(SP)       ;SEE IF EXPECTED BIT = 1
011226  001013                          BNE     9$              ;BR IF YES
011230  132737  000040  002342          BITB    #TXDATA,REDBYT  ;SEE IF ACTUAL BIT = 0
```

CZDMRF.P11 03-NOV-81 10:29 GLOBAL SUBROUTINES

```
4468  011236  001422                          BEQ     12$             ;BR IF YES
4469  011240  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4470                                  ;REPORT TXDATA NOT CLEARED
4471  011244                                  ERRDF   32,EM32,ERR4
4472  011244  104455                                                                  TRAP    C$ERDF
4473  011246  000040                                                                  .WORD   32
4474  011250  013224                                                                  .WORD   EM32
4475  011252  015600                                                                  .WORD   ERR4
4476  011254  000420                          BR      20$             ;TAKE ERROR RETURN
4477  011256  132737  000040  002342  9$:     BITB    #TXDATA,REDBYT  ;SEE IF ACTUAL BIT = 1
4478  011264  001007                          BNE     12$             ;BR IF YES
4479  011266  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
4480                                  ;REPORT TXDATA BIT NOT SET
4481  011272                                  ERRDF   33,EM33,ERR4
4482  011272  104455                                                                  TRAP    C$ERDF
4483  011274  000041                                                                  .WORD   33
4484  011276  013253                                                                  .WORD   EM33
4485  011300  015600                                                                  .WORD   ERR4
4486  011302  000405                          BR      20$             ,TAKE ERROR EXIT
4487  011304  006301                  12$:    ASL     R1              ;SHIFT BIT POINTER
4488  011306  020127  000400                  CMP     R1,#400         ;SEE IF 8 BITS SCANNED YET
4489  011312  001336                          BNE     4$              ;BR IF NO
4490  011314  000405                          BR      22$
4491  011316  013706  002324          20$:    MOV     PSTACK,SP       ;RESTORE PROGRAM STACK POINTER TO BASE LEVEL
4492  011322  013746  002340                  MOV     RETADR,-(SP)    ;FIX UP RETURN PC
4493  011326  000406                          BR      40$
4494  011330  062766  000002  000004  22$:    ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
4495  011336  012601                          MOV     (SP)+,R1        ;RESTORE R1
4496  011340  012637  002356                  MOV     (SP)+,REGNUM    ;RESTORE REG NO.
4497  011344  005037  002330          40$:    CLR     SUBRPC          ;CLEAR SUBR CALL PC
4498  011350  000207                          RTS     PC              ;RETURN
4499
4500
4501
4502
4503
4504                                  ;*****************************************************************************
4505                                  ;* LDMSG1 - THIS SUBROUTINE LOADS THE TRANSMITTER SILO WITH MSG1, AND LOADS
4506                                  ;*       THE DATA CHARS INTO THE RCV MSG BUFFER (RCVBUF:), AS EXPECTED DATA
4507                                  ;*       FOR LATER COMPARISON.
4508                                  ;*****************************************************************************
4509  011352  010146          LDMSG1: MOV     R1,-(SP)        ;SAVE R1
451U  011354  010246                  MOV     R2,-(SP)        ;SAVE R2
4511  011356  004737  010774                  JSR     PC,LODMSG       ;LOAD MSG1 INTO TX SILO
4512  011362  003176                          MSG1
4513  011364  000011                          9.
4514  011366  012701  003202                  MOV     #MSG1+4,R1      ;GET POINTER TO MSG1
4515  011372  012702  003240                  MOV     #RCVBUF,R2      ;GET POINTER TO MSG BUF
4516  011376  012122          3$:     MOV     (R1)+,(R2)+     ;LOAD A CHAR INTO MSG BUF
4517  011400  020127  003214                  CMP     R1,#MSG1+14.    ;SEE IF DID LAST DATA CHAR YET
4518  011404  103774                          BLO     3$              ;BR IF NOT
4519  011406  052762  100400  177776          BIS     #CRCCHK!RXBCC,-2(R2) ;SET EXPECTED BCC
4520  011414  012726  000160                  MOV     #160,(SP)+      ;LOAD HI CRC BYTE
4521  011420  012726  000034                  MOV     #034,(SP)+      ;LOAD LO CRC BYTE
4522  011424  012602                          MOV     (SP)+,R2        ;RESTORE R2
4523  011426  012601                          MOV     (SP)+,R1        ;RESTORE R1
```

```
 4524  011430  000207                    RTS     PC              ;RETURN
 4525
 4526
 4527
 4528
 4529
```

```
                                .SBTTL  GLOBAL ERROR REPORT SECTION

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

011432  052045  047445  022466  FMT1:   .ASCIZ  /%T%O6%N/
011440  000*16
011442  047045  040445  040506  FMT2:   .ASCIZ  /%N%AFAILING REG: /
011450  046111  047111  020107
011456  042522  035107  000040
011464  040445  054105  042520  FMT3:   .ASC*2  /%AEXPECTED: %O3%S5%AACTUAL: %O3%N/
011472  052103  042105  020072
011500  047445  022463  032523
011506  040445  041501  052524
011514  046101  020072  047445
011522  022463  000116
011526  047045  052045  047045  FMT4:   .ASCIZ  /%N%T%N%T%N/
011534  052045  047045     000
011541     045  031517  051445  FMT5:   .ASCIZ  /%O3%S5%O3%S5%O3%S5%O3%N/
011546  022465  031517  051445
011554  022465  031517  051445
011562  022465  031517  047045
011570     000
011571     045  032123  047445  FMT6:   .ASCIZ  /%S4%O3%S5%O3%S5%O3%S5%O3%N/
011576  022463  032523  047445
011604  022463  032523  047445
011612  022463  032523  047445
011620  022463  000116
011624  052045  047445  022462  FMT7:   .ASCIZ  /%T%O2%N/
011632  000116
011634  040445  054105  042524  FMT8:   .ASCIZ  /%AEXTENDED REG AX%O1%A-%T%N/
011642  042116  042105  051040
011650  043505  040440  022530
011656  030517  040445  022455
011664  022524  000116
011670  052045  047045     000  FMT9:   .ASCIZ  /%T%N/
011675     045  050101  020103  FMT10:  .ASCIZ  /%APC OF SUBR CALL: %O6%N/
011702  043117  051440  041125
011710  020122  040503  046114
011716  020072  047445  022466
011724  000116
011726  040445  042522  020107  FMT11:  .ASCIZ  /%AREG %O2%A LOADED WITH: %O3%N/
011734  047445  022462  020101
011742  047514  042101  042105
011750  053440  052111  035110
011756  022440  031517  047045
011764     000
011765     045  022516  052101  FMT19:  .ASCIZ  /%N%ATEST %D2%A NOT RUN%N/
011772  051505  020124  042045
012000  022462  020101  047516
012006  020124  052522  022516
012014  000116
012016  047045  040445  046120  FMT24:  .ASCIZ  /%N%APLEASE INSURE M8207 RUN SWITCH (E28 SW7  IS ON%N/
```

```
 4586  012024  040505  042523  044440
 4587  012032  051516  051125  020105
 4588  012040  034115  030062  020067
 4589  012046  052522  020116  053523
 4590  012054  052111  044103  024040
 4591  012062  031105  020070  053523
 4592  012070  024467  044440  020123
 4593  012076  047117  047045     000
 4594
 4595
 4596
 4597  012103     103  051123  040440  EM1:    .ASCIZ  /CSR ADDRESS TIME-OUT (SEL0)/
 4598  012110  042104  042522  051523
 4599  012116  052040  046511  026505
 4600  012124  052517  020124  051450
 4601  012132  046105  024460     000
 4602  012137     122  043505  047040  EM2:    .ASCIZ  /REG NOT INITIALIZED BY MST CLR/
 4603  012144  052117  044440  044516
 4604  012152  044524  046101  055111
 4605  012160  042105  041040  020131
 4606  012166  051515  020124  046103
 4607  012174  000122
 4608  012176  042522  020107  044515  EM3:    .ASCIZ  /REG MISCOMPARE/
 4609  012204  041523  046517  040520
 4610  012212  042522     000
 4611  012215     122  043505  047040  EM4:    .ASCIZ  /REG NOT INITIALIZED BY UNIBUS RESET (INIT)/
 4612  012222  052117  044440  044516
 4613  012230  044524  046101  055111
 4614  012236  042105  041040  020131
 4615  012244  047125  041111  051525
 4616  012252  051040  051505  052105
 4617  012260  024040  047111  052111
 4618  012266  000051
 4619  012270  040515  047111  020124  EM5:    .ASCIZ  /MAINT CLK BIT STUCK AT 0/
 4620  012276  046103  020113  044502
 4621  012304  020124  052123  041525
 4622  012312  020113  052101  030040
 4623  012320     000
 4624  012321     115  044501  052116  EM6:    .ASCIZ  /MAINT CLK BIT STUCK AT 1/
 4625  012326  041440  045514  041040
 4626  012334  052111  051440  052524
 4627  012342  045503  040440  020124
 4628  012350  000061
 4629  012352  051117  054504  047040  EM7:    .ASCIZ  /ORDY NOT SET/
 4630  012360  052117  051440  052105
 4631  012366     000
 4632  012367     117  042122  020131  EM8:    .ASCIZ  /ORDY NOT CLEARED/
 4633  012374  047516  020124  046103
 4634  012402  040505  042522  000104
 4635  012410  041517  051117  047040  EM9:    .ASCIZ  /OCOR NOT SET/
 4636  012416  052117  051440  052105
 4637  012424     000
 4638  012425     117  047503  020122  EM10:   .ASCIZ  /OCOR NOT CLEARED/
 4639  012432  047516  020124  046103
 4640  012440  040505  042522  000104
 4641  012446  040517  052103  047040  EM11:   .ASCIZ  /OACT NOT SET/
```

```
4642  012454  052117  051440  052105
4643  012462     000
4644  012463     117  041501  020124  EM12:   .ASCIZ  /OACT NOT CLEARED/
4645  012470  047516  020124  046103
4646  012476  040505  042522  000104
4647  012504  047125  051122  047040  EM13:   .ASCIZ  /UNRR NOT CLEARED BY SOM/
4648  012512  052117  041440  042514
4649  012520  051101  042105  041040
4650  012526  020131  047523  000115
4651  012534  047125  051122  047040  EM14:   .ASCIZ  /UNRR NOT SET/
4652  012542  052117  051440  052105
4653  012550     000
4654  012551     125  051116  020122  EM15:   .ASCIZ  /UNRR NOT CLEARED BY OC/
4655  012556  047516  020124  046103
4656  012564  040505  042522  020104
4657  012572  054502  047440  000103
4658  012600  047125  051122  047040  EM16:   .ASCIZ  /UNRR NOT CLEARED/
4659  012606  052117  041440  042514
4660  012614  051101  042105     000
4661  012621     111  042122  020131  EM17:   .ASCIZ  /IRDY NOT SET/
4662  012626  047516  020124  042523
4663  012634  000124
4664  012636  051111  054504  047040  EM18:   .ASCIZ  /IRDY NOT CLEARED/
4665  012644  052117  041440  042514
4666  012652  051101  042105     000
4667  012657     111  044503  020122  EM19:   .ASCIZ  /ICIR NOT SET/
4668  012664  047516  020124  042523
4669  012672  000124
4670  012674  041511  051111  047040  EM20:   .ASCIZ  /ICIR NOT CLEARED/
4671  012702  052117  041440  042514
4672  012710  051101  042105     000
4673  012715     111  041501  020124  EM21:   .ASCIZ  /IACT NOT SET/
4674  012722  047516  020124  042523
4675  012730  000124
4676  012732  040511  052103  047040  EM22:   .ASCIZ  /IACT NOT CLEARED/
4677  012740  052117  041440  042514
4678  012746  051101  042105     000
4679  012753     104  051523  020111  EM23:   .ASCIZ  /DSSI NOT CLEARED/
4680  012760  047516  020124  046103
4681  012766  040505  042522  000104
4682  012774  051504  044523  047040  EM24:   .ASCIZ  /DSSI NOT SET/
4683  013002  052117  051440  052105
4684  013010     000
4685  013011     104  051523  020111  EM25:   .ASCIZ  /DSSI NOT CLEARED BY MST CLR/
4686  013016  047516  020124  046103
4687  013024  040505  042522  020104
4688  013032  054502  046440  052123
4689  013040  041440  051114     000
4690  013045     111  041516  051117  EM26:   .ASCIZ  /INCORRECT DATA CHAR RCV'D/
4691  013052  042522  052103  042040
4692  013060  052101  020101  044103
4693  013066  051101  051040  053103
4694  013074  042047     000
4695  013077     111  041516  051117  EM27:   .ASCIZ  /INCORRECT CRC BYTE RCV'D/
4696  013104  042522  052103  041440
4697  013112  041522  041040  052131
```

```
 4698  013120  020105  041522  023526
 4699  013126  000104
 4700  013130  051522  046517  047040  EM28:   .ASCIZ  /RSOM NOT CLEARED/
 4701  013136  052117  041440  042514
 4702  013144  051101  042105     000
 4703  013151     122  047523  020115  EM29:   .ASCIZ  /RSOM NOT SET/
 4704  013156  047516  020124  042523
 4705  013164  000124
 4706  013166  042522  046517  047040  EM30:   .ASCIZ  /REOM NOT CLEARED/
 4707  013174  052117  041440  042514
 4708  013202  051101  042105     000
 4709  013207     122  047505  020115  EM31:   .ASCIZ  /REOM NOT SET/
 4710  013214  047516  020124  042523
 4711  013222  000124
 4712  013224  054124  040504  040524  EM32:   .ASCIZ  /TXDATA BIT NOT CLEARED/
 4713  013232  041040  052111  047040
 4714  013240  052117  041440  042514
 4715  013246  051101  042105     000
 4716  013253     124  042130  052101  EM33:   .ASCIZ  /TXDATA BIT NOT SET/
 4717  013260  020101  044502  020124
 4718  013266  047516  020124  042523
 4719  013274  000124
 4720  013276  041522  023526  020104  EM34:   .ASCIZ  /RCV'D DATA MISCOMPARE/
 4721  013304  040504  040524  046440
 4722  013312  051511  047503  050115
 4723  013320  051101  000105
 4724  013324  041502  020103  047516  EM35:   .ASCIZ  /BCC NOT CLEARED/
 4725  013332  020124  046103  040505
 4726  013340  042522  000104
 4727  013344  041502  020103  047516  EM36:   .ASCIZ  /BCC NOT SET/
 4728  013352  020124  042523  000124
 4729  013360  041105  045514  047040  EM37:   .ASCIZ  /EBLK NOT CLEARED/
 4730  013366  052117  041440  042514
 4731  013374  051101  042105     000
 4732  013401     105  046102  020113  EM38:   .ASCIZ  /EBLK NOT SET/
 4733  013406  047516  020124  042523
 4734  013414  000124
 4735  013416  040522  020102  047516  FM39:   .ASCIZ  /RAB NOT CLEARED/
 4736  013424  020124  046103  040505
 4737  013432  042522  000104
 4738  013436  040522  020102  047516  EM40:   .ASCIZ  /RAB NOT SET/
 4739  013444  020124  042523  000124
 4740  013452  053117  051122  047040  EM41:   .ASCIZ  /OVRR NOT CLEARED/
 4741  013460  052117  041440  042514
 4742  013466  051101  042105     000
 4743  013473     117  051126  020122  EM42:   .ASCIZ  /OVRR NOT SET/
 4744  013500  047516  020124  042523
 4745  013506  000124
 4746  013510  053523  050040  041501  EM43:   .ASCIZ  /SW PACK #1 INCORRECT/
 4747  013516  020113  030443  044440
 4748  013524  041516  051117  042522
 4749  013532  052103     000
 4750  013535     123  020127  040520  EM44:   .ASCIZ  /SW PACK #2 INCORRECT/
 4751  013542  045503  021440  020062
 4752  013550  047111  047503  051122
 4753  013556  041505  000124
```

```
 4754  013562  053523  050040  041501  EM45:   .ASCIZ  /SW PACK #3 INCORRECT/
 4755  013570  020113  031443  044440
 4756  013576  041516  051117  042522
 4757  013604  052103     000
 4758  013607     122  053103  051440  EM46:   .ASCIZ  /RCV SILO NOT CLEARED BY IC/
 4759  013614  046111  020117  047516
 4760  013622  020124  046103  040505
 4761  013630  042522  020104  054502
 4762  013636  044440  000103
 4763  013642  051501  042523  041115  EM47:   .ASCIZ  /ASSEMB BIT COUNT INCORRECT/
 4764  013650  041040  052111  041440
 4765  013656  052517  052116  044440
 4766  013664  041516  051117  042522
 4767  013672  052103     000
 4768  013675     117  042104  053040  EM48:   .ASCIZ  /ODD VRC PARITY BIT NOT SET/
 4769  013702  041522  050040  051101
 4770  013710  052111  020131  044502
 4771  013716  020124  047516  020124
 4772  013724  042523  000124
 4773  013730  042117  020104  051126  EM49:   .ASCIZ  /ODD VRC PARITY BIT NOT CLEARED/
 4774  013736  020103  040520  044522
 4775  013744  054524  041040  052111
 4776  013752  047040  052117  041440
 4777  013760  042514  051101  042105
 4778  013766     000
 4779  013767     105  042526  020116  EM50:   .ASCIZ  /EVEN VRC PARITY BIT NOT SET/
 4780  013774  051126  020103  040520
 4781  014002  044522  054524  041040
 4782  014010  052111  047040  052117
 4783  014016  051440  052105     000
 4784  014023     105  042526  020116  EM51:   .ASCIZ  /EVEN VRC PARITY BIT NOT CLEARED/
 4785  014030  051126  020103  040520
 4786  014036  044522  054524  041040
 4787  014044  052111  047040  052117
 4788  014052  041440  042514  051101
 4789  014060  042105     000     .
 4790  014063     122  040505  054504  EM52:   .ASCIZ  /READY NOT SET AFTER AX REG WRITE/
 4791  014070  047040  052117  051440
 4792  014076  052105  040440  052106
 4793  014104  051105  040440  020130
 4794  014112  042522  020107  051127
 4795  014120  052111  000105
 4796  014124  042522  042101  020131  EM53:   .ASCIZ  /READY NOT SET AFTER AX REG READ/
 4797  014132  047516  020124  042523
 4798  014140  020124  043101  042524
 4799  014146  020122  054101  051040
 4800  014154  043505  051040  040505
 4801  014162  000104
 4802  014164  054124  052440  042116  EM54:   .ASCIZ  /TX UNDERRUN ERROR/
 4803  014172  051105  052522  020116
 4804  014200  051105  047522  000122
 4805  014206  052122  020123  047516  EM60:   .ASCIZ  /RTS NOT SET/
 4806  014214  020124  042523  000124
 4807  014222  052122  020123  047516  EM65:   .ASCIZ  /RTS NOT CLEARED/
 4808  014230  020124  046103  040505
 4809  014236  042522  000104
```

```
4810
4811
4812
4813  014242  047111  052502  027523  DH1:    .ASCIZ  &INBUS/OUTBUS REG &
4814  014250  052517  041124  051525
4815  014256  051040  043505  000040
4816  014264  044514  042516  052440  DH2:    .ASCIZ  /LINE UNIT INBUS REGS :/
4817  014272  044516  020124  047111
4818  014300  052502  020123  042522
4819  014306  051507  035040     000
4820  014313     122  043505  030061  DH3:    .ASCIZ  /REG10   REG11   REG12   REG13/
4821  014320  020040  051040  043505
4822  014326  030461  020040  051040
4823  014334  043505  031061  020040
4824  014342  051040  043505  031461
4825  014350     000
4826  014351     040  020040  051040  DH4:    .ASCIZ  /    REG14   REG15   REG16   REG17/
4827  014356  043505  032061  020040
4828  014364  051040  043505  032461
4829  014372  020040  051040  043505
4830  014400  033061  020040  051040
4831  014406  043505  033461     000
4832  014413     061  000065          DH5:    .ASCIZ  /15/
4833  014416  033061     000          DH6:    .ASCIZ  /16/
4834  014421     114  047111  020105  DH7:    .ASCIZ  /LINE UNIT EXTENDED REGS :/
4835  014426  047125  052111  042440
4836  014434  052130  047105  042504
4837  014442  020104  042522  051507
4838  014450  035040     000
4839  014453     101  030130  030455  DH8:    .ASCIZ  /AX0-15  AX0-16  AX1-15  AX1-16/
4840  014460  020065  040440  030130
4841  014466  030455  020066  040440
4842  014474  030530  030455  020065
4843  014502  040440  030530  030455
4844  014510  000066
4845  014512  020040  020040  054101  DH9:    .ASCIZ  /    AX2-15  AX2-16  AX3-15  AX3-16/
4846  014520  026462  032461  020040
4847  014526  054101  026462  033061
4848  014534  020040  054101  026463
4849  014542  032461  020040  054101
4850  014550  026463  033061     000
4851
4852          014556                  .EVEN
4853
4854
4855
4856
4857
4858  014556                          BGNMSG  ERR1
4859  014556                                                          ERR1::
4860  014556                          PRINTB  #FM11,#ADDRES,MPCSR
4861  014556  013746  002424                                                  MOV     MPCSR,-(SP)
4862  014562  012746  035522                                                  MOV     #ADDRES,-(SP)
4863  014566  012746  011432                                                  MOV     #FMT1,-(SP)
4864  014572  012746  000003                                                  MOV     #3,-(SP)
4865  014576  010600                                                          MOV     SP,R0
```

```
 4866  014600  104414                                                    TRAP    C$PNTB
 4867  014602  062706  000010                                            ADD     #10,SP
 4868  014606                      ENDMSG
 4869  014606                                                          L10002:
 4870  014606  104423                                                   TRAP    C$MSG
 4871
 4872
 4873
 4874  014610                      BGNMSG  ERR2
 4875  014610                                                          ERR2::
 4876  014610                      PRINTB  #FMT1,#ADDRES,MPCSR
 4877  014610  013746  002424                                           MOV     MPCSR,-(SP)
 4878  014614  012746  035522                                           MOV     #ADDRES,-(SP)
 4879  014620  012746  011432                                           MOV     #FMT1,-(SP)
 4880  014624  012746  000003                                           MOV     #3,-(SP)
 4881  014630  010600                                                   MOV     SP,R0
 4882  014632  104414                                                   TRAP    C$PNTB
 4883  014634  062706  000010                                           ADD     #10,SP
 4884  014640                      PRINTB  #FMT2
 4885  014640  012746  011442                                           MOV     #FMT2,-(SP)
 4886  014644  012746  000001                                           MOV     #1,-(SP)
 4887  014650  010600                                                   MOV     SP,R0
 4888  014652  104414                                                   TRAP    C$PNTB
 4889  014654  062706  000004                                           ADD     #4,SP
 4890  014660                      PRINTB  #FMT7,#DH1,REGNUM
 4891  014660  013746  002356                                           MOV     REGNUM,-(SP)
 4892  014664  012746  014242                                           MOV     #DH1,-(SP)
 4893  014670  012746  011624                                           MOV     #FMT7,-(SP)
 4894  014674  012746  000003                                           MOV     #3,-(SP)
 4895  014700  010600                                                   MOV     SP,R0
 4896  014702  104414                                                   TRAP    C$PNTB
 4897  014704  062706  000010                                           ADD     #10,SP
 4898  014710                      PRINTB  #FMT3,GOODAT,BADDAT
 4899  014710  013746  002364                                           MOV     BADDAT,-(SP)
 4900  014714  013746  002362                                           MOV     GOODAT,-(SP)
 4901  014720  012746  011464                                           MOV     #FMT3,-(SP)
 4902  014724  012746  000003                                           MOV     #3,-(SP)
 4903  014730  010600                                                   MOV     SP,R0
 4904  014732  104414                                                   TRAP    C$PNTB
 4905  014734  062706  000010                                           ADD     #10,SP
 4906  014740                      PRINTX  #FMT4,#DH2,#DH3
 4907  014740  012746  014313                                           MOV     #DH3,-(SP)
 4908  014744  012746  014264                                           MOV     #DH2,-(SP)
 4909  014750  012746  011526                                           MOV     #FMT4,-(SP)
 4910  014754  012746  000003                                           MOV     #3,-(SP)
 4911  014760  010600                                                   MOV     SP,R0
 4912  014762  104415                                                   TRAP    C$PNTX
 4913  014764  062706  000010                                           ADD     #10,SP
 4914  014770                      PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
 4915  014770  013746  002266                                           MOV     LUR13,-(SP)
 4916  014774  013746  002264                                           MOV     LUR12,-(SP)
 4917  015000  013746  002262                                           MOV     LUR11,-(SP)
 4918  015004  013746  002260                                           MOV     LUR10,-(SP)
 4919  015010  012746  011541                                           MOV     #FMT5,-(SP)
 4920  015014  012746  000005                                           MOV     #5,-(SP)
 4921  015020  010600                                                   MOV     SP,R0
```

```
4922  015022  104415                                                    TRAP    C$PNTX
4923  015024  062706  000014                                            ADD     #14,SP
4924  015030                        PRINTX  #FMT9,#DH4
4925  015030  012746  014351                                            MOV     #DH4,-(SP)
4926  015034  012746  011670                                            MOV     #FMT9,-(SP)
4927  015040  012746  000002                                            MOV     #2,-(SP)
4928  015044  010600                                                    MOV     SP,R0
4929  015046  104415                                                    TRAP    C$PNTX
4930  015050  062706  000006                                            ADD     #6,SP
4931  015054                        PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
4932  015054  013746  002276                                            MOV     LUR17,-(SP)
4933  015060  013746  002274                                            MOV     LUR16,-(SP)
4934  015064  013746  002272                                            MOV     LUR15,-(SP)
4935  015070  013746  002270                                            MOV     LUR14,-(SP)
4936  015074  012746  011571                                            MOV     #FMT6,-(SP)
4937  015100  012746  000005                                            MOV     #5,-(SP)
4938  015104  010600                                                    MOV     SP,R0
4939  015106  104415                                                    TRAP    C$PNTX
4940  015110  062706  000014                                            ADD     #14,SP
4941  015114                        ENDMSG
4942  015114                                                  L10003:
4943  015114  104423                                                    TRAP    C$MSG
4944
4945
4946
4947
4948
4949  015116                        BGNMSG  ERR3
4950  015116                                                  ERR3::
4951  015116                        PRINTB  #FMT1,#ADDRES,MPCSR
4952  015116  013746  002424                                            MOV     MPCSR,-(SP)
4953  015122  012746  035522                                            MOV     #ADDRES,-(SP)
4954  015126  012746  011432                                            MOV     #FMT1,-(SP)
4955  015132  012746  000003                                            MOV     #3,-(SP)
4956  015136  010600                                                    MOV     SP,R0
4957  015140  104414                                                    TRAP    C$PNTB
4958  015142  062706  000010                                            ADD     #10,SP
4959  015146                        PRINTB  #FMT2
4960  015146  012746  011442                                            MOV     #FMT2,-(SP)
4961  015152  012746  000001                                            MOV     #1,-(SP)
4962  015156  010600                                                    MOV     SP,R0
4963  015160  104414                                                    TRAP    C$PNTB
4964  015162  062706  000004                                            ADD     #4,SP
4965  015166                        PRINTB  #FMT8,TMP1,TMP0
4966  015166  013746  002506                                            MOV     TMP0,-(SP)
4967  015172  013746  002510                                            MOV     TMP1,-(SP)
4968  015176  012746  011634                                            MOV     #FMT8,-(SP)
4969  015202  012746  000003                                            MOV     #3,-(SP)
4970  015206  010600                                                    MOV     SP,R0
4971  015210  104414                                                    TRAP    C$PNTB
4972  015212  062706  000010                                            ADD     #10,SP
4973  015216                        PRINTB  #FMT3,GOODAT,BADDAT
4974  015216  013746  002364                                            MOV     BADDAT,-(SP)
4975  015222  013746  002362                                            MOV     GOODAT,-(SP)
4976  015226  012746  011464                                            MOV     #FMT3,-(SP)
4977  015232  012746  000003                                            MOV     #3,-(SP)
```

```
 4978  015236  010600                                                    MOV     SP,R0
 4979  015240  104414                                                    TRAP    C$PNTB
 4980  015242  062706  000010                                            ADD     #10,SP
 4981  015246                        PRINTX  #FMT4,#DH2,#DH3
 4982  015246  012746  014313                                            MOV     #DH3,-(SP)
 4983  015252  012746  014264                                            MOV     #DH2,-(SP)
 4984  015256  012746  011526                                            MOV     #FMT4,-(SP)
 4985  015262  012746  000003                                            MOV     #3,-(SP)
 4986  015266  010600                                                    MOV     SP,R0
 4987  015270  104415                                                    TRAP    C$PNTX
 4988  015272  062706  000010                                            ADD     #10,SP
 4989  015276                        PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
 4990  015276  013746  002266                                            MOV     LUR13,-(SP)
 4991  015302  013746  002264                                            MOV     LUR12,-(SP)
 4992  015306  013746  002262                                            MOV     LUR11,-(SP)
 4993  015312  013746  002260                                            MOV     LUR10,-(SP)
 4994  015316  012746  011541                                            MOV     #FMT5,-(SP)
 4995  015322  012746  000005                                            MOV     #5,-(SP)
 4996  015326  010600                                                    MOV     SP,R0
 4997  015330  104415                                                    TRAP    C$PNTX
 4998  015332  062706  000014                                            ADD     #14,SP
 4999  015336                        PRINTX  #FMT9,#DH4
 5000  015336  012746  014351                                            MOV     #DH4,-(SP)
 5001  015342  012746  011670                                            MOV     #FMT9,-(SP)
 5002  015346  012746  000002                                            MOV     #2,-(SP)
 5003  015352  010600                                                    MOV     SP,R0
 5004  015354  104415                                                    TRAP    C$PNTX
 5005  015356  062706  000006                                            ADD     #6,SP
 5006  015362                        PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
 5007  015362  013746  002276                                            MOV     LUR17,-(SP)
 5008  015366  013746  002274                                            MOV     LUR16,-(SP)
 5009  015372  013746  002272                                            MOV     LUR15,-(SP)
 5010  015376  013746  002270                                            MOV     LUR14,-(SP)
 5011  015402  012746  011571                                            MOV     #FMT6,-(SP)
 5012  015406  012746  000005                                            MOV     #5,-(SP)
 5013  015412  010600                                                    MOV     SP,R0
 5014  015414  104415                                                    TRAP    C$PNTX
 5015  015416  062706  000014                                            ADD     #14,SP
 5016  015422                        PRINTX  #FMT4,#DH7,#DH8
 5017  015422  012746  014453                                            MOV     #DH8,-(SP)
 5018  015426  012746  014421                                            MOV     #DH7,-(SP)
 5019  015432  012746  011526                                            MOV     #FMT4,-(SP)
 5020  015436  012746  000003                                            MOV     #3,-(SP)
 5021  015442  010600                                                    MOV     SP,R0
 5022  015444  104415                                                    TRAP    C$PNTX
 5023  015446  062706  000010                                            ADD     #10,SP
 5024  015452                        PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
 5025  015452  013746  002306                                            MOV     AX1.16,-(SP)
 5026  015456  013746  002304                                            MOV     AX1.15,-(SP)
 5027  015462  013746  002302                                            MOV     AX0.16,-(SP)
 5028  015466  013746  002300                                            MOV     AX0.15,-(SP)
 5029  015472  012746  011541                                            MOV     #FMT5,-(SP)
 5030  015476  012746  000005                                            MOV     #5,-(SP)
 5031  015502  010600                                                    MOV     SP,R0
 5032  015504  104415                                                    TRAP    C$PNTX
 5033  015506  062706  000014                                            ADD     #14,SP
```

```
5034  015512
5035  015512  012746  014512          PRINTX  #FMT9,#DH9                      MOV     #DH9,-(SP)
5036  015516  012746  011670                                                  MOV     #FMT9,-(SP)
5037  015522  012746  000002                                                  MOV     #2,-(SP)
5038  015526  010600                                                          MOV     SP,R0
5039  015530  104415                                                          TRAP    C$PNTX
5040  015532  062706  000006                                                  ADD     #6,SP
5041  015536          PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
5042  015536  013746  002316                                                  MOV     AX3.16,-(SP)
5043  015542  013746  002314                                                  MOV     AX3.15,-(SP)
5044  015546  013746  002312                                                  MOV     AX2.16,-(SP)
5045  015552  013746  002310                                                  MOV     AX2.15,-(SP)
5046  015556  012746  011571                                                  MOV     #FMT6,-(SP)
5047  015562  012746  000005                                                  MOV     #5,-(SP)
5048  015566  010600                                                          MOV     SP,R0
5049  015570  104415                                                          TRAP    C$PNTX
5050  015572  062706  000014                                                  ADD     #14,SP
5051  015576          ENDMSG
5052  015576                                                          L10004:
5053  015576  104423                                                          TRAP    C$MSG
5054
5055
5056
5057
5058
5059  015600          BGNMSG  ERR4
5060  015600                                                          ERR4::
5061  015600          PRINTB  #FMT10,SUBRPC
5062  015600  013746  002330                                                  MOV     SUBRPC,-(SP)
5063  015604  012746  011675                                                  MOV     #FMT10,-(SP)
5064  015610  012746  000002                                                  MOV     #2,-(SP)
5065  015614  010600                                                          MOV     SP,R0
5066  015616  104414                                                          TRAP    C$PNTB
5067  015620  062706  000006                                                  ADD     #6,SP
5068  015624          PRINTB  #FMT1,#ADDRES,MPCSR
5069  015624  013746  002424                                                  MOV     MPCSR,-(SP)
5070  015630  012746  035522                                                  MOV     #ADDRES,-(SP)
5071  015634  012746  011432                                                  MOV     #FMT1,-(SP)
5072  015640  012746  000003                                                  MOV     #3,-(SP)
5073  015644  010600                                                          MOV     SP,R0
5074  015646  104414                                                          TRAP    C$PNTB
5075  015650  062706  000010                                                  ADD     #10,SP
5076  015654          PRINTB  #FMT2
5077  015654  012746  011442                                                  MOV     #FMT2,-(SP)
5078  015660  012746  000001                                                  MOV     #1,-(SP)
5079  015664  010600                                                          MOV     SP,R0
5080  015666  104414                                                          TRAP    C$PNTB
5081  015670  062706  000004                                                  ADD     #4,SP
5082  015674          PRINTB  #FMT7,#DH1,REGNUM
5083  015674  013746  002356                                                  MOV     REGNUM,-(SP)
5084  015700  012746  014242                                                  MOV     #DH1,-(SP)
5085  015704  012746  011624                                                  MOV     #FMT7,-(SP)
5086  015710  012746  000003                                                  MOV     #3,-(SP)
5087  015714  010600                                                          MOV     SP,R0
5088  015716  104414                                                          TRAP    C$PNTB
5089  015720  062706  000010                                                  ADD     #10,SP
```

```
5090  015724
5091  015724  012746  014313          PRINTX  #FMT4,#DH2,#DH3
5092  015730  012746  014264                  MOV     #DH3,-(SP)
5093  015734  012746  011526                  MOV     #DH2,-(SP)
5094  015740  012746  000003                  MOV     #FMT4,-(SP)
5095  015744  010600                          MOV     #3,-(SP)
5096  015746  104415                          MOV     SP,R0
5097  015750  062706  000010                  TRAP    C$PNTX
5098  015754                                  ADD     #10,SP
5099  015754  013746  002266          PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
5100  015760  013746  002264                  MOV     LUR13,-(SP)
5101  015764  013746  002262                  MOV     LUR12,-(SP)
5102  015770  013746  002260                  MOV     LUR11,-(SP)
5103  015774  012746  011541                  MOV     LUR10,-(SP)
5104  016000  012746  000005                  MOV     #FMT5,-(SP)
5105  016004  010600                          MOV     #5,-(SP)
5106  016006  104415                          MOV     SP,R0
5107  016010  062706  000014                  TRAP    C$PNTX
5108  016014                                  ADD     #14,SP
5109  016014  012746  014351          PRINTX  #FMT9,#DH4
5110  016020  012746  011670                  MOV     #DH4,-(SP)
5111  016024  012746  000002                  MOV     #FMT9,-(SP)
5112  016030  010600                          MOV     #2,-(SP)
5113  016032  104415                          MOV     SP,R0
5114  016034  062706  000006                  TRAP    C$PNTX
5115  016040                                  ADD     #6,SP
5116  016040  013746  002276          PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
5117  016044  013746  002274                  MOV     LUR17,-(SP)
5118  016050  013746  002272                  MOV     LUR16,-(SP)
5119  016054  013746  002270                  MOV     LUR15,-(SP)
5120  016060  012746  011571                  MOV     LUR14,-(SP)
5121  016064  012746  000005                  MOV     #FMT6,-(SP)
5122  016070  010600                          MOV     #5,-(SP)
5123  016072  104415                          MOV     SP,R0
5124  016074  062706  000014                  TRAP    C$PNTX
5125  016100                                  ADD     #14,SP
5126  016100  012746  014453          PRINTX  #FMT4,#DH7,#DH8
5127  016104  012746  014421                  MOV     #DH8,-(SP)
5128  016110  012746  011526                  MOV     #DH7,-(SP)
5129  016114  012746  000003                  MOV     #FMT4,-(SP)
5130  016120  010600                          MOV     #3,-(SP)
5131  016122  104415                          MOV     SP,R0
5132  016124  062706  000010                  TRAP    C$PNTX
5133  016130                                  ADD     #10,SP
5134  016130  013746  002306          PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
5135  016134  013746  002304                  MOV     AX1.16,-(SP)
5136  016140  013746  002302                  MOV     AX1.15,-(SP)
5137  016144  013746  002300                  MOV     AX0.16,-(SP)
5138  016150  012746  011541                  MOV     AX0.15,-(SP)
5139  016154  012746  000005                  MOV     #FMT5,-(SP)
5140  016160  010600                          MOV     #5,-(SP)
5141  016162  104415                          MOV     SP,R0
5142  016164  062706  000014                  TRAP    C$PNTX
5143  016170                                  ADD     #14,SP
5144  016170  012746  014512          PRINTX  #FMT9,#DH9
5145  016174  012746  011670                  MOV     #DH9,-(SP)
                                              MOV     #FMT9,-(SP)
```

```
5146  016200  012746  000002                                          MOV     #2,-(SP)
5147  016204  010600                                                  MOV     SP,R0
5148  016206  104415                                                  TRAP    C$PNTX
5149  016210  062706  000006                                          ADD     #6,SP
5150  016214                  PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
5151  016214  013746  002316                                          MOV     AX3.16,-(SP)
5152  016220  013746  002314                                          MOV     AX3.15,-(SP)
5153  016224  013746  002312                                          MOV     AX2.16,-(SP)
5154  016230  013746  002310                                          MOV     AX2.15,-(SP)
5155  016234  012746  011571                                          MOV     #FMT6,-(SP)
5156  016240  012746  000005                                          MOV     #5,-(SP)
5157  016244  010600                                                  MOV     SP,R0
5158  016246  104415                                                  TRAP    C$PNTX
5159  016250  062706  000014                                          ADD     #14,SP
5160  016254                  ENDMSG
5161  016254                                                  L10005:
5162  016254  104423                                                  TRAP    C$MSG
5163
5164
5165
5166
5167
5168  016256                  BGNMSG  ERR5
5169  016256                                                  ERR5::
5170  016256                  PRINTB  #FMT1,#ADDRES,MPCSR
5171  016256  013746  002424                                          MOV     MPCSR,-(SP)
5172  016262  012746  035522                                          MOV     #ADDRES,-(SP)
5173  016266  012746  011432                                          MOV     #FMT1,-(SP)
5174  016272  012746  000003                                          MOV     #3,-(SP)
5175  016276  010600                                                  MOV     SP,R0
5176  016300  104414                                                  TRAP    C$PNTB
5177  016302  062706  000010                                          ADD     #10,SP
5178  016306                  PRINTB  #FMT11,REGNUM,LOADAT
5179  016306  013746  002366                                          MOV     LOADAT,-(SP)
5180  016312  013746  002356                                          MOV     REGNUM,-(SP)
5181  016316  012746  011726                                          MOV     #FMT11,-(SP)
5182  016322  012746  000003                                          MOV     #3,-(SP)
5183  016326  010600                                                  MOV     SP,R0
5184  016330  104414                                                  TRAP    C$PNTB
5185  016332  062706  000010                                          ADD     #10,SP
5186  016336                  PRINTB  #FMT2
5187  016336  012746  011442                                          MOV     #FMT2,-(SP)
5188  016342  012746  000001                                          MOV     #1,-(SP)
5189  016346  010600                                                  MOV     SP,R0
5190  016350  104414                                                  TRAP    C$PNTB
5191  016352  062706  000004                                          ADD     #4,SP
5192  016356                  PRINTB  #FMT8,TMP1,TMP0
5193  016356  013746  002506                                          MOV     TMP0,-(SP)
5194  016362  013746  002510                                          MOV     TMP1,-(SP)
5195  016366  012746  011634                                          MOV     #FMT8,-(SP)
5196  016372  012746  000003                                          MOV     #3,-(SP)
5197  016376  010600                                                  MOV     SP,R0
5198  016400  104414                                                  TRAP    C$PNTB
5199  016402  062706  000010                                          ADD     #10,SP
5200  016406                  PRINTB  #FMT3,GOODAT,BADDAT
5201  016406  013746  002364                                          MOV     BADDAT,-(SP)
```

```
5202  016412  013746  002362                                    MOV     GOODAT,-(SP)
5203  016416  012746  011464                                    MOV     #FMT3,-(SP)
5204  016422  012746  000003                                    MOV     #3,-(SP)
5205  016426  010600                                            MOV     SP,R0
5206  016430  104414                                            TRAP    C$PNTB
5207  016432  062706  000010                                    ADD     #10,SP
5208  016436                  PRINTX  #FMT4,#DH2,#DH3
5209  016436  012746  014313                                    MOV     #DH3,-(SP)
5210  016442  012746  014264                                    MOV     #DH2,-(SP)
5211  016446  012746  011526                                    MOV     #FMT4,-(SP)
5212  016452  012746  000003                                    MOV     #3,-(SP)
5213  016456  010600                                            MOV     SP,R0
5214  016460  104415                                            TRAP    C$PNTX
5215  016462  062706  000010                                    ADD     #10,SP
5216  016466                  PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
5217  016466  013746  002266                                    MOV     LUR13,-(SP)
5218  016472  013746  002264                                    MOV     LUR12,-(SP)
5219  016476  013746  002262                                    MOV     LUR11,-(SP)
5220  016502  013746  002260                                    MOV     LUR10,-(SP)
5221  016506  012746  011541                                    MOV     #FMT5,-(SP)
5222  016512  012746  000005                                    MOV     #5,-(SP)
5223  016516  010600                                            MOV     SP,R0
5224  016520  104415                                            TRAP    C$PNTX
5225  016522  062706  000014                                    ADD     #14,SP
5226  016526                  PRINTX  #FMT9,#DH4
5227  016526  012746  014351                                    MOV     #DH4,-(SP)
5228  016532  012746  011670                                    MOV     #FMT9,-(SP)
5229  016536  012746  000002                                    MOV     #2,-(SP)
5230  016542  010600                                            MOV     SP,R0
5231  016544  104415                                            TRAP    C$PNTX
5232  016546  062706  000006                                    ADD     #6,SP
5233  016552                  PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
5234  016552  013746  002276                                    MOV     LUR17,-(SP)
5235  016556  013746  002274                                    MOV     LUR16,-(SP
5236  016562  013746  002272                                    MOV     LUR15,-(SP)
5237  016566  013746  002270                                    MOV     LUR14,-(SP)
5238  016572  012746  011571                                    MOV     #FMT6,-(SP)
5239  016576  012746  000005                                    MOV     #5,-(SP)
5240  016602  010600                                            MOV     SP,R0
5241  016604  104415                                            TRAP    C$PNTX
5242  016606  062706  000014                                    ADD     #14,SP
5243  016612                  PRINTX  #FMT4,#DH7,#DH8
5244  016612  012746  014453                                    MOV     #DH8,-(SP)
5245  016616  012746  014421                                    MOV     #DH7,-(SP)
5246  016622  012746  011526                                    MOV     #FMT4,-(SP)
5247  016626  012746  000003                                    MOV     #3,-(SP)
5248  016632  010600                                            MOV     SP,R0
5249  016634  104415                                            TRAP    C$PNTX
5250  016636  062706  000010                                    ADD     #10,SP
5251  016642                  PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
5252  016642  013746  002306                                    MOV     AX1.16,-(SP)
5253  016646  013746  002304                                    MOV     AX1.15,-(SP)
5254  016652  013746  002302                                    MOV     AX0.16,-(SP)
5255  016656  013746  002300                                    MOV     AX0.15,-(SP)
5256  016662  012746  011541                                    MOV     #FMT5,-(SP)
5257  016666  012746  000005                                    MOV     #5,-(SP)
```

```
5258  016672  010600
5259  016674  104415
5260  016676  062706  000014
5261  016702
5262  016702  012746  014512
5263  016706  012746  011670
5264  016712  012746  000002
5265  016716  010600
5266  016720  104415
5267  016722  062706  000006
5268  016726
5269  016726  013746  002316
5270  016732  013746  002314
5271  016736  013746  002312
5272  016742  013746  002310
5273  016746  012746  011571
5274  016752  012746  000005
5275  016756  010600
5276  016760  104415
5277  016762  062706  000014
5278  016766
5279  016766
5280  016766  104423
5281
5282
5283
5284
5285
5286  016770
5287  016770
5288  016770
5289  016770  013746  002330
5290  016774  012746  011675
5291  017000  012746  000002
5292  017004  010600
5293  017006  104414
5294  017010  062706  000006
5295  017014
5296  017014  013746  002424
5297  017020  012746  035522
5298  017024  012746  011432
5299  017030  012746  000003
5300  017034  010600
5301  017036  104414
5302  017040  062706  000010
5303  017044
5304  017044  012746  011442
5305  017050  012746  000001
5306  017054  010600
5307  017056  104414
5308  017060  062706  000004
5309  017064
5310  017064  013746  002506
5311  017070  013746  002510
5312  017074  012746  011634
5313  017100  012746  000003
```

```
        PRINTX  #FMT9,#DH9

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

        ENDMSG

        BGNMSG  ERR6

        PRINTB  #FMT10,SUBRPC

        PRINTB  #FMT1,#ADDRES,MPCSR

        PRINTB  #FMT2

        PRINTB  #FMT8,TMP1,TMP0
```

```
        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

L10006:
        TRAP    C$MSG
```

```
ERR6::
        MOV     SUBRPC,-(SP)
        MOV     #FMT10,-(SP)
        MOV     #2,-(SP)
        MOV     SP,R0
        TRAP    C$PNTB
        ADD     #6,SP

        MOV     MPCSR,-(SP)
        MOV     #ADDRES,-(SP)
        MOV     #FMT1,-(SP)
        MOV     #3,-(SP)
        MOV     SP,R0
        TRAP    C$PNTB
        ADD     #10,SP

        MOV     #FMT2,-(SP)
        MOV     #1,-(SP)
        MOV     SP,R0
        TRAP    C$PNTB
        ADD     #4,SP

        MOV     TMP0,-(SP)
        MOV     TMP1,-(SP)
        MOV     #FMT8,-(SP)
        MOV     #3,-(SP)
```

```
5314  017104  010600                                          MOV     SP,R0
5315  017106  104414                                          TRAP    C$PNTB
5316  017110  062706  000010                                  ADD     #10,SP
5317  017114                  PRINTX  #FMT4,#DH2,#DH3
5318  017114  012746  014313                                  MOV     #DH3,-(SP)
5319  017120  012746  014264                                  MOV     #DH2,-(SP)
5320  017124  012746  011526                                  MOV     #FMT4,-(SP)
5321  017130  012746  000003                                  MOV     #3,-(SP)
5322  017134  010600                                          MOV     SP,R0
5323  017136  104415                                          TRAP    C$PNTX
5324  017140  062706  000010                                  ADD     #10,SP
5325  017144                  PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
5326  017144  013746  002266                                  MOV     LUR13,-(SP)
5327  017150  013746  002264                                  MOV     LUR12,-(SP)
5328  017154  013746  002262                                  MOV     LUR11,-(SP)
5329  017160  013746  002260                                  MOV     LUR10,-(SP)
5330  017164  012746  011541                                  MOV     #FMT5,-(SP)
5331  017170  012746  000005                                  MOV     #5,-(SP)
5332  017174  010600                                          MOV     SP,R0
5333  017176  104415                                          TRAP    C$PNTX
5334  017200  062706  000014                                  ADD     #14,SP
5335  017204                  PRINTX  #FMT9,#DH4
5336  017204  012746  014351                                  MOV     #DH4,-(SP)
5337  017210  012746  011670                                  MOV     #FMT9,-(SP)
5338  017214  012746  000002                                  MOV     #2,-(SP)
5339  017220  010600                                          MOV     SP,R0
5340  017222  104415                                          TRAP    C$PNTX
5341  017224  062706  000006                                  ADD     #6,SP
5342  017230                  PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
5343  017230  013746  002276                                  MOV     LUR17,-(SP)
5344  017234  013746  002274                                  MOV     LUR16,-(SP)
5345  017240  013746  002272                                  MOV     LUR15,-(SP)
5346  017244  013746  002270                                  MOV     LUR14,-(SP)
5347  017250  012746  011571                                  MOV     #FMT6,-(SP`
5348  017254  012746  000005                                  MOV     #5,-(SP)
5349  017260  010600                                          MOV     SP,R0
5350  017262  104415                                          TRAP    C$PNTX
5351  017264  062706  000014                                  ADD     #14,SP
5352  017270                  PRINTX  #FMT4,#DH7,#DH8
5353  017270  012746  014453                                  MOV     #DH8,-(SP)
5354  017274  012746  014421                                  MOV     #DH7,-(SP)
5355  017300  012746  011526                                  MOV     #FMT4,-(SP)
5356  017304  012746  000003                                  MOV     #3,-(SP)
5357  017310  010600                                          MOV     SP,R0
5358  017312  104415                                          TRAP    C$PNTX
5359  017314  062706  000010                                  ADD     #10,SP
5360  017320                  PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
5361  017320  013746  002306                                  MOV     AX1.16,-(SP)
5362  017324  013746  002304                                  MOV     AX1.15,-(SP)
5363  017330  013746  002302                                  MOV     AX0.16,-(SP)
5364  017334  013746  002300                                  MOV     AX0.15,-(SP)
5365  017340  012746  011541                                  MOV     #FMT5,-(SP)
5366  017344  012746  000005                                  MOV     #5,-(SP)
5367  017350  010600                                          MOV     SP,R0
5368  017352  104415                                          TRAP    C$PNTX
5369  017354  062706  000014                                  ADD     #14,SP
```

```
5370  017360
5371  017360  012746  014512          PRINTX  #FMT9,#DH9                      MOV     #DH9,-(SP)
5372  017364  012746  011670                                                  MOV     #FMT9,-(SP)
5373  017370  012746  000002                                                  MOV     #2,-(SP)
5374  017374  010600                                                          MOV     SP,R0
5375  017376  104415                                                          TRAP    C$PNTX
5376  017400  062706  000006                                                  ADD     #6,SP
5377  017404                          PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
5378  017404  013746  002316                                                  MOV     AX3.16,-(SP)
5379  017410  013746  002314                                                  MOV     AX3.15,-(SP)
5380  017414  013746  002312                                                  MOV     AX2.16,-(SP)
5381  017420  013746  002310                                                  MOV     AX2.15,-(SP)
5382  017424  012746  011571                                                  MOV     #FMT6,-(SP)
5383  017430  012746  000005                                                  MOV     #5,-(SP)
5384  017434  010600                                                          MOV     SP,R0
5385  017436  104415                                                          TRAP    C$PNTX
5386  017440  062706  000014                                                  ADD     #14,SP
5387  017444                          ENDMSG
5388  017444                                                          L10007:
5389  017444  104423                                                          TRAP    C$MSG
5390
5391
5392
5393
5394
5395  017446                          BGNMSG  ERR7
5396  017446                                                          ERR7::
5397  017446                          PRINTB  #FMT1,#ADDRES,MPCSR
5398  017446  013746  002424                                                  MOV     MPCSR,-(SP)
5399  017452  012746  035522                                                  MOV     #ADDRES,-(SP)
5400  017456  012746  011432                                                  MOV     #FMT1,-(SP)
5401  017462  012746  000003                                                  MOV     #3,-(SP)
5402  017466  010600                                                          MOV     SP,R0
5403  017470  104414                                                          TRAP    C$PNTB
5404  017472  062706  000010                                                  ADD     #10,SP
5405  017476                          PRINTB  #FMT2
5406  017476  012746  011442                                                  MOV     #FMT2,-(SP)
5407  017502  012746  000001                                                  MOV     #1,-(SP)
5408  017506  010600                                                          MOV     SP,R0
5409  017510  104414                                                          TRAP    C$PNTB
5410  017512  062706  000004                                                  ADD     #4,SP
5411  017516                          PRINTB  #FMT7,#DH1,REGNUM
5412  017516  013746  002356                                                  MOV     REGNUM,-(SP)
5413  017522  012746  014242                                                  MOV     #DH1,-(SP)
5414  017526  012746  011624                                                  MOV     #FMT7,-(SP)
5415  017532  012746  000003                                                  MOV     #3,-(SP)
5416  017536  010600                                                          MOV     SP,R0
5417  017540  104414                                                          TRAP    C$PNTB
5418  017542  062706  000010                                                  ADD     #10,SP
5419  017546                          PRINTX  #FMT4,#DH2,#DH3
5420  017546  012746  014313                                                  MOV     #DH3,-(SP)
5421  017552  012746  014264                                                  MOV     #DH2,-(SP)
5422  017556  012746  011526                                                  MOV     #FMT4,-(SP)
5423  017562  012746  000003                                                  MOV     #3,-(SP)
5424  017566  010600                                                          MOV     SP,R0
5425  017570  104415                                                          TRAP    C$PNTX
```

```
5426  017572  062706  J0010          PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
5427  017576                                                                  ADD     #10,SP
5428  017576  013746  002266                                                  MOV     LUR13,-(SP)
5429  017602  013746  002264                                                  MOV     LUR12,-(SP)
5430  017606  013746  002262                                                  MOV     LUR11,-(SP)
5431  017612  013746  002260                                                  MOV     LUR10,-(SP)
5432  017616  012746  011541                                                  MOV     #FMT5,-(SP)
5433  017622  012746  000005                                                  MOV     #5,-(SP)
5434  017626  010600                                                          MOV     SP,R0
5435  017630  104415                                                          TRAP    C$PNTX
5436  017632  062706  000014                                                  ADD     #14,SP
5437  017636                         PRINTX  #FMT9,#DH4
5438  017636  012746  014351                                                  MOV     #DH4,-(SP)
5439  017642  012746  011670                                                  MOV     #FMT9,-(SP)
5440  017646  012746  000002                                                  MOV     #2,-(SP)
5441  017652  010600                                                          MOV     SP,R0
5442  017654  104415                                                          TRAP    C$PNTX
5443  017656  062706  000006                                                  ADD     #6,SP
5444  017662                         PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
5445  017662  013746  002276                                                  MOV     LUR17,-(SP)
5446  017666  013746  002274                                                  MOV     LUR16,-(SP)
5447  017672  013746  002272                                                  MOV     LUR15,-(SP)
5448  017676  013746  002270                                                  MOV     LUR14,-(SP)
5449  017702  012746  011571                                                  MOV     #FMT6,-(SP)
5450  017706  012746  000005                                                  MOV     #5,-(SP)
5451  017712  010600                                                          MOV     SP,R0
5452  017714  104415                                                          TRAP    C$PNTX
5453  017716  062706  000014                                                  ADD     #14,SP
5454  017722                         PRINTX  #FMT4,#DH7,#DH8
5455  017722  012746  014453                                                  MOV     #DH8,-(SP)
5456  017726  012746  014421                                                  MOV     #DH7,-(SP)
5457  017732  012746  011526                                                  MOV     #FMT4,-(SP)
5458  017736  012746  000003                                                  MOV     #3,-(SP)
5459  017742  010600                                                          MOV     SP,R0
5460  017744  104415                                                          TRAP    C$PNTX
5461  017746  062706  000010                                                  ADD     #10,SP
5462  017752                         PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
5463  017752  013746  002306                                                  MOV     AX1.16,-(SP)
5464  017756  013746  002304                                                  MOV     AX1.15,-(SP)
5465  017762  013746  002302                                                  MOV     AX0.16,-(SP)
5466  017766  013746  002300                                                  MOV     AX0.15,-(SP)
5467  017772  012746  011541                                                  MOV     #FMT5,-(SP)
5468  017776  012746  000005                                                  MOV     #5,-(SP)
5469  020002  010600                                                          MOV     SP,R0
5470  020004  104415                                                          TRAP    C$PNTX
5471  020006  062706  000014                                                  ADD     #14,SP
5472  020012                         PRINTX  #FMT9,#DH9
5473  020012  012746  014512                                                  MOV     #DH9,-(SP)
5474  020016  012746  011670                                                  MOV     #FMT9,-(SP)
5475  020022  012746  000002                                                  MOV     #2,-(SP)
5476  020026  010600                                                          MOV     SP,R0
5477  020030  104415                                                          TRAP    C$PNTX
5478  020032  062706  000006                                                  ADD     #6,SP
5479  020036                         PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
5480  020036  013746  002316                                                  MOV     AX3.16,-(SP)
5481  020042  013746  002314                                                  MOV     AX3.15,-(SP)
```

```
 5482  020046  013746  002312                                                    MOV     AX2.16,-(SP)
 5483  020052  013746  002310                                                    MOV     AX2.15,-(SP)
 5484  020056  012746  011571                                                    MOV     #FMT6,-(SP)
 5485  020062  012746  000005                                                    MOV     #5,-(SP)
 5486  020066  010600                                                           MOV     SP,R0
 5487  020070  104415                                                           TRAP    C$PNTX
 5488  020072  062706  000014                                                   ADD     #14,SP
 5489  020076                        ENDMSG
 5490  020076                                                          L10010:
 5491  020076  104423                                                           TRAP    C$MSG
 5492
 5493
 5494
 5495
 5496
 5497  020100                        BGNMSG  ERR8
 5498  020100                                                          ERR8::
 5499  020100                        PRINTB  #FMT10,SUBRPC
 5500  020100  013746  002330                                                    MOV     SUBRPC,-(SP)
 5501  020104  012746  011675                                                    MOV     #FMT10,-(SP)
 5502  020110  012746  000002                                                    MOV     #2,-(SP)
 5503  020114  010600                                                           MOV     SP,R0
 5504  020116  104414                                                           TRAP    C$PNTB
 5505  020120  062706  000006                                                   ADD     #6,SP
 5506  020124                        PRINTB  #FMT1,#ADDRES,MPCSR
 5507  020124  013746  002424                                                    MOV     MPCSR,-(SP)
 5508  020130  012746  035522                                                    MOV     #ADDRES,-(SP)
 5509  020134  012746  011432                                                    MOV     #FMT1,-(SP)
 5510  020140  012746  000003                                                    MOV     #3,-(SP)
 5511  020144  010600                                                           MOV     SP,R0
 5512  020146  104414                                                           TRAP    C$PNTB
 5513  020150  062706  000010                                                   ADD     #10,SP
 5514  020154                        PRINTB  #FMT2
 5515  020154  012746  011442                                                    MOV     #FMT2,-(SP
 5516  020160  012746  000001                                                    MOV     #1,-(SP)
 5517  020164  010600                                                           MOV     SP,R0
 5518  020166  104414                                                           TRAP    C$PNTB
 5519  020170  062706  000004                                                   ADD     #4,SP
 5520  020174                        PRINTB  #FMT7,#DH1,REGNUM
 5521  020174  013746  002356                                                    MOV     REGNUM,-(SP)
 5522  020200  012746  014242                                                    MOV     #DH1,-(SP)
 5523  020204  012746  011624                                                    MOV     #FMT7,-(SP)
 5524  020210  012746  000003                                                    MOV     #3,-(SP)
 5525  020214  010600                                                           MOV     SP,R0
 5526  020216  104414                                                           TRAP    C$PNTB
 5527  020220  062706  000010                                                   ADD     #10,SP
 5528  020224                        PRINTB  #FMT3,GOODAT,BADDAT
 5529  020224  013746  002364                                                    MOV     BADDAT,-(SP)
 5530  020230  013746  002362                                                    MOV     GOODAT,-(SP)
 5531  020234  012746  011464                                                    MOV     #FMT3,-(SP)
 5532  020240  012746  000003                                                    MOV     #3,-(SP)
 5533  020244  010600                                                           MOV     SP,R0
 5534  020246  104414                                                           TRAP    C$PNTB
 5535  020250  062706  000010                                                   ADD     #10,SP
 5536  020254                        PRINTX  #FMT4,#DH2,#DH3
 5537  020254  012746  014313                                                    MOV     #DH3,-(SP)
```

```
5538  020260  012746  014264                                    MOV     #DH2,-(SP)
5539  020264  012746  011526                                    MOV     #FMT4,-(SP)
5540  020270  012746  000003                                    MOV     #3,-(SP)
5541  020274  010600                                           MOV     SP,R0
5542  020276  104415                                           TRAP    C$PNTX
5543  020300  062706  000010                                   ADD     #10,SP
5544  020304                  PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
5545  020304  013746  002266                                    MOV     LUR13,-(SP)
5546  020310  013746  002264                                    MOV     LUR12,-(SP)
5547  020314  013746  002262                                    MOV     LUR11,-(SP)
5548  020320  013746  002260                                    MOV     LUR10,-(SP)
5549  020324  012746  011541                                    MOV     #FMT5,-(SP)
5550  020330  012746  000005                                    MOV     #5,-(SP)
5551  020334  010600                                           MOV     SP,R0
5552  020336  104415                                           TRAP    C$PNTX
5553  020340  062706  000014                                   ADD     #14,SP
5554  020344                  PRINTX  #FMT9,#DH4
5555  020344  012746  014351                                    MOV     #DH4,-(SP)
5556  020350  012746  011670                                    MOV     #FMT9,-(SP)
5557  020354  012746  000002                                    MOV     #2,-(SP)
5558  020360  010600                                           MOV     SP,R0
5559  020362  104415                                           TRAP    C$PNTX
5560  020364  062706  000006                                   ADD     #6,SP
5561  020370                  PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
5562  020370  013746  002276                                    MOV     LUR17,-(SP)
5563  020374  013746  002274                                    MOV     LUR16,-(SP)
5564  020400  013746  002272                                    MOV     LUR15,-(SP)
5565  020404  013746  002270                                    MOV     LUR14,-(SP)
5566  020410  012746  011571                                    MOV     #FMT6,-(SP)
5567  020414  012746  000005                                    MOV     #5,-(SP)
5568  020420  010600                                           MOV     SP,R0
5569  020422  104415                                           TRAP    C$PNTX
5570  020424  062706  000014                                   ADD     #14,SP
5571  020430                  PRINTX  #FMT4,#DH7,#DH8
5572  020430  012746  014453                                    MOV     #DH8,-(SP)
5573  020434  012746  014421                                    MOV     #DH7,-(SP)
5574  020440  012746  011526                                    MOV     #FMT4,-(SP)
5575  020444  012746  000003                                    MOV     #3,-(SP)
5576  020450  010600                                           MOV     SP,R0
5577  020452  104415                                           TRAP    C$PNTX
5578  020454  062706  000010                                   ADD     #10,SP
5579  020460                  PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
5580  020460  013746  002306                                    MOV     AX1.16,-(SP)
5581  020464  013746  002304                                    MOV     AX1.15,-(SP)
5582  020470  013746  002302                                    MOV     AX0.16,-(SP)
5583  020474  013746  002300                                    MOV     AX0.15,-(SP)
5584  020500  012746  011541                                    MOV     #FMT5,-(SP)
5585  020504  012746  000005                                    MOV     #5,-(SP)
5586  020510  010600                                           MOV     SP,R0
5587  020512  104415                                           TRAP    C$PNTX
5588  020514  062706  000014                                   ADD     #14,SP
5589  020520                  PRINTX  #FMT9,#DH9
5590  020520  012746  014512                                    MOV     #DH9,-(SP)
5591  020524  012746  011670                                    MOV     #FMT9,-(SP)
5592  020530  012746  000002                                    MOV     #2,-(SP)
5593  020534  010600                                           MOV     SP,R0
```

```
5594  020536  104415                                                    TRAP    C$PNTX
5595  020540  062706  000006                                            ADD     #6,SP
5596  020544                    PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
5597  020544  013746  002316                                            MOV     AX3.16,-(SP)
5598  020550  013746  002314                                            MOV     AX3.15,-(SP)
5599  020554  013746  002312                                            MOV     AX2.16,-(SP)
5600  020560  013746  002310                                            MOV     AX2.15,-(SP)
5601  020564  012746  011571                                            MOV     #FMT6,-(SP)
5602  020570  012746  000005                                            MOV     #5,-(SP)
5603  020574  010600                                                    MOV     SP,R0
5604  020576  104415                                                    TRAP    C$PNTX
5605  020600  062706  000014                                            ADD     #14,SP
5606  020604                    ENDMSG
5607  020604                                                  L10011:
5608  020604  104423                                                    TRAP    C$MSG
5609
5610
5611
5612
5613
5614  020606                    BGNMSG  ERR9
5615  020606                                                  ERR9::
5616  020606                    PRINTB  #FMT1,#ADDRES,MPCSR
5617  020606  013746  002424                                            MOV     MPCSR,-(SP)
5618  020612  012746  035522                                            MOV     #ADDRES,-(SP)
5619  020616  012746  011432                                            MOV     #FMT1,-(SP)
5620  020622  012746  000003                                            MOV     #3,-(SP)
5621  020626  010600                                                    MOV     SP,R0
5622  020630  104414                                                    TRAP    C$PNTB
5623  020632  062706  000010                                            ADD     #10,SP
5624  020636                    PRINTB  #FMT2
5625  020636  012746  011442                                            MOV     #FMT2,-(SP)
5626  020642  012746  000001                                            MOV     #1,-(SP)
5627  020646  010600                                                    MOV     SP,R0
5628  020650  104414                                                    TRAP    C$PNTB
5629  020652  062706  000004                                            ADD     #4,SP
5630  020656                    PRINTB  #FMT7,#DH1,REGNUM
5631  020656  013746  002356                                            MOV     REGNUM,-(SP)
5632  020662  012746  014242                                            MOV     #DH1,-(SP)
5633  020666  012746  011624                                            MOV     #FMT7,-(SP)
5634  020672  012746  000003                                            MOV     #3,-(SP)
5635  020676  010600                                                    MOV     SP,R0
5636  020700  104414                                                    TRAP    C$PNTB
5637  020702  062706  000010                                            ADD     #10,SP
5638  020706                    PRINTX  #FMT4,#DH2,#DH3
5639  020706  012746  014313                                            MOV     #DH3,-(SP)
5640  020712  012746  014264                                            MOV     #DH2,-(SP)
5641  020716  012746  011526                                            MOV     #FMT4,-(SP)
5642  020722  012746  000003                                            MOV     #3,-(SP)
5643  020726  010600                                                    MOV     SP,R0
5644  020730  104415                                                    TRAP    C$PNTX
5645  020732  062706  000010                                            ADD     #10,SP
5646  020736                    PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
5647  020736  013746  002266                                            MOV     LUR13,-(SP)
5648  020742  013746  002264                                            MOV     LUR12,-(SP)
5649  020746  013746  002262                                            MOV     LUR11,-(SP)
```

```
5650  020752  013746  002260                                        MOV     LUR10,-(SP)
5651  020756  012746  011541                                        MOV     #FMT5,-(SP)
5652  020762  012746  000005                                        MOV     #5,-(SP)
5653  020766  010600                                               MOV     SP,R0
5654  020770  104415                                               TRAP    C$PN1X
5655  020772  062706  000014                                       ADD     #14,SP
5656  020776                      PRINTX  #FMT9,#DH4
5657  020776  012746  014351                                       MOV     #DH4,-(SP)
5658  021002  012746  011670                                       MOV     #FMT9,-(SP)
5659  021006  012746  000002                                       MOV     #2,-(SP)
5660  021012  010600                                               MOV     SP,R0
5661  021014  104415                                               TRAP    C$PNTX
5662  021016  062706  000006                                       ADD     #6,SP
5663  021022                      PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
5664  021022  013746  002276                                       MOV     LUR17,-(SP)
5665  021026  013746  002274                                       MOV     LUR16,-(SP)
5666  021032  013746  002272                                       MOV     LUR15,-(SP)
5667  021036  013746  002270                                       MOV     LUR14,-(SP)
5668  021042  012746  011571                                       MOV     #FMT6,-(SP)
5669  021046  012746  000005                                       MOV     #5,-(SP)
5670  021052  010600                                               MOV     SP,R0
5671  021054  104415                                               TRAP    C$PNTX
5672  021056  062706  000014                                       ADD     #14,SP
5673  021062                      ENDMSG
5674  021062                                            L10012:
5675  021062  104423                                               TRAP    C$MSG
5676
5677
5678
5679
5680
```

```
CZDMRF.P11    03-NOV-81 10:29          REPORT CODING SECTION

 5681                                  .SBTTL  REPORT CODING SECTION
 5682
 5683
 5684                                  ;////////////////////////////////////////////////////////////////////////////
 5685                                  ;/ THE REPORT CODING SECTION CONTAINS THE
 5686                                  ;/ 'PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
 5687                                  ;////////////////////////////////////////////////////////////////////////////
 5688
 5689  021064                                  BGNRPT
 5690  021064                                                                  L$RPT::
 5691
 5692  021064                                  ENDRPT
 5693  021064                                                                  L10013:
 5694  021064  104425                                                                 TRAP    C$RPT
 5695
 5696
 5697
 5698
 5699
```

```
CZDMRF.P11    03-NOV-81 10:29          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.
                                       ;////////////////////////////////////////////////////////////////////////////////

 021066                                        BGNPROT
 021066                                                                          L$PROT::
 021066  177777                                .WORD   -1      ;DON'T CHK CSR ADRS
 021070  177777                                .WORD   -1      ;DON'T CHK MASSBUS UNIT NO.
 021072  177777                                .WORD   -1      ;DON'T CHK DRIVE NO.
 021074                                        ENDPROT
```

```
                                .SBTTL  INITIALIZE SECTION

                                ;////////////////////////////////////////////////////////////////////
                                ;/ THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ;/ AT THE BEGINNING OF THE TEST SEQUENCE ON THE NEXT UNIT.
                                ;////////////////////////////////////////////////////////////////////

                                        BGNINIT
                                                                                        L$INIT::

021074  010637  002324                  MOV     SP,PSTACK       ;SAVE BASE-LEVEL STACK POINTER
021100  005037  002330                  CLR     SUBRPC          ;CLEAR SUBR CALL PC
021104  005037  002404                  CLR     DISILO          ;CLEAR CURRENT STATE OF DISSI
021110  005037  002406                  CLR     CHPTYP          ;CLEAR USYRT CHIP TYPE INDICATOR
021114  005037  002376                  CLR     ERROR1          ;CLEAR ERROR FLAG
021120  005037  002410                  CLR     SAVLEN          ;CLEAR CHAR LENGTH FROM SETUP
021124  005737  002370                  TST     FRSTIM          ;SEE IF FIRST TIME THROUGH AFTER LOAD
021130  001007                          BNE     6$              ;BR IF NOT
021132  013737  000004  002372          MOV     @#4,SAVE4       ;SAVE ERROR TRAP VECTOR
021140  013737  000006  002374          MOV     @#6,SAVE6
021146  000406                          BR      9$
021150  013737  002372  000004  6$:     MOV     SAVE4,@#4       ;RESTORE ERROR TRAP VECTOR
021156  013737  002374  000006          MOV     SAVE6,@#6
021164  012737  000001  002370  9$:     MOV     #1,FRSTIM       ;MARK FLAG FOR NEXT TIME THROUGH
                                ;SEE IF PROGRAM JUST STARTED, BR IF YES
021172                                  READEF  #EF.START
021172  012700  000040                                                  MOV     #EF.START,R0
021176  104447                                                          TRAP    C$REFG
021200                                  BCOMPLETE       STARST
021200  103415                                                          BCS     STARST
                                ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
021202                                  READEF  #EF.RESTART
021202  012700  000037                                                  MOV     #EF.RESTART,R0
021206  104447                                                          TRAP    C$REFG
021210                                  BCOMPLETE       STARST
021210  103411                                                          BCS     STARST
                                ;SEE IF THIS IS A NEW PASS, BR IF YES
021212                                  READEF  #EF.NEW
021212  012700  000035                                                  MOV     #EF.NEW,R0
021216  104447                                                          TRAP    C$REFG
021220                                  BCOMPLETE       NEWST
021220  103411                                                          BCS     NEWST
                                ;SEE IF PROGRAM WAS JUST CONTINUED
021222                                  READEF  #EF.CONTINUE
021222  012700  000036                                                  MOV     #EF.CONTINUE,R0
021226  104447                                                          TRAP    C$REFG
021230                                  BCOMPLETE       ENDIT
021230  103467                                                          BCS     ENDIT
021232  000414                          BR      GETPRM
021234                          STARST:
021234  005037  002422                  CLR     STARES          ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
                                ;CLEAR DEVICE MAP
021240  005037  002412                  CLR     DEVMAP
021244                          NEWST:
021244  012737  177777  002322          MOV     #-1,LOGDEV      ;RESET LOGICAL DEVICE TO -1
021252  005237  002422                  INC     STARES          ;INCR NO. OF PASSES SINCE STA OR RES
```

```
5774  021256  012737  000001  002414          MOV     #BIT0,DEVPTR    ;INIT DEVICE MAP BIT POINTER
5775                                  ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
5776                                  ;   CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
5777  021264                          GETPRM:
5778  021264  005237  002322                  INC     LOGDEV          ;INCREMENT LOGICAL DEVICE NUMBER
5779  021270  023737  002322  002012          CMP     LOGDEV,L$UNIT   ;SEE IF MAXIMUM UNIT NO. EXCEEDED
5780  021276  002362                          BGE     NEWST           ;BR IF YES
5781  021300                                  GPHARD  LOGDEV,R1       ;GET P-TABLE POINTER INTO R1
5782  021300  013700  002322                                                          MOV     LOGDEV,R0
5783  021304  104442                                                                  TRAP    C$GPHRD
5784  021306  010001                                                                  MOV     R0,R1
5785  021310                                  BCOMPLETE       10$     ;BR IF DEVICE AVAILABLE
5786  021310  103403                                                                  BCS     10$
5787  021312  006337  002414                  ASL     DEVPTR          ;SHIFT DEVICE POINTER
5788  021316  000762                          BR      GETPRM          ;SKIP THIS DEVICE
5789  021320  053737  002414  002412  10$:    BIS     DEVPTR,DEVMAP   ;SET BIT FOR THIS DEVICE
5790  021326  006337  002414                  ASL     DEVPTR          ;SHIFT BIT POINTER
5791  021332  011137  002424                  MOV     (R1),MPCSR      ;STORE POINTER TO MICROPROCESSOR CSR'S
5792  021336  011137  002426                  MOV     (R1),BSEL1
5793  021342  005237  002426                  INC     BSEL1           ;GET POINTER TO BSEL1 (MAINTENANCE REGISTER)
5794  021346  013737  002426  002430          MOV     BSEL1,BSEL2
5795  021354  005237  002430                  INC     BSEL2           ;GET POINTER TO BSEL2
5796  021360  011137  002432                  MOV     (R1),SEL4
5797  021364  062737  000004  002432          ADD     #4,SEL4         ;GET POINTER TO SEL4
5798  021372  012137  002434                  MOV     (R1)+,SEL6
5799  021376  062737  000006  002434          ADD     #6,SEL6         ;STORE POINTER TO SEL6
5800  021404  011137  002454                  MOV     (R1),RUNINH     ;GET STATE OF MICROPROCESSOR RUN SWITCH
5801  021410                          ENDIT:
5802  021410                                  ENDINIT
5803  021410                                                          L10015:
5804  021410  104411                                                          TRAP    C$INIT
5805
5806
5807
5808
5809
5810
```

```
5811                                    .SBTTL  AUTO DROP UNIT SECTION
5812
5813                                    ;////////////////////////////////////////////////////////////////////////////////
5814                                    ;/ THE AUTO DROP CODING DETERMINES WHETHER OR NOT THE DEVICE WHOSE P-TABLE
5815                                    ;/ WAS JUST OBTAINED IS READY FOR TESTING, AND IT IS DROPPED IF NOT READY.
5816                                    ;////////////////////////////////////////////////////////////////////////////////
5817  021412                                    BGNAUTO
5818  021412                                                                                          L$AUTO::
5819                                    ;ESTABLISH CPU PRIORITY = 7
5820  021412                                    SETPRI  #PRI07
5821  021412  012700  000340                                                                                MOV     #PRI07,R0
5822  021416  104441                                                                                        TRAP    C$SPRI
5823  021420  012737  021442  000004              MOV     #6$,@#4         ;SET UP NON-EXISTENT MEMORY ERROR TRAP VECTOR
5824  021426  012737  000340  000006              MOV     #PRI07,@#6
5825  021434  005777  160764                      TST     @MPCSR          ;ADDRESS SELO
5826  021440  000405                              BR      9$              ;TAKE THIS BR IF DEVICE RESPONDS
5827                                    ;COME HERE IF DEVICE CSR IS NON-EXISTENT
5828  021442  062706  000004            6$:       ADD     #4,SP           ;CLEAN UP THE STACK POINTER
5829  021446                                    DODU    LOGDEV          ;DROP THIS UNIT FROM TESTING
5830  021446  013700  002322                                                                                MOV     LOGDEV,R0
5831  021452  104451                                                                                        TRAP    C$DODU
5832  021454  013737  002372  000004  9$:       MOV     SAVE4,@#4       ;RESTORE ERROR TRAP VECTOR
5833  021462  013737  002374  000006              MOV     SAVE6,@#6
5834  021470                                    ENDAJTO
5835  021470                                                                                          L10016:
5836  021470  10446'                                                                                        TRAP    C$AUTC
5837
5838
5839
5840
5841
```

```
5842
 843
5844
5845
5846
5847
5848
5849  021472                          .SBTTL  CLEANUP CODING SECTION
5850  021472
5851                                  ;////////////////////////////////////////////////////////////////////////
5852                                  ;/ THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
5853  021472                          ;/ AT THE END OF THE TEST SEQUENCE ON A PARTICULAR UNIT.
5854  021472                          ;////////////////////////////////////////////////////////////////////////
5855  021472  104412                          BGNCLN
5856                                                                  L$CLEAN::
5857
5858                                          ENDCLN
5859                                                                  L10017:
5860                                                                          TRAP    C$CLEAN
```

```
5861                                    .SBTTL  DROP UNIT SECTION
5862
5863                                    ;////////////////////////////////////////////////////////////////////////////////
5864                                    ;/ THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
5865                                    ;/ TO NO LONGER BE TESTED.
5866                                    ;////////////////////////////////////////////////////////////////////////////////
5867
5868  021474                                    BGNDU
5869  021474                                                                          L$DU::
5870                                    ;ISSUE UNIBUS RESET TO CLEAN UP
5871  021474                                    BRESET
5872  021474  104433                                                                          TRAP    C$RESET
5873                                    ;PRINT 'UNIT XX DROPPED"
5874  021476                                    PRINTF  #FMT27,LOGDEV
5875  021476  013746  002322                                                                 MOV     LOGDEV,-(SP)
5876  021502  012746  021524                                                                 MOV     #FMT27,-(SP)
5877  021506  012746  000002                                                                 MOV     #2,-(SP)
5878  021512  010600                                                                          MOV     SP,R0
5879  021514  104417                                                                          TRAP    C$PNTF
5880  021516  062706  000006                                                                 ADD     #6,SP
5881  021522                                    ENDDU
5882  021522                                                                          L10020:
5883  021522  104453                                                                          TRAP    C$DU
5884
5885  021524  047045  040445  047125  FMT27:  .ASCIZ  /%N%AUNIT %D2%A DROPPED%N/
5886  021532  052111  022440  031104
5887  021540  040445  042040  047522
5888  021546  050120  042105  047045
5889  021554     000
5890          021556                    .EVEN
5891
5892
5893
5894
5895
```

```
CZDMRF.P11    03-NOV-81 10:29          ADD UNIT SECTION

  5896                                 .SBTTL  ADD UNIT SECTION
  5897
  5898                                 ;////////////////////////////////////////////////////////////////////
  5899                                 ;/ THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
  5900                                 ;/  TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING.  IF
  5901                                 ;/  "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
  5902                                 ;////////////////////////////////////////////////////////////////////
  5903
  5904  021556                               BGNAU
  5905  021556                                                                  L$AU::
  5906  021556                               ENDAU
  5907  021556                                                                  L10021:
  5908  021556  104452                                                                 TRAP    C$AU
  5909
  5910
  5911
  5912
  5913
  5914
```

```
                                        .SBTTL  HARDWARE TESTS
                                        ;;*****************************************************************************
                                        .SBTTL      TEST 1 - MICROPROCESSOR CSR ADDRESSING TEST (SEL0)
                                        ;*
                                        ;* THIS TEST ADDRESSES THE FIRST MICROPROCESSOR CSR (SEL0), TO MAKE SURE
                                        ;* THAT A NON-EXISTENT MEMORY TIME-OUT TRAP DOES NOT OCCUR WHILE
                                        ;* ATTEMPTING TO ADDRESS THE MICROPROCESSOR.
                                        ;;*****************************************************************************
021560                                  BGNTST
021560                                                                                          T1::
                                        ;ESTABLISH CPU PRIORITY = 7
021560                                          SETPRI  #PRI07
021560  012700  000340                                                                                  MOV     #PRI07,R0
021564  104441                                                                                          TRAP    C$SPRI
021566  012737  021610  000004                 MOV     #6$,@#4         ;SET UP NON-EXISTENT MEMORY ERROR TRAP VECTOR
021574  012737  000340  000006                 MOV     #PRI07,@#6
021602  005777  160616                         TST     @#MPCSR         ;ADDRESS SEL0
021606  000406                                 BR      9$              ;TAKE THIS BR IF DEVICE RESPONDS
                                        ;COME HERE IF DEVICE CSR IS NON-EXISTENT
021610  062706  000004                 6$:     ADD     #4,SP           ;CLEAN UP THE STACK POINTER
                                        ;REPORT CSR ADDRESS TIME-OUT
021614                                          ERRDF   1,EM1,ERR1
021614  104455                                                                                          TRAP    C$ERDF
021616  000001                                                                                          .WORD   1
021620  012103                                                                                          .WORD   EM1
021622  014556                                                                                          .WORD   ERR1
021624  013737  002372  000004         9$:     MOV     SAVE4,@#4       ;RESTORE ERROR TRAP VECTOR
021632  013737  002374  000006                 MOV     SAVE6,@#6
021640                                  ENDTST
021640                                                                                  L10022:
021640  104401                                                                                  TRAP    C$ETST
                                        ;;*****************************************************************************
                                        .SBTTL      TEST 2 - INBUS/OUTBUS REG 14 INITIALIZATION TEST
                                        ;*
                                        ;* MASTER CLEAR (MCLR) IS SET IN THE MICROPROCESSOR, IBUS REG 14 IS READ
                                        ;* AND COMPARED TO 200.
                                        ;;*****************************************************************************
021642                                  BGNTST
021642                                                                                          T2::
021642  012737  000014  002356                 MOV     #14,REGNUM      ;SET LU REG NO. = 14
021650  004737  003554                         JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
021654  004737  003650                         JSR     PC,READLU       ;READ REG 14
021660  123737  002342  003004                 CMPB    REDBYT,PATM+4   ;CHK FOR INITIALIZED STATE
021666  001416                                 BEQ     6$              ;BR IF YES
021670  005037  002362                         CLR     GOODAT          ;SET EXPECTED REG CONTENTS = 000
021674  113737  003004  002362                 MOVB    PATM+4,GOODAT   ;SET EXPECTED DATA
```

```
5971  021702  013737  002342  002364          MOV     REDBYT,BADDAT   ;SET ACTUAL REG CONTENTS
5972  021710  004737  003774                  JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
5973                                  ;REPORT REG NOT CLEARED BY MASTER CLEAR
5974  021714                                  ERRDF   2,EM2,ERR2
5975  021714  104455                                                          TRAP    C$ERDF
5976  021716  000002                                                          .WORD   2
5977  021720  012137                                                          .WORD   EM2
5978  021722  014610                                                          .WORD   ERR2
5979  021724                          6$:
5980  021724                          ENDTST
5981  021724                                                          L10023:
5982  021724  104401                                                          TRAP    C$ETST
5983
5984
5985
5986
5987
5988                                  ;;*************************************************************************
5989                                  .SBTTL      TEST 3 - INBUS/OUTBUS REG 14 READ/WRITE BIT TEST
5990                                  ;*
5991                                  ;* WRITE, READ, AND COMPARE ALL WORDS OF DATA PATTERN A INTO REG 14,
5992                                  ;* A BYTE AT A TIME. NON-R/W BITS ARE MASKED OFF TO 0 BEFORE WRITING AND
5993                                  ;* READING.
5994                                  ;*    DATA PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
5995                                  ;*                     375,373,367,357,337,277,177.
5996                                  ;;*************************************************************************
5997  021726                          BGNTST
5998  021726                                                          T3::
5999  021726  004737  003554                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
6000  021732  012737  000014  002356          MOV     #14,REGNUM      ;SET LU REG NO. = 14
6001  021740  012701  002547                  MOV     #PATA,R1        ;GET POINTER TO DATA PAT IN R1
6002  021744                          3$:
6003  021744                                  BGNSEG
6004  021744  104404                                                          TRAP    C$BSEG
6005  021746  111137  002344                  MOVB    (R1),WRIBYT     ;GET A BYTE OF PAT A
6006  021752  143737  002536  002344          BICB    R14NRW,WRIBYT   ;MASK OFF NON-READ/WRITE BITS
6007  021760  004737  003726                  JSR     PC,WRITLU       ;WRITE DATA BYTE INTO REG 14
6008  021764  004737  003650                  JSR     PC,READLU       ;READ DATA BYTE FROM REG 14
6009  021770  143737  002536  002342          BICB    R14NRW,REDBYT   ;MASK OFF NON-READ/WRITE BITS
6010  021776  123737  002342  002344          CMPB    REDBYT,WRIBYT   ;COMPARE BYTE READ TO BYTE WRITTEN
6011  022004  001414                          BEQ     6$              ;BR IF BYTES MATCH
6012  022006  013737  002344  002362          MOV     WRIBYT,GOODAT   ;SET EXPECTED REG CONTENTS
6013  022014  013737  002342  002364          MOV     REDBYT,BADDAT   ;SET ACTUAL REG CONTENTS
6014  022022  004737  003774                  JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
6015                                  ;REPORT LINE UNIT REG MISCOMPARE
6016  022026                                  ERRDF   3,EM3,ERR2
6017  022026  104455                                                          TRAP    C$ERDF
6018  022030  000003                                                          .WORD   3
6019  022032  012176                                                          .WORD   EM3
6020  022034  014610                                                          .WORD   ERR2
6021  022036                          6$:
6022  022036                                  ENDSEG
6023  022036                                                          10000$:
6024  022036  104405                                                          TRAP    C$ESEG
6025  022040  005201                          INC     R1              ;INCREMENT DATA PATTERN POINTER
6026  022042  020127  002573                  CMP     R1,#PATB        ;SEE IF ALL WORDS OF PATTERN A USED YET
```

```
                                        BLO     3$              ;BR IF NOT DONE YET
                                ENDTST
                                                                                L10024:
                                                                                        TRAP    C$ETST

                                ;;*************************************************************************
                                .SBTTL      TEST 4 - REG 14 MASTER CLEAR TEST
                                ;* WRITE 377 INTO REG 14, ISSUE MASTER CLEAR, READ REG 14 AND COMPARE
                                ;* TO 200.
                                ;;*************************************************************************
                                BGNTST
                                                                                T4::
                                        JSR     PC,MSTCLR       ;PERFORM MASTER CLEAR
                                        MOV     #14,REGNUM      ;SET LU REG NO. = 14
                                        MOVB    #377,WRIBYT     ;SET DATA BYTE = 377
                                        JSR     PC,WRITLU       ;WRITE 377 INTO REG 14
                                        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
                                        JSR     PC,READLU       ;READ REG 14
                                        CMPB    REDBYT,PATM+4   ;CHK FOR INIT'D STATE
                                        BEQ     6$              ;BR IF REG GOT CLEARED
                                        CLR     GOODAT
                                        MOVB    PATM+4,GOODAT   ;SET EXPECTED DATA
                                        MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
                                        JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
                                ;REPORT REG NOT CLEARED BY MASTER CLEAR
                                        ERRDF   2,EM2,ERR2
                                                                                        TRAP    C$ERDF
                                                                                        .WORD   2
                                                                                        .WORD   EM2
                                                                                        .WORD   ERR2
                                6$:
                                ENDTST
                                                                                L10025:
                                                                                        TRAP    C$ETST

                                ;;*************************************************************************
                                .SBTTL      TEST 5 - REG 14 UNIBUS RESET (INIT) TEST
                                ;* WRITE 377 INTO REG 14, ISSUE UNIBUS RESET (INIT), READ REG 14 AND COMPARE
                                ;* TO 200.
                                ;;*************************************************************************
                                BGNTST
                                                                                T5::
                                        JSR     PC,MSTCLR       ;PERFORM MASTER CLEAR
                                        MOV     #14,REGNUM      ;SET LU REG NO. = 14
                                        MOVB    #377,WRIBYT     ;SET DATA BYTE = 377
                                        JSR     PC,WRITLU       ;WRITE 377 INTO REG 14
                                        BRESET                  ;ISSUE UNIBUS RESET (INIT)
                                                                                        TRAP    C$RESET
```

```
 6083  022202  142777  000200  160216          BICB    #RUN,@BSEL1     ;CLEAR RUN BIT
 6084  022210  012701  000024                  MOV     #20.,R1         ;INIT LOOP COUNTER
 6085  022214  000240                  2$:     NOP
 6086  022216  005301                          DEC     R1              ;DECR COUNTER
 6087  022220  001375                          BNE     2$              ;BR TO STALL
 6088  022222  004737  003650                  JSR     PC,READLU       ;READ REG 14
 6089  022226  142737  000100  002342          BICB    #TXEN,REDBYT    ;CLEAR UNPREDICTABLE BIT
 6090  022234  123737  002342  003004          CMPB    REDBYT,PATM+4   ;CHK FOR INIT'D STATE
 6091  022242  001416                          BEQ     6$              ;BR IF REG GOT CLEARED
 6092  022244  005037  002362                  CLR     GOODAT
 6093  022250  113737  003004  002362          MOVB    PATM+4,GOODAT   ;SET EXPECTED DATA
 6094  022256  013737  002342  002364          MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
 6095  022264  004737  003774                  JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
 6096                          ;REPORT REG NOT CLEARED BY UNIBUS RESET (INIT)
 6097  022270                  ERRDF   4,EM4,ERR2
 6098  022270  104455                                                  TRAP    C$ERDF
 6099  022272  000004                                                  .WORD   4
 6100  022274  012215                                                  .WORD   EM4
 6101  022276  014610                                                  .WORD   ERR2
 6102  022300                  6$:
 6103  022300                  ENDTST
 6104  022300                                                  L10026:
 6105  022300  104401                                                  TRAP    C$ETST
 6106
 6107
 6108
 6109
 6110
 6111                          ;;*****************************************************************************
 6112                          .SBTTL      TEST 6 - LINE UNIT FALSE SELECTION TEST
 6113                          ;*
 6114                          ;* FIRST, A MASTER CLEAR IS PERFORMED. THEN, THE PROGRAM SINGLE-STEPS THE
 6115                          ;* MICROPROCESSOR THROUGH AN INSTRUCTION WHICH LOADS 041 (OCT) INTO THE MAR
 6116                          ;* REGISTER (OBUS* ADRS 14). THEN, THE LINE UNIT REGISTER 14 IS READ AND CHECKED
 6117                          ;* TO BE UNAFFECTED (STILL = 0). THIS TEST IS INTENDED TO DETECT A FALSE
 6118                          ;* SELECTION OF THE LINE UNIT REGISTERS, WHEN THE LINE UNIT IS NOT BEING
 6119                          ;* ACCESSED.
 6120                          ;;*****************************************************************************
 6121  022302                  BGNTST
 6122  022302                                                          T6::
 6123  022302  004737  003554          JSR     PC,MSTCLR       ;ISSUE A MASTER CLEAR
 6124  022306  012737  000014  002356  MOV     #14,REGNUM      ;SET LU REG NO. = 14
 6125  022314  013701  002356          MOV     REGNUM,R1       ;SET DESTINATION = OBUS* REG 14
 6126  022320  052701  000100          BIS     #100,R1         ;SET SOURCE = BSEL4
 6127  022324  052701  121000          BIS     #MVIXOX,R1      ;SET REST OF MOVE INSTRUCTION
 6128  022330  010137  022346          MOV     R1,2$           ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
 6129  022334  112777  000041  160070  MOVB    #041,@BSEL4     ;SET DATA BYTE = 041
 6130  022342  004737  003516          JSR     PC,STPCLK       ;EXECUTE MOVE INSTRUCTION
 6131  022346  000000          2$:     .WORD   0               ;INSTRUCTION GOES HERE
 6132  022350  004737  003650          JSR     PC,READLU       ;READ LU REG 14
 6133  022354  123737  002342  003004  CMPB    REDBYT,PATM+4   ;CHECK FOR LU REG 14 UNCHANGED
 6134  022362  001416                  BEQ     4$              ;BR IF LU REG 14 UNCHANGED
 6135  022364  005037  002362          CLR     GOODAT          ;SET EXPECTED DATA
 6136  022370  113737  003004  002362  MOVB    PATM+4,GOODAT
 6137  022376  013737  002342  002364  MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
 6138  022404  004737  003774          JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
```

```
                                ;REPORT REGISTER MISCOMPARE
                                        ERRDF   3.EM3,ERR2
                                                                                TRAP    C$ERDF
                                                                                .WORD   3
                                                                                .WORD   EM3
                                                                                .WORD   ERR2
                                4$:
                                ENDTST
                                                                        L10027:
                                                                                TRAP    C$ETST


                                ;;*************************************************************************************
                                .SBTTL      TEST 7 - INBUS REG MASTER CLEAR TEST
                                ;*
                                ;* FIRST, ALL READ/WRITE BITS OF REGS 10-17 ARE SET BY LOADING A
                                ;* DIFFERENT WORD OF PATTERN G INTO EACH REG.  THEN,
                                ;* A MASTER CLEAR IS ISSUED AND EACH REG IS READ AND COMPARED TO A WORD OF
                                ;* PATTERN M, WHICH CONTAINS THE INITIALIZED STATES OF THE REGS. (UNPREDICTABLE
                                ;* BITS ARE MASKED OFF TO 0 BEFORE COMPARISON).
                                ;*   PATTERN G = 000,000,240,120,177,000,000,001
                                ;*   PATTERN M = 000,020,000,000,200,000,000,051
                                ;;*************************************************************************************
                                BGNTST
                                                                                T7::
                                        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
                                        MOV     #10,REGNUM      ;INIT REG NO. TO 10
                                        MOV     #PATG,R1        ;INIT DATA PATTERN POINTER
                                2$:     MOVB    (R1)+,WRIBYT    ;SET DATA PATTERN BYTE TO BE WRITTEN
                                        JSR     PC,WRITLU       ;WRITE BYTE INTO REG
                                        INC     REGNUM          ;INCREMENT REG NO. FOR WRITING
                                        CMP     R1,#PATH        ;SEE IF ALL BYTES WRITTEN YET
                                        BLO     2$              ;BR IF NOT DONE WRITING YET
                                        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
                                        MOV     #10,REGNUM      ;INIT LU REG NO. TO 10
                                        MOV     #PATM,R2        ;INIT DATA PATTERN POINTER
                                        MOV     #UPBITS,R1      ;INIT POINTER TO UNPREDICTABLE BITS
                                3$:
                                        JSR     PC,READLU       ;READ A LINE UNIT REG
                                        BICB    (R1)+,REDBYT    ;MASK OUT UNPREDICTABLE BITS FOR THIS REG
                                        CMPB    REDBYT,(R2)     ;COMPARE MASKED DATA TO EXPECTED
                                        BEQ     6$              ;BR IF DATA READ IS OK
                                        CLR     GOODAT
                                        MOVB    (R2),GOODAT     ;SET EXPECTED DATA
                                        MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
                                        JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
                                ;REPORT REG NOT CLEARED BY MASTER CLEAR
                                        ERRDF   2,EM2,ERR2
                                                                                TRAP    C$ERDF
                                                                                .WORD   2
                                                                                .WORD   EM2
                                                                                .WORD   ERR2
                                        ESCAPE  TST
```

```
6195  022554  104410                                                          TRAP    C$ESCAPE
6196  022556  000016                                                          .WORD   L10030-.
6197  022560  005237  002356    6$:     INC     REGNUM          ;INCREMENT REG NO.
6198  022564  005202              INC     R2              ;INCR DATA PATTERN POINTER
6199  022566  020227  003010      CMP     R2,#PATN        ;SEE IF ALL DONE YET
6200  022572  103744              BLO     3$              ;BR IF NOT DONE YET
6201  022574                    ENDTST
6202  022574                                                          L10030:
6203  022574  104401                                                          TRAP    C$ETST
6204
6205
6206
6207
6208
6209                            ;;*************************************************************************
6210                            .SBTTL      TEST 8 - REGISTER 10-17 ADDRESSING TEST
6211                            ;*
6212                            ;* FIRST, A MASTER CLEAR IS ISSUED. THEN,
6213                            ;* WRITE A DIFFERENT WORD OF DATA PATTERN B INTO EACH OF REGS 10-17,
6214                            ;* AND AFTER EACH WRITE, READ AND COMPARE ALL REGS TO EXPECTED VALUES.
6215                            ;* UNPREDICTABLE BITS ARE MASKED OFF TO 0 WHEN READ FOR COMPARISON.
6216                            ;*    PATTERN B = 000,000,040,100,220,000,000,051
6217                            ;;*************************************************************************
6218  022576                    BGNTST
6219  022576                                                          T8::
6220  022576  004737  003554              JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
6221  022602  012701  002456              MOV     #REDDAT,R1      ;INIT POINTER TO EXPECTED DATA AREA
6222  022606  012702  003000              MOV     #PATM,R2        ;GET POINTER TO PATTERN M
6223  022612  012703  000010              MOV     #8.,R3          ;SET COUNTER
6224  022616  112221          3$:     MOVB    (R2)+,(R1)+     ;LOAD BYTE OF PATRN INTO EXPECTED DATA AREA
6225  022620  005303              DEC     R3              ;DECR COUNTER
6226  022622  001375              BNE     3$              ;BR IF NOT DONE LOADING YET
6227  022624  005001              CLR     R1              ;INIT DATA PATTERN INDEX FOR WRITING
6228  022626  010137  002356      6$:     MOV     R1,REGNUM
6229  022632  062737  000010  002356      ADD     #10,REGNUM      ;GET REG NO. FOR WRITING
6230  022640  116137  002573  002344      MOVB    PATB(R1),WRIBYT ;SET DATA BYTE TO BE WRITTEN
6231  022646  113761  002344  002456      MOVB    WRIBYT,REDDAT(R1) ;SET EXPECTED DATA FOR READ
6232  022654  146161  002526  002456      BICB    UPBITS(R1),REDDAT(R1) ;MASK OUT UNPREDICTABLE BITS
6233  022662  004737  003726              JSR     PC,WRITLU       ;WRITE DATA BYTE INTO REG
6234  022666  005003              CLR     R3              ;INIT DATA PAT INDEX FOR READS
6235  022670  010337  002356      9$:     MOV     R3,REGNUM
6236  022674  062737  000010  002356      ADD     #10,REGNUM      ;GET REG NO. FOR READING
6237  022702  004737  003650              JSR     PC,READLU       ;READ A LINE UNIT REG
6238  022706  146337  002526  002342      BICB    UPBITS(R3),REDBYT ;MASK OUT UNPREDICTABLE BITS
6239  022714  023727  002356  000011      CMP     REGNUM,#11      ;SEE IF READING REG 11
6240  022722  001006              BNE     10$             ;BR IF NOT
6241  022724  142737  000020  002342      BICB    #ORDY,REDBYT    ;MASK ORDY BIT IN ACTUAL BYTE
6242  022732  142763  000020  002456      BICB    #ORDY,REDDAT(R3) ;MASK ORDY BIT IN EXPECTED BYTE
6243  022740  123763  002342  002456  10$:CMPB    REDBYT,REDDAT(R3) ;COMPARE BYTE READ TO EXPECTED
6244  022746  001420              BEQ     12$             ;BR IF DATA MATCHES
6245  022750  005037  002362              CLR     GOODAT
6246  022754  116337  002456  002362      MOVB    REDDAT(R3),GOODAT ;SET EXPECTED DATA
6247  022762  013737  002342  002364      MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
6248  022770  004737  003774              JSR     PC,GETREG       ;READ AND STORE REGS 10-17 FOR PRINTOUT
6249                          ;REPORT REG MISCOMPARE
6250  022774                              ERRDF   3,EM3,ERR2
```

```
6251  022774  104455                                                          TRAP    C$ERDF
6252  022776  000003                                                          .WORD   3
6253  023000  012176                                                          .WORD   EM3
6254  023002  014610                                                          .WORD   ERR2
6255  023004                      ESCAPE  TST
6256  023004  104410                                                          TRAP    C$ESCAPE
6257  023006  000022                                                          .WORD   L10031-.
6258  023010  005203          12$:    INC     R3          ;INCREMENT DATA PATTERN INDEX FOR READING
6259  023012  020327  000010        CMP     R3,#10      ;SEE IF ALL REGS READ YET
6260  023016  002724                BLT     9$          ;BR IF NOT
6261  023020  005201                INC     R1          ;INCREMENT DATA PAT INDEX FOR WRITING
6262  023022  020127  000010        CMP     R1,#10      ;SEE IF ALL REGS WRITTEN YET
6263  023026  002677                BLT     6$          ;BR IF NOT
6264  023030                    ENDTST
6265  023030                                                          L10031:
6266  023030  104401                                                          TRAP    C$ETST
6267
6268
6269
6270
6271
6272                      ;;*****************************************************************************
6273                      .SBTTL      TEST 9 - REG 11 READ/WRITE BIT TEST
6274                      ;*
6275                      ;* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN C INTO REG 11 :
6276                      ;*      DATA PATTERN C = 020,020,020.
6277                      ;;*****************************************************************************
6278  023032              BGNTST
6279  023032                                                              T9::
6280  023032  004737  003554        JSR     PC,MSTCLR   ;ISSUE MASTER CLEAR
6281  023036  012737  000011  002356    MOV     #11,REGNUM  ;SET LU REG NO. = 11
6282  023044  012701  002603        MOV     #PATC,R1    ;GET POINTER TO DATA PAT IN R1
6283  023050                    3$:
6284  023050                        BGNSEG
6285  023050  104404                                                          TRAP    C$BSEG
6286  023052  111137  002344        MOVB    (R1),WRIBYT ;GET A BYTE OF PAT C
6287  023056  004737  003726        JSR     PC,WRITLU   ;WRITE DATA BYTE INTO REG 11
6288  023062  004737  003650        JSR     PC,READLU   ;READ DATA BYTE FROM REG 11
6289  023066  143737  002527  002342    BICB    UPBITS+1,REDBYT ;MASK OUT UNPREDICTABLE BITS
6290  023074  123737  002342  002344    CMPB    REDBYT,WRIBYT   ;COMPARE BYTE READ TO BYTE WRITTEN
6291  023102  001414                BEQ     6$          ;BR IF BYTES MATCH
6292  023104  013737  002344  002362    MOV     WRIBYT,GOODAT   ;SET EXPECTED REG CONTENTS
6293  023112  013737  002342  002364    MOV     REDBYT,BADDAT   ;SET ACTUAL REG CONTENTS
6294  023120  004737  003774        JSR     PC,GETREG   ;GET REGS FOR PRINTOUT
6295                    ;REPORT LINE UNIT REG MISCOMPARE
6296  023124                        ERRDF   3,EM3,ERR2
6297  023124  104455                                                          TRAP    C$ERDF
6298  023126  000003                                                          .WORD   3
6299  023130  012176                                                          .WORD   EM3
6300  023132  014610                                                          .WORD   ERR2
6301  023134                    6$:
6302  023134                        ENDSEG
6303  023134                                                          10000$:
6304  023134  104405                                                          TRAP    C$ESEG
6305  023136  005201                INC     R1          ;INCREMENT DATA PATTERN POINTER
6306  023140  020127  002606        CMP     R1,#PATD    ;SEE IF ALL WORDS OF PATTERN C USED YET
```

```
                        BLO     3$              ;BR IF NOT DONE YET
                ENDTST
                                                                        L10032:
                                                                                TRAP    C$ETST
                ;;**********************************************************************************
                .SBTTL      TEST 10 - REG 12 READ/WRITE BIT TEST
                ;*
                ;* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN D INTO REG 12 :
                ;*      DATA PATTERN D = 000,040,000.
                ;;**********************************************************************************
                BGNTST
                                                                        T10::
                        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
                        MOV     #12,REGNUM      ;SET LU REG NO. = 12
                        MOV     #PATD,R1        ;GET POINTER TO DATA PAT IN R1
                3$:
                        BGNSEG
                                                                                TRAP    C$BSEG
                        MOVB    (R1),WRIBYT     ;GET A BYTE OF PAT D
                        JSR     PC,WRITLU       ;WRITE DATA BYTE INTO REG 12
                        JSR     PC,READLU       ;READ DATA BYTE FROM REG 12
                        BICB    UPBITS+2,REDBYT ;MASK OUT UNPREDICTABLE BITS
                        CMPB    REDBYT,WRIBYT   ;COMPARE BYTE READ TO BYTE WRITTEN
                        BEQ     6$              ;BR IF BYTES MATCH
                        MOV     WRIBYT,GOODAT   ;SET EXPECTED REG CONTENTS
                        MOV     REDBYT,BADDAT   ;SET ACTUAL REG CONTENTS
                        JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
                ;REPORT LINE UNIT REG MISCOMPARE
                        ERRDF   3,EM3,ERR2
                                                                                TRAP    C$ERDF
                                                                                .WORD   3
                                                                                .WORD   EM3
                                                                                .WORD   ERR2
                6$:
                        ENDSEG
                                                                        10000$:
                                                                                TRAP    C$ESEG
                        INC     R1              ;INCREMENT DATA PATTERN POINTER
                        CMP     R1,#PATE        ;SEE IF ALL WORDS OF PATTERN D USED YET
                        BLO     3$              ;BR IF NOT DONE YET
                ENDTST
                                                                        L10033:
                                                                                TRAP    C$ETST
                ;;**********************************************************************************
                .SBTTL      TEST 11 - REG 13 READ/WRITE BIT TEST
                ;*
```

```
6363
6364
6365
6366  023266                        ;* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN E INTO REG 13 :
6367  023266                        ;*      DATA PATTERN E = 000,120,020,100,120,000.
6368  023266  004737  003554        ;;**********************************************************************
6369  023272  012737  000013  002356 BGNTST
6370  023300  012701  002611                                                        T11::
6371  023304                                JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
6372  023304                                MOV     #13,REGNUM      ;SET LU REG NO. = 13
6373  023304  104404                       MOV     #PATE,R1        ;GET POINTER TO DATA PAT IN R1
6374  023306  111137  002344        3$:
6375  023312  004737  003726                BGNSEG
6376  023316  004737  003650                                                              TRAP    C$BSEG
6377  023322  143737  002531  002342        MOVB    (R1),WRIBYT     ;GET A BYTE OF PAT E
6378  023330  123737  002342  002344        JSR     PC,WRITLU       ;WRITE DATA BYTE INTO REG 13
6379  023336  001414                       JSR     PC,READLU       ;READ DATA BYTE FROM REG 13
6380  023340  013737  002344  002362        BICB    UPBITS+3,REDBYT ;MASK OUT UNPREDICTABLE BITS
6381  023346  013737  002342  002364        CMPB    REDBYT,WRIBYT   ;COMPARE BYTE READ TO BYTE WRITTEN
6382  023354  004737  003774                BEQ     6$              ;BR IF BYTES MATCH
6383                                        MOV     WRIBYT,GOODAT   ;SET EXPECTED REG CONTENTS
6384  023360                                MOV     REDBYT,BADDAT   ;SET ACTUAL REG CONTENTS
6385  023360  104455                       JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
6386  023362  000003                ;REPORT LINE UNIT REG MISCOMPARE
6387  023364  012176                       ERRDF   3,EM3,ERR2
6388  023366  014610                                                              TRAP    C$ERDF
6389  023370                                                                      .WORD   3
6390  023370                                                                      .WORD   EM3
6391  023370                                                                      .WORD   ERR2
6392  023370  104405                6$:
6393  023372  005201                       ENDSEG
6394  023374  020127  002617                                                        10000$:
6395  023400  103741                                                                TRAP    C$ESEG
6396  023402                                INC     R1              ;INCREMENT DATA PATTERN POINTER
6397  023402                                CMP     R1,#PATF        ;SEE IF ALL WORDS OF PATTERN E USED YET
6398  023402  104401                       BLO     3$              ;BR IF NOT DONE YET
6399                                ENDTST
6400
6401                                                                    L10034:
6402                                                                            TRAP    C$ETST
6403
6404                                ;;**********************************************************************
6405                                .SBTTL      TEST 12 - REG 17 READ/WRITE BIT TEST
6406                                ;*
6407                                ;* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN F INTO REG 17 :
6408                                ;*      DATA PATTERN F = 050,051,050.
6409                                ;;**********************************************************************
6410  023404                        BGNTST
6411  023404                                                                T12::
6412  023404  004737  003554                JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
6413  023410  012737  000017  002356        MOV     #17,REGNUM      ;SET LU REG NO. = 17
6414  023416  012701  002617                MOV     #PATF,R1        ;GET POINTER TO DATA PAT IN R1
6415  023422                        3$:
6416  023422                                BGNSEG
6417  023422  104404                                                              TRAP    C$BSEG
6418  023424  111137  002344                MOVB    (R1),WRIBYT     ;GET A BYTE OF PAT F
```

```
6419  023430  004737  003726          JSR     PC,WRITLU       ;WRITE DATA BYTE INTO REG 17
6420  023434  004737  003650          JSR     PC,READLU       ;READ DATA BYTE FROM REG 17
6421  023440  143737  002535  002342  BICB    UPBITS+7,REDBYT ;MASK OUT UNPREDICTABLE BITS
6422  023446  123737  002342  002344  CMPB    REDBYT,WRIBYT   ;COMPARE BYTE READ TO BYTE WRITTEN
6423  023454  001414                  BEQ     6$              ;BR IF BYTES MATCH
6424  023456  013737  002344  002362  MOV     WRIBYT,GOODAT   ;SET EXPECTED REG CONTENTS
6425  023464  013737  002342  002364  MOV     REDBYT,BADDAT   ;SET ACTUAL REG CONTENTS
6426  023472  004737  003774          JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
6427                          ;REPORT LINE UNIT REG MISCOMPARE
6428  023476                  ERRDF   3,EM3,ERR2
6429  023476  104455                                                  TRAP    C$ERDF
6430  023500  000003                                                  .WORD   3
6431  023502  012176                                                  .WORD   EM3
6432  023504  014610                                                  .WORD   ERR2
6433  023506                  6$:
6434  023506                  ENDSEG
6435  023506                                                      10000$:
6436  023506  104405                                                  TRAP    C$ESEG
6437  023510  005201          INC     R1              ;INCREMENT DATA PATTERN POINTER
6438  023512  020127  002622  CMP     R1,#PATG        ;SEE IF ALL WORDS OF PATTERN F USED YET
6439  023516  103741          BLO     3$              ;BR IF NOT DONE YET
6440  023520                  ENDTST
6441  023520                                                      L10035:
6442  023520  104401                                                  TRAP    C$ETST
6443
6444
6445
6446
6447
6448                          ;;*****************************************************************************
6449                          .SBTTL      TEST 13 - MAINTENANCE CLOCK BIT TEST
6450                          ;*
6451                          ;* FIRST, A MASTER CLEAR IS ISSUED TO INIT ALL REGS. THEN, THE MICROPROCESSOR
6452                          ;* IS PLACED IN A LOOP ON AN INSTRUCTION, BY SETTING THE INSTRUCTION IN SEL6
6453                          ;* AND SETTING ROMI AND RUN IN BSEL1. THE INSTRUCTION IS ONE WHICH REPETITIVELY
6454                          ;* READS LINE UNIT REG 17 INTO BSEL2. THE PDP-11 CAN THEN SCAN BSEL2 TO MONITOR
6455                          ;* THE MAINTENANCE CLOCK BIT, MCLK. THE FOLLOWING SEQUENCE IS THEN PERFORMED
6456                          ;* TO MONITOR MCLK :
6457                          ;* - THE PROGRAM REPEATEDLY CHECKS THE MCLK BIT FOR THE 1 STATE, AND IF IT IS
6458                          ;*   NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC (DEPENDING ON THE PROCESSOR)
6459                          ;*   AN ERROR IS REPORTED. (THE MAINTENANCE CLOCK HAS A PERIOD OF 41.6 MICRO-
6460                          ;*   SEC).
6461                          ;* - THE PROGRAM NEXT REPEATEDLY CHECKS THE MCLK BIT FOR THE 0 STATE, AND IF
6462                          ;*   IT IS NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC AN ERROR IS REPORTED.
6463                          ;* - THE PROGRAM NEXT REPEATEDLY CHECKS MCLK BIT FOR THE 1 STATE AGAIN, AND
6464                          ;*   IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC, AN ERROR IS REPORTED.
6465                          ;*
6466                          ;* IF THE P-TABLE FOR THIS UNIT INDICATES THAT THE M8207 RUN SWITCH (E28 SW7)
6467                          ;* IS OFF, THE TEST WILL BE SKIPPED.
6468                          ;;*****************************************************************************
6469  023522                  BGNTST
6470  023522                                                      T13::
6471  023522  004737  003554          JSR     PC,MSTCLR       ;PERFORM MASTER CLEAR
6472  023526  012737  000015  002420  MOV     #13.,TSTNUM     ;SET TEST NO.
6473  023534  005737  002454          TST     RUNINH          ;SEE IF RUN SWITCH IS SET
6474  023540  001020                  BNE     1$              ;BR IF YES, TO RUN TEST
```

```
6475  023542  023727  002422  000001          CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
6476  023550  001012                          BNE     40$             ;BR IF NOT, TO SKIP PRINTING
6477  023552                                  PRINTF  #FMT19,TSTNUM   ;PRINT MSG TO SAY TEST NOT RUN
6478  023552  013746  002420                                                  MOV     TSTNUM,-(SP)
6479  023556  012746  011765                                                  MOV     #FMT19,-(SP)
6480  023562  012746  000002                                                  MOV     #2,-(SP)
6481  023566  010600                                                          MOV     SP,R0
6482  023570  104417                                                          TRAP    C$PNTF
6483  023572  062706  000006                                                  ADD     #6,SP
6484  023576  000137  024144          40$:    JMP     A1              ;GO TO SKIP TEST
6485  023602  012737  000017  002356  1$:     MOV     #17,REGNUM      ;SET REG NO. = 17
6486  023610  012701  000360                  MOV     #360,R1         ;SET INSTRUCTION SOURCE = INBUS REG 17
6487  023614  052701  000002                  BIS     #2,R1           ;SET INSTRUCTION DESTINATION = BSEL2
6488  023620  052701  021000                  BIS     #MVIOX,R1       ;SET REST OF MOVE INSTRUCTION
6489  023624  010137  023634                  MOV     R1,2$           ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
6490  023630  004737  004052                  JSR     PC,LOOPIN       ;GET MICROPROCESSOR LOOPING ON MOVE INSTRUCTION
6491  023634  000000                  2$:     .WORD   0               ;INSTRUCTION GOES HERE
6492
6493                                  ;-------------------------------------------------------------------------
6494                                  ; WAIT FOR MCLK BIT TO BE SET TO 1
6495                                  ;-------------------------------------------------------------------------
6496  023636  005037  002466                  CLR     REG0            ;INIT PROGRAM TIMER
6497  023642  117737  156562  002342  3$:     MOVB    @BSEL2,REDBYT   ;GET REG 17 INTO REDBYT
6498  023650  132737  000002  002342          BITB    #MCLK,REDBYT    ;SEE IF MCLK BIT = 1 YET
6499  023656  001031                          BNE     6$              ;BR IF MCLK = 1
6500  023660  005237  002466                  INC     REG0            ;INCREMENT TIMER
6501  023664  001366                          BNE     3$              ;BR IF PROGRAM TIMER DID NOT TIME OUT YET
6502                                                                  ; (TIME OUT = SEVERAL HUNDRED MILLI-SEC)
6503  023666  012737  000002  002362          MOV     #MCLK,GOODAT    ;SET EXPECTED REG CONTENTS
6504  023674  013737  002342  002364          MOV     REDBYT,BADDAT   ;SET ACTUAL REG CONTENTS
6505  023702  004737  003774                  JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
6506                                  ;REPORT MCLK BIT STUCK AT 0
6507  023706                                  ERRDF   5,EM5,ERR2
6508  023706  104455                                                          TRAP    C$ERDF
6509  023710  000005                                                          .WORD   5
6510  023712  012270                                                          .WORD   EM5
6511  023714  014610                                                          .WORD   ERR2
6512
6513  023716                          ;TYPE 'PLEASE INSURE M8207 RUN SWITCH (E28 SW7) IS ON''
6514  023716  012746  012016                  PRINTF  #FMT24
6515  023722  012746  000001                                                  MOV     #FMT24,-(SP)
6516  023726  010600                                                          MOV     #1,-(SP)
6517  023730  104417                                                          MOV     SP,R0
6518  023732  062706  000004                                                  TRAP    C$PNTF
6519  023736  000137  024144                                                  ADD     #4,SP
6520                                          JMP     A1              ;ESCAPE TO END OF TEST
6521
6522                                  ;-------------------------------------------------------------------------
6523                                  ; WAIT FOR MCLK BIT TO BE CLEARED TO 0
6524  023742                          ;-------------------------------------------------------------------------
6525  023742  005037  002466          6$:
6526  023746  117737  156456  002342          CLR     REG0            ;INIT PROGRAM TIMER
6527  023754  132737  000002  002342  8$:     MOVB    @BSEL2,REDBYT   ;GET REG 17 INTO REDBYT
6528  023762  001430                          BITB    #MCLK,REDBYT    ;SEE IF MCLK BIT = 0 YET
6529  023764  005237  002466                  BEQ     10$             ;BR IF MCLK = 0
6530  023770  001366                          INC     REG0            ;INCREMENT TIMER
                                              BNE     8$              ;BR IF TIMER DID NOT TIME OUT YET
```

```
6531  023772  005037  002362                  CLR     GOODAT          ;SET EXPECTED REG CONTENTS
6532  023776  013737  002342  002364          MOV     REDBYT,BADDAT   ;SET ACTUAL REG CONTENTS
6533  024004  004737  003774                  JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
6534                                  ;REPORT MCLK BIT STUCK AT 1
6535  024010                                  ERRDF   6,EM6,ERR2
6536  024010  104455                                                                  TRAP    C$ERDF
6537  024012  000006                                                                  .WORD   6
6538  024014  012321                                                                  .WORD   EM6
6539  024016  014610                                                                  .WORD   ERR2
6540                                  ;TYPE 'PLEASE INSURE M8207 RUN SWITCH (E28 SW7) IS ON'
6541  024020                                  PRINTF  #FMT24
6542  024020  012746  012016                                                          MOV     #FMT24,-(SP)
6543  024024  012746  000001                                                          MOV     #1,-(SP)
6544  024030  010600                                                                  MOV     SP,R0
6545  024032  104417                                                                  TRAP    C$PNTF
6546  024034  062706  000004                                                          ADD     #4,SP
6547  024040  000137  024144                  JMP     A1              ;ESCAPE TO END OF TEST
6548
6549                                  ;-------------------------------------------------------------------------
6550                                  ; WAIT FOR MCLK BIT TO BE SET TO 1 AGAIN
6551                                  ;-------------------------------------------------------------------------
6552  024044                          10$:
6553  024044  005037  002466                  CLR     REG0            ;INIT PROGRAM TIMER
6554  024050  117737  156354  002342  12$:    MOVB    @BSEL2,REDBYT   ;GET REG 17 INTO REDBYT
6555  024056  132737  000002  002342          BITB    #MCLK,REDBYT    ;SEE IF MCLK BIT = 1 YET
6556  024064  001027                          BNE     A1              ;BE IF MCLK = 1
6557  024066  005237  002466                  INC     REG0            ;INCREMENT TIMER
6558  024072  001366                          BNE     12$             ;BR IF TIMER DID NOT TIME OUT YET
6559  024074  012737  000002  002362          MOV     #MCLK,GOODAT    ;SET EXPECTED REG CONTENTS
6560  024102  013737  002342  002364          MOV     REDBYT,BADDAT   ;SET ACTUAL REG CONTENTS
6561  024110  004737  003774                  JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
6562                                  ;REPORT MCLK BIT STUCK AT 0
6563  024114                                  ERRDF   5,EM5,ERR2
6564  024114  104455                                                                  TRAP    C$ERDF
6565  024116  000005                                                                  .WORD   5
6566  024120  012270                                                                  .WORD   EM5
6567  024122  014610                                                                  .WORD   ERR2
6568                                  ;TYPE 'PLEASE INSURE M8207 RUN SWITCH (E28 SW7) IS ON'
6569  024124                                  PRINTF  #FMT24
6570  024124  012746  012016                                                          MOV     #FMT24,-(SP)
6571  024130  012746  000001                                                          MOV     #1,-(SP)
6572  024134  010600                                                                  MOV     SP,R0
6573  024136  104417                                                                  TRAP    C$PNTF
6574  024140  062706  000004                                                          ADD     #4,SP
6575  024144                          A1:
6576  024144  004737  003554                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
6577  024150                          ENDTST
6578  024150                                                                  L10036:
6579  024150  104401                                                                  TRAP    C$ETST
6580
6581
6582
6583
6584
6585                                  ;;*************************************************************************
6586                                  .SBTTL      TEST 14 - EXTENDED REGISTER MASTER CLEAR TEST
```

```
                                        ;*
                                        ;* FIRST, ALL READ/WRITE BITS OF EXTENDED REGS AX0-AX3 ARE SET BY LOADING
                                        ;* A DIFFERENT WORD OF PATTERN H INTO EACH REG. THEN, A MASTER CLEAR IS
                                        ;* ISSUED AND EACH REG IS READ AND COMPARED TO A WORD OF PATTERN I, WHICH
                                        ;* CONTAINS THE INITIALIZED STATES OF ALL THE EXTENDED REGS.
                                        ;*    PATTERN H = 000,000,377,017,377,377,375,377
                                        ;*    PATTERN I = 000,000,000,000,000,103,000,000
                                        ;;*************************************************************************
024152                                  BGNTST
024152                                                                          T14::
024152  004737  003554                         JSR     PC,MSTCLR       ;ISSUE AN INITIAL MASTER CLEAR
024156  005037  002360                         CLR     AXNUM           ;INIT AX REG BYTE NO. TO 0
024162  012701  002632                         MOV     #PATH,R1        ;INIT DATA PATTERN POINTER
024166  112137  002352                 1$:     MOVB    (R1)+,WAX15     ;SET BITS TO LOAD INTO LO BYTE
024172  112137  002354                         MOVB    (R1)+,WAX16     ;SET BITS TO LOAD INTO HI BYTE
024176  004737  004270                         JSR     PC,WRITAX       ;WRITE EXTENDED REG
024202  032737  000002  002376               BIT     #WRDYTO,ERROR1  ;SEE IF READY FAILED TO SET
024210  001413                               BEQ     8$              ;BR IF NOT
024212  012737  000014  002356               MOV     #14,REGNUM      ;SET LU REG NO = 14
024220  004737  003774                       JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
                                        ;REPORT READY NOT SET AFTER AX REG WRITE
                                                ERRDF   52,EM52,ERR9
024224
024224  104455                                                                TRAP    C$ERDF
024226  000064                                                                .WORD   52
024230  014063                                                                .WORD   EM52
024232  020606                                                                .WORD   ERR9
024234                                          ESCAPE  TST
024234  104410                                                                TRAP    C$ESCAPE
024236  000244                                                                .WORD   L10037-.
024240  062737  000002  002360         8$:     ADD     #2,AXNUM        ;INCR REG BYTE NO.
024246  020127  002642                         CMP     R1,#PATI        ;SEE IF ALL REGS WRITTEN YET
024252  103745                               BLO     1$              ;BR IF NOT DONE WRITING YET
024254  004737  003554                       JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
024260  005037  002360                       CLR     AXNUM           ;INIT EXTENDED REG BYTE NO. TO 0
024264  012701  002642                       MOV     #PATI,R1        ;INIT DATA PAT POINTER FOR READS
024270  004737  004102                 2$:     JSR     PC,READAX       ;READ AN EXTENDED REG
024274  032737  000001  002376               BIT     #RRDYTO,ERROR1  ;SEE IF READY FAILED TO SET
024302  001413                               BEQ     10$             ;BR IF NOT
024304  012737  000014  002356               MOV     #14,REGNUM      ;SET LU REG NO. = 14
024312  004737  003774                       JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
                                        ;REPORT READY NOT SET AFTER AX REG READ
                                                ERRDF   53,EM53,ERR9
024316
024316  104455                                                                TRAP    C$ERDF
024320  000065                                                                .WORD   53
024322  014124                                                                .WORD   EM53
024324  020606                                                                .WORD   ERR9
024326                                          ESCAPE  TST
024326  104410                                                                TRAP    C$ESCAPE
024330  000152                                                                .WORD   L10037-.
024332  023727  002360  000006         10$:    CMP     AXNUM,#6        ;SEE IF AX3-15
024340  001003                               BNE     3$              ;BR IF NOT
024342  142737  000372  002346               BICB    #AX315U,RAX15   ;MASK OFF UNPREDICTABLE BITS
024350  123711  002346                 3$:     CMPB    RAX15,(R1)      ;COMPARE LO BYTE TO EXPECTED VALUE
024354  001417                               BEQ     4$              ;BR IF DATA MATCHES
024356  005037  002362                       CLR     GOODAT
024362  111137  002362                       MOVB    (R1),GOODAT     ;GET EXPECTED DATA BYTE
```

```
6643  024366  013737  002346  002364          MOV     RAX15,BADDAT    ;GET ACTUAL DATA BYTE
6644  024374  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
6645                                  ;REPORT REG NOT INITIALIZED BY MASTER CLEAR
6646  024400                                  ERRDF   2,EM2,ERR3
6647  024400  104455                                                          TRAP    C$ERDF
6648  024402  000002                                                          .WORD   2
6649  024404  012137                                                          .WORD   EM2
6650  024406  015116                                                          .WORD   ERR3
6651  024410                                  ESCAPE  TST
6652  024410  104410                                                          TRAP    C$ESCAPE
6653  024412  000070                                                          .WORD   L10037-.
6654  024414  005237  002360          4$:     INC     AXNUM           ;INCREMENT AX BYTE NO.
6655  024420  005201                          INC     R1              ;INCREMENT PAT POINTER
6656  024422  123711  002350                  CMPB    RAX16,(R1)      ;COMPARE HI BYTE TO EXPECTED VALUE
6657  024426  001417                          BEQ     6$              ;BR IF DATA MATCHES
6658  024430  005037  002362                  CLR     GOODAT
6659  024434  111137  002362                  MOVB    (R1),GOODAT     ;GET EXPECTED DATA BYTE
6660  024440  013737  002350  002364          MOV     RAX16,BADDAT    ;GET ACTUAL DATA BYTE
6661  024446  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
6662                                  ;REPORT REG NOT INITIALIZED BY MASTER CLEAR
6663  024452                                  ERRDF   2,EM2,ERR3
6664  024452  104455                                                          TRAP    C$ERDF
6665  024454  000002                                                          .WORD   2
6666  024456  012137                                                          .WORD   EM2
6667  024460  015116                                                          .WORD   ERR3
6668  024462                                  ESCAPE  TST
6669  024462  104410                                                          TRAP    C$ESCAPE
6670  024464  000016                                                          .WORD   L10037-.
6671  024466  005237  002360          6$:     INC     AXNUM           ;INCR AX BYTE NO.
6672  024472  005201                          INC     R1              ;INCR PAT POINTER
6673  024474  020127  002652                  CMP     R1,#PATJ        ;SEE IF ALL REGS READ YET
6674  024500  103673                          BLO     2$              ;BR IF NOT DONE READING YET
6675  024502                          ENDTST
6676  024502                                                          L10037:
6677  024502  104401                                                          TRAP    C$ETST
6678
6679
6680
6681
6682
6683                                  ;;*************************************************************************
6684                                  .SBTTL      TEST 15 - EXTENDED REGISTER ADDRESSING TEST
6685                                  ;*
6686                                  ;* FIRST, ISSUE A MASTER CLEAR TO PUT REGS INTO INITIALIZED STATES SHOWN IN
6687                                  ;* PATTERN I. THEN, WRITE A DIFFERENT WORD OF PATTERN J INTO EACH EXTENDED (AX)
6688                                  ;* REG, AND AFTER EACH WRITE, READ AND COMPARE ALL EXTENDED REGS TO EXPECTED
6689                                  ;* VALUES.
6690                                  ;*   PATTERN I = 000,000,000,000,000,103,000,000
6691                                  ;*   PATTERN J = 000,000,010,002,004,103,001,100
6692                                  ;;*************************************************************************
6693  024504                          BGNTST
6694  024504                                                          T15::
6695  024504  004737  003554                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
6696  024510  012701  002642                  MOV     #PATI,R1        ;INIT POINTER TO PAT I
6697  024514  012702  002456                  MOV     #REDDAT,R2      ;INIT POINTER TO EXPECTED DATA AREA
6698  024520  112122                  3$:     MOVB    (R1)+,(R2)+     ;MOVE PAT I INTO REDDAT TABLE
```

```
6699  024522  020127  002652                  CMP     R1,#PATJ
6700  024526  103774                          BLO     3$
6701  024530  005001                          CLR     R1                  ;INIT INDEX FOR WRITING
6702  024532  010137  002360          6$:     MOV     R1,AXNUM            ;GET AX BYTE NO. FOR WRITING
6703  024536  116137  002652  002352          MOVB    PATJ(R1),WAX15      ;SET LO AX BYTE TO BE WRITTEN
6704  024544  116137  002653  002354          MOVB    PATJ+1(R1),WAX16    ;SET HI AX BYTE TO BE WRITTEN
6705  024552  113761  002352  002456          MOVB    WAX15,REDDAT(R1)    ;SET EXPECTED LO BYTE IN TABLE
6706  024560  113761  002354  002457          MOVB    WAX16,REDDAT+1(R1)  ;SET EXPECTED HI BYTE IN TABLE
6707  024566  004737  004270                  JSR     PC,WRITAX           ;WRITE DATA BYTES INTO EXTENDED REG
6708  024572  005003                          CLR     R3                  ;INIT INDEX FOR READING
6709  024574  010337  002360          9$:     MOV     R3,AXNUM            ;GET AX BYTE NO. FOR READING
6710  024600  004737  004102                  JSR     PC,READAX           ;READ 2 AX BYTES
6711  024604  023727  002360  000006          CMP     AXNUM,#6            ;SEE IF AX3-15
6712  024612  001003                          BNE     10$                 ;BR IF NOT
6713  024614  142737  000372  002346          BICB    #AX315U,RAX15       ;MASK OFF UNPREDICTABLE BITS
6714  024622  123763  002346  002456  10$:    CMPB    RAX15,REDDAT(R3)    ;COMPARE LO BYTE READ TO EXPECTED
6715  024630  001420                          BEQ     12$                 ;BR IF LO BYTE MATCHES EXPECTED
6716  024632  005037  002362                  CLR     GOODAT
6717  024636  116337  002456  002362          MOVB    REDDAT(R3),GOODAT   ;GET EXPECTED LO BYTE
6718  024644  013737  002346  002364          MOV     RAX15,BADDAT        ;GET ACTUAL DATA
6719  024652  004737  004504                  JSR     PC,GETALL           ;GET REGS FOR PRINTOUT
6720                                  ;REPORT REG MISCOMPARE
6721  024656                                  ERRDF   3,EM3,ERR3
6722  024656  104455                                                      TRAP    C$ERDF
6723  024660  000003                                                      .WORD   3
6724  024662  012176                                                      .WORD   EM3
6725  024664  015116                                                      .WORD   ERR3
6726  024666                                  ESCAPF  IST
6727  024666  104410                                                      TRAP    C$ESCAPE
6728  024670  000102                                                      .WORD   L10040-.
6729  024672  005237  002360          2$:     INC     AXNUM               ;INCR AX BYTE NO.
6730  024676  123763  002350  002457          CMPB    RAX16,REDDAT+1(R3)  ;COMPARE HI BYTE READ TO EXPECTED
6731  024704  001420                          BEQ     15$                 ;BR IF HI BYTE MATCHES EXPECTED
6732  024706  005037  002362                  CLR     GOODAT
6733  024712  116337  002457  002362          MOVB    REDDAT+1(R3),GOODAT ;GET EXPECTED HI BYTE
6734  024720  013737  002350  002364          MOV     RAX16,BADDAT        ;GET ACTUAL DATA
6735  024726  004737  004504                  JSR     PC,GETALL           ;GET REGS FOR PRINTOUT
6736                                  ;REPORT REG MISCOMPARE
6737  024732                                  ERRDF   3,EM3,ERR3
6738  024732  104455                                                      TRAP    C$ERDF
6739  024734  000003                                                      .WORD   3
6740  024736  012176                                                      .WORD   EM3
6741  024740  015116                                                      .WORD   ERR3
6742  024742                                  ESCAPE  TST
6743  024742  104410                                                      TRAP    C$ESCAPE
6744  024744  000026                                                      .WORD   L10040-.
6745  024746  062703  000002          15$:    ADD     #2,R3               ;INCR INDEX FOR READS
6746  024752  020327  000010                  CMP     R3,#10              ;SEE IF ALL AX BYTES READ YET
6747  024756  002706                          BLT     9$                  ;BR IF NOT DONE READING YET
6748  024760  062701  000002                  ADD     #2,R1               ;INCR INDEX FOR WRITES
6749  024764  020127  000010                  CMP     R1,#10              ;SEE IF ALL AX BYTES WRITTEN YET
6750  024770  002660                          BLT     6$                  ;BR IF NOT DONE WRITING YET
6751  024772                          ENDTST
6752  024772                                                      L10040:
6753  024772  104401                                                      TRAP    C$ETST
6754
```

```
;;*************************************************************************
.SBTTL      TEST 16 - REGS 15,16 / AX2-15,AX2-16 READ/WRITE BIT TEST
;*
;* USING REGS 15,16, THE INDIRECT REGS AX2-15,AX2-16 (USYRT REGS 4,5) ARE
;* WRITTEN AND READ USING EACH WORD OF PATTERN K. AX2-15 IS COMPARED
;* TO THE WORD WRITTEN, AND AX2-16 IS ALWAYS COMPARED TO 103. (AX2-16 IS NOT
;* WRITEABLE).
;*    PATTERN K =
;*      FOR REG 15: 000,377,125,252,001,002,004,010,020,040,100,200,000,000,
;*                  000,000,000,000,000,000,376,375,373,367,357,337,277,177,
;*                  377,377,377,377,377,377,377,377
;*      FOR REG 16: 000,377,125,252,000,000,000,000,000,000,000,000,001,002,
;*                  004,010,020,040,100,200,377,377,377,377,377,377,377,377,
;*                  376,375,373,367,357,337,277,177.
;;*************************************************************************
BGNTST
                                                                    T16::
        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
        MOV     #PATK,R1        ;INIT DATA PATTERN POINTER
3$:
        BGNSEG
                                                                    TRAP    C$BSEG
        MOV     #4,AXNUM        ;SET BYTE NO. = 4
        MOVB    (R1),WAX15      ;SET DATA TO WRITE INTO LO BYTE
        MOVB    1(R1),WAX16     ;SET DATA TO WRITE INTO HI BYTE
        BICB    ANBITS+4,WAX15  ;MASK OFF NON-READ/WRITE BITS IN LO BYTE
        BICB    ANBITS+5,WAX16  ;MASK OFF NON-READ/WRITE BITS IN HI BYTE
        JSR     PC,WRITAX       ;LOAD DATA INTO AX2-15,AX2-16
        JSR     PC,READAX       ;READ AX2-15 AND AX2-16
        CMPB    RAX15,WAX15     ;COMPARE LO BYTE DATA READ
        BEQ     6$              ;BR IF DATA MATCHES
        MOV     WAX15,GOODAT    ;SET EXPECTED DATA
        MOV     RAX15,BADDAT    ;SET ACTUAL DATA
        JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
;REPORT REG MISCOMPARE
        ERRDF   3,EM3,ERR3
                                                                    TRAP    C$ERDF
                                                                    .WORD   3
                                                                    .WORD   EM3
                                                                    .WORD   ERR3
        ESCAPE  SEG
                                                                    TRAP    C$ESCAPE
                                                                    .WORD   10000$-.
6$:     INC     AXNUM           ;SET AX BYTE NO. = 5
        CMPB    RAX16,PATI+5    ;COMPARE HI BYTE DATA READ
        BEQ     9$              ;BR IF DATA MATCHES
        CLR     GOODAT
        MOVB    PATI+5,GOODAT   ;SET EXPECTED DATA
        MOV     RAX16,BADDAT    ;SET ACTUAL DATA
        JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
;REPORT REG MISCOMPARE
        ERRDF   3,EM3,ERR3
```

```
6811  025156  104455                                                                  TRAP    C$ERDF
6812  025160  000003                                                                  .WORD   3
6813  025162  012176                                                                  .WORD   EM3
6814  025164  015116                                                                  .WORD   ERR3
6815  025166                  9$:
6816  025166                          ENDSEG
6817  025166                                                                  10000$:
6818  025166  104405                                                                  TRAP    C$ESEG
6819  025170  062701  000002          ADD     #2,R1           ;INCR PATTERN POINTER
6820  025174  020127  002772          CMP     R1,#PATL        ;SEE IF ALL DATA WRITTEN YET
6821  025200  103701                  BLO     3$              ;BR IF NOT DONE YET
6822  025202                  ENDTST
6823  025202                                                                  L10041:
6824  025202  104401                                                                  TRAP    C$ETST
6825
6826
6827
6828
6829
6830                            ;;*****************************************************************************
6831                            .SBTTL      TEST 17 - AX0-15,AX0-16 READ/WRITE BIT TEST
6832                            ;*
6833                            ;* IN THIS TEST, A MASTER CLEAR IS DONE, AND THEN A WRITE, READ, AND COMPARE
6834                            ;* ARE PERFORMED IN REGS AX0-15,AX0-16 USING EACH WORD OF PATTERN L.
6835                            ;* ANY BITS IN AX0-15,AX0-16 WHICH ARE NOT READ/WRITE ARE MASKED OFF (TO 0)
6836                            ;* IN THE EXPECTED VALUE BEFORE COMPARISON.
6837                            ;*    PATTERN L =
6838                            ;*      FOR REG 15: 000,377,000
6839                            ;*      FOR REG 16: 000,377,000.
6840                            ;;*****************************************************************************
6841  025204                  BGNTST
6842  025204                                                                  T17::
6843  025204  004737  003554          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
6844  025210  012701  002772          MOV     #PATL,R1        ;INIT DATA PATTERN POINTER
6845  025214                  3$:
6846  025214                          BGNSEG
6847  025214  104404                                                                  TRAP    C$BSEG
6848  025216  012737  000000  002360  MOV     #0,AXNUM        ;SET BYTE NO. = 0
6849  025224  111137  002352          MOVB    (R1),WAX15      ;SET DATA TO WRITE INTO LO BYTE
6850  025230  116137  000001  002354  MOVB    1(R1),WAX16     ;SET DATA TO WRITE INTO HI BYTE
6851  025236  143737  002537  002352  BICB    ANBITS+0,WAX15  ;MASK OFF NON-READ/WRITE BITS IN LO BYTE
6852  025244  143737  002540  002354  BICB    ANBITS+1,WAX16  ;MASK OFF NON-READ/WRITE BITS IN HI BYTE
6853  025252  004737  004270          JSR     PC,WRITAX       ;LOAD DATA INTO AX0-15,AX0-16
6854  025256  004737  004102          JSR     PC,READAX       ;READ AX0-15 AND AX0-16
6855  025262  123737  002346  002352  CMPB    RAX15,WAX15     ;COMPARE LO BYTE DATA READ
6856  025270  001416                  BEQ     6$              ;BR IF DATA MATCHES
6857  025272  013737  002352  002362  MOV     WAX15,GOODAT    ;SET EXPECTED DATA
6858  025300  013737  002346  002364  MOV     RAX15,BADDAT    ;SET ACTUAL DATA
6859  025306  004737  004504          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
6860                  ;REPORT REG MISCOMPARE
6861  025312                          ERRDF   3,EM3,ERR3
6862  025312  104455                                                                  TRAP    C$ERDF
6863  025314  000003                                                                  .WORD   3
6864  025316  012176                                                                  .WORD   EM3
6865  025320  015116                                                                  .WORD   ERR3
6866  025322                          ESCAPE  SEG
```

```
 6867  025322  104410                                                                  TRAP    C$ESCAPE
 6868  025324  000046                                                                  .WORD   10000$-.
 6869  025326  005237  002360          6$:     INC     AXNUM           ;SET AX BYTE NO. = 1
 6870  025332  123737  002350  002354          CMPB    RAX16,WAX16     ;COMPARE HI BYTE DATA READ
 6871  025340  001414                          BEQ     9$              ;BR IF DATA MATCHES
 6872  025342  013737  002354  002362          MOV     WAX16,GOODAT    ;SET EXPECTED DATA
 6873  025350  013737  002350  002364          MOV     RAX16,BADDAT    ;SET ACTUAL DATA
 6874  025356  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 6875                                  ;REPORT REG MISCOMPARE
 6876  025362                                  ERRDF   3,EM3,ERR3
 6877  025362  104455                                                                  TRAP    C$ERDF
 6878  025364  000003                                                                  .WORD   3
 6879  025366  012176                                                                  .WORD   EM3
 6880  025370  015116                                                                  .WORD   ERR3
 6881  025372                          9$:
 6882  025372                                  ENDSEG
 6883  025372                                                                  10000$:
 6884  025372  104405                                                                  TRAP    C$ESEG
 6885  025374  062701  000002                  ADD     #2,R1           ;INCR PATTERN POINTER
 6886  025400  020127  003000                  CMP     R1,#PATM        ;SEE IF ALL DATA WRITTEN YET
 6887  025404  103703                          BLO     3$              ;BR IF NOT DONE YET
 6888  025406                          ENDTST
 6889  025406                                                                  L10042:
 6890  025406  104401                                                                  TRAP    C$ETST
 6891
 6892
 6893
 6894
 6895
 6896                                  ;;*****************************************************************************
 6897                                  .SBTTL      TEST 18 - AX1-15,AX1-16 READ/WRITE BIT TEST
 6898                                  ;*
 6899                                  ;* IN THIS TEST, A MASTER CLEAR IS DONE, AND THEN A WRITE, READ, AND COMPARE
 6900                                  ;* ARE PERFORMED IN REGS AX1-15,AX1-16 USING EACH WORD OF PATTERN K.
 6901                                  ;* ANY BITS IN AX1-15,AX1-16 WHICH ARE NOT READ/WRITE ARE MASKED OFF (TO 0)
 6902                                  ;* IN THE EXPECTED VALUE BEFORE COMPARISON.
 6903                                  ;;*****************************************************************************
 6904  025410                          BGNTST
 6905  025410                                                                  T18::
 6906  025410  004737  003554                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 6907  025414  012701  002662                  MOV     #PATK,R1        ;INIT DATA PATTERN POINTER
 6908  025420                          3$:
 6909  025420                                  BGNSEG
 6910  025420  104404                                                                  TRAP    C$BSEG
 6911  025422  012737  000002  002360          MOV     #2,AXNUM        ;SET BYTE NO. = 2
 6912  025430  111137  002352                  MOVB    (R1),WAX15      ;SET DATA TO WRITE INTO LO BYTE
 6913  025434  116137  000001  002354          MOVB    1(R1),WAX16     ;SET DATA TO WRITE INTO HI BYTE
 6914  025442  143737  002541  002352          BICB    ANBITS+2,WAX15  ;MASK OFF NON-READ/WRITE BITS IN LO BYTE
 6915  025450  143737  002542  002354          BICB    ANBITS+3,WAX16  ;MASK OFF NON-READ/WRITE BITS IN HI BYTE
 6916  025456  004737  004270                  JSR     PC,WRITAX       ;LOAD DATA INTO AX1-15,AX1-16
 6917  025462  004737  004102                  JSR     PC,READAX       ;READ AX1-15 AND AX1-16
 6918  025466  123737  002346  002352          CMPB    RAX15,WAX15     ;COMPARE LO BYTE DATA READ
 6919  025474  001416                          BEQ     6$              ;BR IF DATA MATCHES
 6920  025476  013737  002352  002362          MOV     WAX15,GOODAT    ;SET EXPECTED DATA
 6921  025504  013737  002346  002364          MOV     RAX15,BADDAT    ;SET ACTUAL DATA
 6922  025512  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
```

```
6923                                    ;REPORT REG MISCOMPARE
6924  025516                                    ERRDF   3,EM3,ERR3
6925  025516  104455                                                                  TRAP    C$ERDF
6926  025520  000003                                                                  .WORD   3
6927  025522  012176                                                                  .WORD   EM3
6928  025524  015116                                                                  .WORD   ERR3
6929  025526                                    ESCAPE  SEG
6930  025526  104410                                                                  TRAP    C$ESCAPE
6931  025530  000046                                                                  .WORD   10000$-.
6932  025532  005237  002360          6$:       INC     AXNUM           ;SET AX BYTE NO. = 3
6933  025536  123737  002350  002354          CMPB    RAX16,WAX16     ;COMPARE HI BYTE DATA READ
6934  025544  001414                          BEQ     9$              ;BR IF DATA MATCHES
6935  025546  013737  002354  002362          MOV     WAX16,GOODAT    ;SET EXPECTED DATA
6936  025554  013737  002350  002364          MOV     RAX16,BADDAT    ;SET ACTUAL DATA
6937  025562  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
6938                                    ;REPORT REG MISCOMPARE
6939  025566                                    ERRDF   3,EM3,ERR3
6940  025566  104455                                                                  TRAP    C$ERDF
6941  025570  000003                                                                  .WORD   3
6942  025572  012176                                                                  .WORD   EM3
6943  025574  015116                                                                  .WORD   ERR3
6944  025576                            9$:
6945  025576                                    ENDSEG
6946  025576                                                                10000$:
6947  025576  104405                                                                  TRAP    C$ESEG
6948  025600  062701  000002                    ADD     #2,R1           ;INCR PATTERN POINTER
6949  025604  020127  002772                    CMP     R1,#PATL        ;SEE IF ALL DATA WRITTEN YET
6950  025610  103703                          BLO     3$              ;BR IF NOT DONE YET
6951  025612                            ENDTST
6952  025612                                                                L10043:
6953  025612  104401                                                                  TRAP    C$ETST
6954
6955
6956
6957
6958
6959                                    ;;*****************************************************************************
6960                                    .SBTTL      TEST 19 - AX3-15,AX3-16 READ/WRITE BIT TEST
6961                                    ;*
6962                                    ;* IN THIS TEST A MASTER CLEAR IS DONE AND THEN A WRITE, READ, AND COMPARE ARE
6963                                    ;* PERFORMED IN REGS AX3-15,AX3-16 USING EACH WORD OF PATTERN V FOR WRITING,
6964                                    ;* AND PATTERN U FOR COMPARING.
6965                                    ;* ANY BITS IN AX3-15,AX3-16 WHICH ARE NOT READ/WRITE ARE MASKED OFF (TO 0)
6966                                    ;* IN THE EXPECTED VALUE BEFORE COMPARISON.
6967                                    ;*   PATTERN V =
6968                                    ;*      FOR REG 15 : 000,333,331,323,313,233,133,000,000,000,000,
6969                                    ;*                   000,000,000,000,000,000,000,000
6970                                    ;*      FOR REG 16 : 000,000,000,000,000,000,000,001,002,004,040,
6971                                    ;*                   100,200,346,345,343,307,247,147
6972                                    ;*   PATTERN U =
6973                                    ;*      FOR REG 15 : 000,001,013,011,021,101,301,000,000,000,000,
6974                                    ;*                   000,000,000,000,000,000,000,000
6975                                    ;*      FOR REG 16 : 000,000,000,000,000,000,000,001,002,004,040,
6976                                    ;*                   100,200,346,345,343,307,247,147
6977                                    ;;*****************************************************************************
6978  025614                            BGNTST
```

```
 6979  025614                                                                                  T19::
 6980  025614  004737  003554                JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 6981  025620  142777  000010  154600        BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
 6982  025626  012702  003130                MOV     #PATV,R2        ;INIT PATTERN V POINTER
 6983  025632  012701  003062                MOV     #PATU,R1        ;INIT PATTERN U POINTER
 6984  025636                          3$:
 6985  025636                                BGNSEG
 6986  025636  104404                                                                 TRAP    C$BSEG
 6987  025640  012737  000006  002360        MOV     #6,AXNUM        ;SET BYTE NO. = 6
 6988  025646  111237  002352                MOVB    (R2),WAX15      ;SET DATA TO WRITE INTO LO BYTE
 6989  025652  116237  000001  002354        MOVB    1(R2),WAX16     ;SET DATA TO WRITE INTO HI BYTE
 6990  025660  004737  004270                JSR     PC,WRITAX       ;LOAD DATA INTO AX3-15,AX3-16
 6991  025664  004737  004102                JSR     PC,READAX       ;READ AX3-15 AND AX3-16
 6992  025670  132737  000001  002352        BITB    #TEST,WAX15     ;SEE IF AN INTERFACE IS SELECTED
 6993  025676  001003                        BNE     4$              ;BR IF YES
 6994  025700  142737  000372  002346        BICB    #AX315U,RAX15   ;MASK OFF UNPREDICTABLE BITS
 6995  025706  142737  000040  002346  4$:   BICB    #C32BCC,RAX15   ;CLEAR CRC32 BCC BIT
 6996  025714  123711  002346                CMPB    RAX15,(R1)      ;COMPARE LO BYTE DATA READ
 6997  025720  001417                        BEQ     6$              ;BR IF DATA MATCHES
 6998  025722  005037  002362                CLR     GOODAT
 6999  025726  111137  002362                MOVB    (R1),GOODAT     ;SET EXPECTED DATA
 7000  025732  013737  002346  002364        MOV     RAX15,BADDAT    ;SET ACTUAL DATA
 7001  025740  004737  004504                JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 7002                          ;REPORT REG MISCOMPARE
 7003  025744                                ERRDF   3,EM3,ERR3
 7004  025744  104455                                                                 TRAP    C$ERDF
 7005  025746  000003                                                                 .WORD   3
 7006  025750  012176                                                                 .WORD   EM3
 7007  025752  015116                                                                 .WORD   ERR3
 7008  025754                                ESCAPE  SEG
 7009  025754  104410                                                                 TRAP    C$ESCAPE
 7010  025756  000052                                                                 .WORD   10000$-.
 7011  025760  005237  002360        6$:     INC     AXNUM           ;SET AX BYTE NO. = 7
 7012  025764  123761  002350  000001        CMPB    RAX16,1(R1)     ;COMPARE HI BYTE DATA READ
 7013  025772  001416                        BEQ     9$              ;BR IF DATA MATCHES
 7014  025774  005037  002362                CLR     GOODAT
 7015  026000  116137  000001  002362        MOVB    1(R1),GOODAT    ;SET EXPECTED DATA
 7016  026006  013737  002350  002364        MOV     RAX16,BADDAT    ;SET ACTUAL DATA
 7017  026014  004737  004504                JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 7018                          ;REPORT REG MISCOMPARE
 7019  026020                                ERRDF   3,EM3,ERR3
 7020  026020  104455                                                                 TRAP    C$ERDF
 7021  026022  000003                                                                 .WORD   3
 7022  026024  012176                                                                 .WORD   EM3
 7023  026026  015116                                                                 .WORD   ERR3
 7024  026030                          9$:
 7025  026030                                ENDSEG
 7026  026030                                                                  10000$:
 7027  026030  104405                                                                 TRAP    C$ESEG
 7028  026032  062702  000002                ADD     #2,R2           ;INCR PATTERN V POINTER
 7029  026036  062701  000002                ADD     #2,R1           ;INCR PATTERN U POINTER
 7030  026042  020127  003130                CMP     R1,#PATV        ;SEE IF ALL DATA WRITTEN YET
 7031  026046  103673                        BLO     3$              ;BR IF NOT DONE YET
 7032  026050                          ENDTST
 7033  026050                                                                  L10044:
 7034  026050  104401                                                                 TRAP    C$ETST
```

```
                                ;;*************************************************************************
                                .SBTTL      TEST 20 - REG 17 - AX2-16 READ/WRITE, MASTER CLEAR TEST
                                ;*
                                ;* THIS TEST CONSISTS OF 2 SUBTESTS. IN THE FIRST SUBTEST, EACH BYTE OF PAT O
                                ;* IS WRITTEN INTO REG 17 AND AFTER EACH WRITE, AX2-16 IS READ AND COMPARED
                                ;* TO A BYTE OF PAT P.
                                ;*   PATTERN O = 000,041,004,010,020,040,100,101,200,201,300,111,301,375
                                ;*   PATTERN P = 000,113,200,040,020,010,001,104,007,105,007,144,107,157
                                ;* IN THE SECOND SUBTEST, REG 17 IS LOADED WITH 375, A MASTER CLEAR IS ISSUED,
                                ;* AND AX2-16 IS COMPARED TO ITS INITIALIZED STATE (103).
                                ;;*************************************************************************
026052                          BGNTST
                                                                                        T20::
026052  004737  003554                 JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
026056  012701  003026                 MOV     #PATO,R1        ;INIT PAT O POINTER
026062  012702  003044                 MOV     #PATP,R2        ;INIT PAT P POINTER
026066  012737  000017  002356         MOV     #17,REGNUM      ;SET LU REG NO. = 17
026074  012737  000005  002360         MOV     #5,AXNUM        ;SET AX BYTE NO. = 5 FOR AX2-16
                                ;-------------------------------------------------------------------------
                                ; WRITE REG 17, READ AND COMPARE AX2-16
                                ;-------------------------------------------------------------------------
026102                                  BGNSUB
                                                                                        T20.1:
026102  104402                                                                                TRAP    C$BSUB
026104                          3$:
026104                                  BGNSEG
026104  104404                                                                                TRAP    C$BSEG
026106  111137  002344                 MOVB    (R1),WRIBYT     ;SET BYTE TO BE WRITTEN
026112  004737  003726                 JSR     PC,WRITLU       ;WRITE DATA BYTE INTO REG 17
026116  004737  004102                 JSR     PC,READAX       ;READ AX2
026122  123712  002350                 CMPB    RAX16,(R2)      ;COMPARE AX2-16 TO EXPECTED DATA
026126  001421                         BEQ     6$              ;BR IF DATA MATCHES
026130  005037  002366                 CLR     LOADAT
026134  111137  002366                 MOVB    (R1),LOADAT     ;SET DATA WHICH WAS WRITTEN
026140  005037  002362                 CLR     GOODAT
026144  111237  002362                 MOVB    (R2),GOODAT     ;SET EXPECTED DATA READ
026150  013737  002350  002364         MOV     RAX16,BADDAT    ;SET ACTUAL DATA READ
026156  004737  004504                 JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                ;REPORT REG MISCOMPARE
026162                                  ERRDF   3,EM3,ERR5
026162  104455                                                                              TRAP    C$ERDF
026164  000003                                                                              .WORD   3
026166  012176                                                                              .WORD   EM3
026170  016256                                                                              .WORD   ERR5
026172                          6$:
026172                                  ENDSEG
026172                                                                              10000$:
026172  104405                                                                              TRAP    C$ESEG
026174  005201                         INC     R1              ;INCR PAT O POINTER
026176  005202                         INC     R2              ;INCR PAT P POINTER
026200  020127  003044                 CMP     R1,#PATP        ;SEE IF ALL BYTES LOADED YET
```

```
7091  026204  103737                        BLO     3$              ;BR IF NOT
7092  026206                                ENDSUB
7093  026206                                                                L10046:
7094  026206  104403                                                               TRAP    C$ESUB
7095                          ;-------------------------------------------------------------------
7096                          ; LOAD REG 17, DO MASTER CLEAR, READ AND COMPARE AX2-16
7097                          ;-------------------------------------------------------------------
7098  026210                  BGNSUB
7099  026210                                                                T20.2:
7100  026210  104402                                                               TRAP    C$BSUB
7101  026212  112737  000375  002344          MOVB    #375,WRIBYT     ;SET DATA TO BE LOADED
7102  026220  004737  003726                  JSR     PC,WRITLU       ;LOAD DATA INTO REG 17
7103  026224  004737  003554                  JSR     PC,MSTCLR       ;PERFORM MASTER CLEAR
7104  026230  004737  004102                  JSR     PC,READAX       ;READ AX2-15,AX2-16
7105  026234  123737  002350  002647          CMPB    RAX16,PATI+5    ;SEE IF AX2-16 WAS INIT'D CORRECTLY
7106  026242  001416                          BEQ     6$              ;BR IF YES
7107  026244  005037  002362                  CLR     GOODAT
7108  026250  113737  002647  002362          MOVB    PATI+5,GOODAT   ;SET EXPECTED DATA
7109  026256  013737  002350  002364          MOV     RAX16,BADDAT    ;SET ACTUAL DATA
7110  026264  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7111                  ;REPORT REG NOT INITIALIZED BY MASTER CLEAR
7112  026270                                  ERRDF   2,EM2,ERR3
7113  026270  104455                                                               TRAP    C$ERDF
7114  026272  000002                                                               .WORD   2
7115  026274  012137                                                               .WORD   EM2
7116  026276  015116                                                               .WORD   ERR3
7117  026300                  6$:
7118  026300                          ENDSUB
7119  026300                                                                L10047:
7120  026300  104403                                                               TRAP    C$ESUB
7121  026302                  ENDTST
7122  026302                                                                L10045:
7123  026302  104401                                                               TRAP    C$ETST
7124
7125
7126
7127
7128
7129                          ;;*****************************************************************************
7130                          .SBTTL      TEST 21 - TRANSMITTER BUFFER DATA TEST
7131                          ;* A MASTER CLEAR IS DONE FIRST, AND THEN A BYTE OF PATTERN N IS LOADED INTO
7132                          ;* REG 11 AND THE NEXT BYTE IS LOADED TWICE INTO REG 10. THE PROGRAM THEN WAITS
7133                          ;* AT LEAST 50 MICRO-SEC, AND THEN IT READS AND COMPARES AX1-15 TO THE BYTE
7134                          ;* WHICH WAS LOADED INTO REG 10, AND IT READS AND COMPARES AX1-16 TO THE BYTE
7135                          ;* WHICH WAS LOADED INTO REG 11. THIS PROCESS IS REPEATED (INCLUDING THE MASTER
7136                          ;* CLEAR) FOR EACH PAIR OF BYTES IN PATTERN N.
7137                          ;*   PATTERN N =
7138                          ;*     FOR REG 10: 000,125,252,377,000,000,000
7139                          ;*     FOR REG 11: 000,000,000,000,005,012,017
7140                          ;;*****************************************************************************
7141  026304                  BGNTST
7142  026304                                                                T21::
7143  026304  012701  003010                  MOV     #PATN,R1        ;INIT PATTERN POINTER
7144  026310  012737  026552  002340          MOV     #A2,RETADR      ;SET SUBROUTINE ERROR RETURN ADDRESS
7145  026316                                  BGNSUB
7146  026316                                                                T21.1:
```

```
7147  026316  104402                                                          TRAP    C$BSUB
7148  026320  004737  003554          3$:     JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
7149  026324  004737  004640                  JSR     PC,OSIRDY       ;CHECK ORDY AND OCOR FOR EXPECTED STATES
7150  026330  000001                          1
7151  026332  012737  000011  002356          MOV     #11,REGNUM      ;SET LU REG NO. = 11
7152  026340  111137  002344                  MOVB    (R1),WRIBYT     ;SET DATA BYTE TO BE WRITTEN
7153  026344  004737  003726                  JSR     PC,WRITLU       ;WRITE BYTE INTO REG 11
7154  026350  012737  000010  002356          MOV     #10,REGNUM      ;SET LU REG NO. = 10
7155  026356  116137  000001  002344          MOVB    1(R1),WRIBYT    ;SET DATA BYTE TO BE WRITTEN
7156  026364  004737  003726                  JSR     PC,WRITLU       ;WRITE BYTE INTO REG 10
7157  026370  004737  003726                  JSR     PC,WRITLU       ;WRITE IT AGAIN (SO 2 ENTRIES ARE IN SILO)
7158  026374  004737  005124                  JSR     PC,WAIT50       ;WAIT FOR SILO DATA TO RIPPLE
7159  026400  004737  004640                  JSR     PC,OSIRDY       ;CHECK ORDY AND OCOR FOR EXPECTED STATES
7160  026404  000003                          3
7161  026406  012737  000002  002360          MOV     #2,AXNUM        ;SET BYTE NO. FOR AX1-15
7162  026414  004737  004102                  JSR     PC,READAX       ;READ AX1-15, AX1-16
7163  026420  123761  002346  000001          CMPB    RAX15,1(R1)     ;COMPARE AX1-15 TO EXPECTED
7164  026426  001420                          BEQ     6$              ;BR IF MATCH
7165  026430  005037  002362                  CLR     GOODAT
7166  026434  116137  000001  002362          MOVB    1(R1),GOODAT    ;SET EXPECTED DATA
7167  026442  013737  002346  002364          MOV     RAX15,BADDAT    ;SET ACTUAL DATA
7168  026450  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7169                                  ;REPORT REG MISCOMPARE
7170  026454                                  ERRDF   3,EM3,ERR3
7171  026454  104455                                                          TRAP    C$ERDF
7172  026456  000003                                                          .WORD   3
7173  026460  012176                                                          .WORD   EM3
7174  026462  015116                                                          .WORD   ERR3
7175  026464                                  ESCAPE  SUB
7176  026464  104410                                                          TRAP    C$ESCAPE
7177  026466  000064                                                          .WORD   L10051-.
7178  026470  005237  002360          6$:     INC     AXNUM           ;INCR AX BYTE NO.
7179  026474  123711  002350                  CMPB    RAX16,(R1)      ;COMPARE AX1-16 TO EXPECTED
7180  026500  001417                          BEQ     9$              ;BR IF MATCH
7181  026502  005037  002362                  CLR     GOODAT
7182  026506  111137  002362                  MOVB    (R1),GOODAT     ;SET EXPECTED DATA
7183  026512  013737  002350  002364          MOV     RAX16,BADDAT    ;SET ACTUAL DATA
7184  026520  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7185                                  ;REPORT REG MISCOMPARE
7186  026524                                  ERRDF   3,EM3,ERR3
7187  026524  104455                                                          TRAP    C$ERDF
7188  026526  000003                                                          .WORD   3
7189  026530  012176                                                          .WORD   EM3
7190  026532  015116                                                          .WORD   ERR3
7191  026534                                  ESCAPE  SUB
7192  026534  104410                                                          TRAP    C$ESCAPE
7193  026536  000014                                                          .WORD   L10051-.
7194  026540  062701  000002          9$:     ADD     #2,R1           ;INCR DATA PATTERN POINTER
7195  026544  020127  003026                  CMP     R1,#PAT0        ;SEE IF ALL DATA BYTES WRITTEN YET
7196  026550  103663                          BLO     3$              ;BR IF NOT DONE YET
7197  026552                          A2:
7198  026552                                  ENDSUB
7199  026552                                                          L10051:
7200  026552  104403                                                          TRAP    C$ESUB
7201  026554                          ENDTST
7202  026554                                                          L10050:
```

```
 7203  026554  104401                                                    TRAP    C$ETST
 7204
 7205
 7206
 7207
 7208
 7209                    ;;*****************************************************************************
 7210                    .SBTTL      TEST 22 - TRANSMITTER BUFFER SEQUENCING TEST
 7211                    ;*
 7212                    ;* FIRST, A MASTER CLEAR IS DONE, AND THE PROGRAM CHECKS FOR ORDY=1, OCOR=0.
 7213                    ;* THEN, 2 TSOM CHARS ARE LOADED INTO THE TX SILO, AND ALLOWED TO RIPPLE
 7214                    ;* DOWN TO THE OUTPUT. THE PROGRAM CHECKS FOR ORDY=1, OCOR=1.
 7215                    ;* NEXT, THE PROGRAM CYCLES THE STEPLU BIT UNTIL OCOR=0 AGAIN, AND CHECKS FOR
 7216                    ;* THIS TO OCCUR WITHIN 3 CYCLES.
 7217                    ;* THE SILO IS THEN FILLED WITH 64 BYTES OF A 256-BYTE BINARY COUNT PATTERN
 7218                    ;* (000-377) AND THE PROGRAM CHECKS FOR ORDY=0 AFTER THE 64TH CHAR IS LOADED.
 7219                    ;* THE PROGRAM CYCLES STEPLU FOR 8 CYCLES AND CHECKS THAT AFTER THE 8TH, ORDY=1.
 7220                    ;* AX1-15 IS READ AND COMPARED TO EXPECTED DATA.
 7221                    ;* THE REST OF THE BINARY COUNT DATA BYTES ARE LOADED, CYCLED 8 CLOCKS, READ AND
 7222                    ;* COMPARED, A BYTE AT A TIME. UPON COMPLETION, THE SILO IS CHECKED TO BE EMPTY
 7223                    ;* WITH ORDY=1, OCOR=0.
 7224                    ;;*****************************************************************************
 7225  026556           BGNTST
 7226  026556                                                            T22::
 7227  026556  012737  027232  002340          MOV     #A3,RETADR      ;SET SUBR ERROR RETURN ADDR
 7228
 7229                    ;-----------------------------------------------------------------------------
 7230                    ; SET MASTER CLEAR, CHECK FOR ORDY=1, OCOR=0
 7231  026564  004737  003554                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 7232  026570  004737  004640                  JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=0
 7233  026574  000001                          1
 7234                    ;-----------------------------------------------------------------------------
 7235                    ; LOAD 2 SOM CHARS, ALLOW SILO TO RIPPLE, CHK ORDY=1, OCOR=1
 7236                    ;-----------------------------------------------------------------------------
 7237  026576  012737  000400  002400          MOV     #TXSOM,TXWORD   ;SET DATA TO WRITE INTO SILO
 7238  026604  004737  005152                  JSR     PC,LDTXSI       ;LOAD THE SILO WITH SOM
 7239  026610  004737  005152                  JSR     PC,LDTXSI       ;LOAD ANOTHER SOM
 7240  026614  004737  005124                  JSR     PC,WAIT50       ;WAIT FOR DATA TO RIPPLE
 7241  026620  004737  004640                  JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=1
 7242  026624  000003                          3
 7243                    ;-----------------------------------------------------------------------------
 7244                    ; CLOCK LINE UNIT, CHK FOR OCOR = 0 WITHIN 3 CYCLES
 7245                    ;-----------------------------------------------------------------------------
 7246  026626  005001                          CLR     R1              ;INIT CYCLE COUNTER TO 0
 7247  026630  012737  000017  002356          MOV     #17,REGNUM      ;SET REG NO. = 17
 7248  026636  004737  005232          3$:     JSR     PC,STPLU        ;STEP LU 1 CYCLE
 7249  026642  000001                          1
 7250  026644  004737  003650                  JSR     PC,READLU       ;READ REG 17
 7251  026650  132737  000020  002342          BITB    #OCOR,REDBYT    ;SEE IF OCOR = 0 YET
 7252  026656  001404                          BEQ     6$              ;BR IF OCOR = 0
 7253  026660  005201                          INC     R1              ;INCR CYCLE COUNT
 7254  026662  020127  000003                  CMP     R1,#3           ;SEE IF 3 CYCLES DONE YET
 7255  026666  002763                          BLT     3$              ;BR IF NO
 7256  026670  004737  004640          6$:     JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=0
 7257  026674  000001                          1
 7258                    ;-----------------------------------------------------------------------------
```

```
                                        ; LOAD 64 BINARY COUNT CHARS INTO SILO, CHK ORDY=0
                                        ;-------------------------------------------------------------------------
                                        ;
026676  005003                          CLR     R3              ;INIT PATTERN FOR WRITING
026700  010337  002400          8$:     MOV     R3,TXWORD       ;SET DATA TO BE WRITTEN
026704  004737  005152                  JSR     PC,LDTXSI       ;LOAD DATA CHAR INTO TX SILO
026710  004737  005124                  JSR     PC,WAIT50       ;WAIT FOR DATA TO RIPPLE IN SILO
026714  020327  000077                  CMP     R3,#63.         ;SEE IF 64TH CHAR JUST LOADED
026720  001004                          BNE     9$              ;BR IF NO
026722  012737  000002  026744          MOV     #2,14$          ;SET UP TO CHK ORDY=0,OCOR=1
026730  000403                          BR      12$
026732  012737  000003  026744  9$:     MOV     #3,14$          ;SET UP TO CHK ORDY=1,OCOR=1
026740  004737  004640          12$:    JSR     PC,OSIRDY       ;CHK ORDY, OCOR
026744  000000                  14$:    .WORD   0
026746  005203                          INC     R3              ;INCR PATTERN FOR WRITES
026750  020327  000100                  CMP     R3,#64.         ;SEE IF 64 CHARS LOADED YET
026754  002751                          BLT     8$              ;BR IF NO
                                        ;-------------------------------------------------------------------------
                                        ; CLOCK LINE UNIT, CHECK ORDY = 1 WITHIN 8 CYCLES
                                        ;-------------------------------------------------------------------------
                                        ;
026756  012737  000011  002356          MOV     #11,REGNUM      ;SET REG NO. = 11
026764  005001                          CLR     R1              ;INIT CYCLE COUNT
026766  004737  005232          16$:    JSR     PC,STPLU        ;CLOCK LU FOR 1 CYCLE
026772  000001                          1
026774  004737  003650                  JSR     PC,READLU       ;READ REG 11
027000  132737  000020  002342          BITB    #ORDY,REDBYT    ;SEE IF ORDY = 1 YET
027006  001004                          BNE     19$             ;BR IF YES
027010  005201                          INC     R1              ;INCR CYCLE COUNT
027012  020127  000010                  CMP     R1,#8.          ;SEE IF 8 CYCLES YET
027016  002763                          BLT     16$             ;BR IF NOT YET
027020  004737  004640          19$:    JSR     PC,OSIRDY       ;CHK ORDY = 1, OCOR = 1
027024  000003                          3
                                        ;-------------------------------------------------------------------------
                                        ; READ AND COMPARE FIRST CHARACTER IN AX1-15
                                        ;-------------------------------------------------------------------------
                                        ;
027026  005004                          CLR     R4              ;INIT PATTERN FOR READING
027030  012737  000002  002360          MOV     #2,AXNUM        ;SET AX BYTE NO. FOR AX1-15
027036  004737  004102                  JSR     PC,READAX       ;READ AX1-15
027042  123704  002346                  CMPB    RAX15,R4        ;COMPARE AX1-15 TO EXPECTED
027046  001415                          BEQ     20$             ;BR IF MATCH
027050  010437  002362                  MOV     R4,GOODAT       ;SET EXPECTED DATA
027054  013737  002346  002364          MOV     RAX15,BADDAT    ;SET ACTUAL DATA
027062  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                ;REPORT REG MISCOMPARE
027066                                  ERRDF   3,EM3,ERR3
027066  104455                                                                  TRAP    C$ERDF
027070  000003                                                                  .WORD   3
027072  012176                                                                  .WORD   EM3
027074  015116                                                                  .WORD   ERR3
027076                                  ESCAPE  TST
027076  104410                                                                  TRAP    C$ESCAPE
027100  000132                                                                  .WORD   L10052-.
027102                          20$:
                                        ;-------------------------------------------------------------------------
                                        ; LOAD AND COMPARE REST OF CHARS, MONITOR ORDY, OCOR
                                        ;-------------------------------------------------------------------------
                                        ;
027102  020327  000377          24$:    CMP     R3,#255.        ;SEE IF ALL CHARS LOADED YET
```

```
7315  027106  003010                    BGT     26$             ;BR IF YES
7316  027110  010337  002400            MOV     R3,TXWORD       ;SET DATA TO BE WRITTEN
7317  027114  004737  005152            JSR     PC,LDTXSI       ;LOAD DATA CHAR INTO TX SILO
7318  027120  005203                    INC     R3              ;INCR DATA TO BE WRITTEN
7319  027122  004737  004640            JSR     PC,OSIRDY       ;CHK ORDY=0, OCOR=1
7320  027126  000002                    2
7321  027130  005204            26$:    INC     R4              ;INCR PAT FOR READING
7322  027132  004737  005232            JSR     PC,STPLU        ;CLOCK LINE UNIT FOR 8 CYCLES
7323  027136  000010                    8.
7324  027140  004737  004102            JSR     PC,READAX       ;READ AX1-15
7325  027144  123704  002346            CMPB    RAX15,R4        ;COMPARE AX1-15 TO EXPECTED
7326  027150  001415                    BEQ     27$             ;BR IF MATCH
7327  027152  010437  002362            MOV     R4,GOODAT       ;SET EXPECTED DATA
7328  027156  013737  002346  002364    MOV     RAX15,BADDAT    ;SET ACTUAL DATA
7329  027164  004737  004504            JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7330                            ;REPORT REG MISCOMPARE
7331  027170                            ERRDF   3,EM3,ERR3
7332  027170  104455                                                    TRAP    C$ERDF
7333  027172  000003                                                    .WORD   3
7334  027174  012176                                                    .WORD   EM3
7335  027176  015116                                                    .WORD   ERR3
7336  027200                            ESCAPE  TST
7337  027200  104410                                                    TRAP    C$ESCAPE
7338  027202  000030                                                    .WORD   L10052-.
7339  027204  020427  000377    27$:    CMP     R4,#255.        ;SEE IF WE READ LAST CHAR YET
7340  027210  001004                    BNE     29$             ;BR IF NOT
7341  027212  004737  004640            JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=0
7342  027216  000001                    1
7343  027220  000404                    BR      A3
7344  027222  004737  004640    29$:    JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=1
7345  027226  000003                    3
7346  027230  000724                    BR      24$             ;CONTINUE
7347  027232                    A3:
7348  027232                    ENDTST
7349  027232                                                          L10052:
7350  027232  104401                                                          TRAP    C$ETST
7351
7352
7353
7354
7355
7356                            ;;*****************************************************************************
7357                            .SBTTL      TEST 23 - TX MSG TIMING TEST, CHAR MODE, WITH CRC
7358                            ;*
7359                            ;* IN THIS TEST,  N ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
7360                            ;* AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
7361                            ;* (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
7362                            ;* CHARACTER ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND CRC-16.
7363                            ;* THE FOLLOWING STEPS ARE DONE:
7364                            ;* A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN CHAR MODE.
7365                            ;* SOM IS SET TWICE TO SEND 2 SYNCH CHARS. THEN, 2 000 CHARS ARE SENT, AND
7366                            ;* THEN 2 TERMINATING SYNCHS ARE SENT.
7367                            ;* THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
7368                            ;* CLEARED STATE AFTER THE 3RD SYNCH COMPLETES.
7369                            ;;*****************************************************************************
7370  027234                    BGNTST
```

```
7371  027234                                                                  T23::
7372  027234  012737  027346  002340        MOV     #A4,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
7373  027242  004737  005516                JSR     PC,INITRN       ;DO MASTER CLR, LOAD 2 SOM'2
7374  027246  000226                        SYNCH
7375  027250  000011                        STRIP!DDCMP
7376  027252  004737  006100                JSR     PC,TXCHAR       ;LOAD A 000 CHAR, TX FIRST SYNCH (226)
7377  027256  000000                        000
7378  027260  100010                        CHPCHK!8.
7379  027262  004737  006100                JSR     PC,TXCHAR       ;LOAD 2ND 000 CHAR, TX 2ND SYNCH
7380  027266  000000                        000
7381  027270  000010                        8.
7382  027272  004737  006100                JSR     PC,TXCHAR       ;LOAD EOM CHAR, TX FIRST 000 CHAR
7383  027276  001000                        TXEOM
7384  027300  000010                        8.
7385  027302  004737  006100                JSR     PC,TXCHAR       ;LOAD EOM CHAR, TX 2ND 000 CHAR
7386  027306  001000                        TXEOM
7387  027310  000010                        8.
7388  027312  004737  006100                JSR     PC,TXCHAR       ;LOAD EOM, TX CRC-16 CHAR
7389  027316  001000                        TXEOM
7390  027320  000020                        16.
7391  027322  004737  006100                JSR     PC,TXCHAR       ;LOAD EOM, TX FIRST TERMINATING SYNCH
7392  027326  001000                        TXEOM
7393  027330  000010                        8.
7394  027332  004737  006100                JSR     PC,TXCHAR       ;LOAD EOM, TX 2ND TERMINATING SYNCH
7395  027336  001000                        TXEOM
7396  027340  000010                        8.
7397  027342  004737  006252                JSR     PC,ENDTRN       ;SET OC, MONITOR OCOR
7398  027346                        A4:
7399  027346  004737  003554                JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
7400  027352                        ENDTST
7401  027352                                                                  L10053:
7402  027352  104401                                                                  TRAP    C$ETST
7403
7404
7405
7406
7407
7408                                  ;;*************************************************************************
7409                                  .SBTTL      TEST 24 - TX MSG TIMING TEST, BIT MODE, WITH CRC
7410                                  ;*
7411                                  ;* IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
7412                                  ;* AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
7413                                  ;* (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
7414                                  ;* BIT ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND CRC-CCITT-1.
7415                                  ;* THE FOLLOWING STEPS ARE DONE:
7416                                  ;* A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN BIT MODE.
7417                                  ;* SOM IS SET TWICE TO SEND 2 FLAG CHARS. THEN, 2 000 CHARS ARE SENT, AND
7418                                  ;* THEN 2 TERMINATING FLAGS ARE SENT.
7419                                  ;* THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
7420                                  ;* CLEARED STATE.
7421                                  ;;*************************************************************************
7422  027354                        BGNTST
7423  027354                                                                  T24::
7424  027354  012737  027456  002340        MOV     #A5,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
7425  027362  004737  005516                JSR     PC,INITRN       ;DO MASTER CLR, LOAD 2 SOM'2
7426  027366  000000                        000
```

```
                        TEST 24 - TX MSG TIMING TEST, BIT MODE, WITH CRC

 7427  027370  000000                      000
 7428  027372  004737  006100              JSR     PC,TXCHAR       ;LOAD A 000 CHAR, TX FIRST FLAG
 7429  027376  000000                      000
 7430  027400  100010                      CHPCHK!8.
 7431  027402  004737  006100              JSR     PC,TXCHAR       ;LOAD 2ND 000 CHAR, TX 2ND FLAG
 7432  027406  000000                      000
 7433  027410  000010                      8.
 7434  027412  004737  006100              JSR     PC,TXCHAR       ;LOAD EOM CHAR, TX FIRST 000 CHAR
 7435  027416  001000                      TXEOM
 7436  027420  000010                      8.
 7437  027422  004737  006100              JSR     PC,TXCHAR       ;LOAD EOM, TX 2ND 000 CHAR AND CRC-CCITT-1 CHAR
 7438  027426  001000                      TXEOM
 7439  027430  000030                      24.
 7440  027432  004737  006100              JSR     PC,TXCHAR       ;LOAD EOM, TX FIRST TERMINATING FLAG
 7441  027436  001000                      TXEOM
 7442  027440  000010                      8.
 7443  027442  004737  006100              JSR     PC,TXCHAR       ;LOAD EOM, TX 2ND TERMINATING FLAG
 7444  027446  001000                      TXEOM
 7445  027450  000010                      8.
 7446  027452  004737  006252              JSR     PC,ENDTRN       ;SET OC, MONITOR OCOR
 7447  027456                      A5:
 7448  027456  004737  003554              JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
 7449  027462                      ENDTST
 7450  027462                                                              L10054:
 7451  027462  104401                                                              TRAP    C$ETST
 7452
 7453
 7454
 7455
 7456
 7457                            ;;*************************************************************************
 7458                            .SBTTL      TEST 25 - TX MSG TIMING TEST, CHAR MODE, WITH NO CRC
 7459                            ;*
 7460                            ;* IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
 7461                            ;* AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
 7462                            ;* (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
 7463                            ;* CHARACTER ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND NO ERROR CHECKING.
 7464                            ;* THE FOLLOWING STEPS ARE DONE:
 7465                            ;* A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN CHAR MODE.
 7466                            ;* SOM IS SET TWICE TO SEND 2 SYNCH CHARS. THEN, 2 000 CHARS ARE SENT, AND
 7467                            ;* THEN 2 TERMINATING SYNCHS ARE SENT.
 7468                            ;* THE TEST IS PERFORMED WITH TXEN (REG14, BIT6) SET, AND THE PROGRAM CHECKS
 7469                            ;* THAT THIS HOLDS RTS HIGH PAST THE END OF THE MESSAGE.
 7470                            ;* THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
 7471                            ;* CLEARED STATE.
 7472                            ;;*************************************************************************
 7473  027464                    BGNTST
 7474  027464                                                                      T25::
 7475  027464  012737  027652  002340          MOV     #A6,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
 7476  027472  004737  005516                  JSR     PC,INITRN       ;DO MASTER CLR, LOAD 2 SOM'2
 7477  027476  000226                          SYNCH
 7478  027500  000311                          CRC2!CRC1!STRIP!DDCMP
 7479  027502  012737  000014  002356          MOV     #14,REGNUM      ;SET REG NO. = 14
 7480  027510  012737  000100  002344          MOV     #TXEN,WRIBYT    ;SET TXEN BIT
 7481  027516  004737  003726                  JSR     PC,WRITLU       ;LOAD TXEN BIT IN REG 14
 7482  027522  004737  006100                  JSR     PC,TXCHAR       ;LOAD A 000 CHAR, TX FIRST SYNCH (226)
```

```
7483  027526  000000                          000
7484  027530  100010                          CHPCHK!8.
7485  027532  004737  006100                  JSR     PC,TXCHAR       ;LOAD 2ND 000 CHAR, TX 2ND SYNCH
7486  027536  000000                          000
7487  027540  000010                          8.
7488  027542  004737  006100                  JSR     PC,TXCHAR       ;LOAD EOM CHAR, TX FIRST 000 CHAR
7489  027546  001000                          TXEOM
7490  027550  000010                          8.
7491  027552  004737  006100                  JSR     PC,TXCHAR       ;LOAD EOM CHAR, TX 2ND 000 CHAR
7492  027556  001000                          TXEOM
7493  027560  000010                          8.
7494  027562  004737  006100                  JSR     PC,TXCHAR       ;LOAD EOM, TX FIRST TERMINATING SYNCH
7495  027566  001000                          TXEOM
7496  027570  000010                          8.
7497  027572  004737  006100                  JSR     PC,TXCHAR       ;LOAD EOM, TX 2ND TERMINATING SYNCH
7498  027576  001000                          TXEOM
7499  027600  000010                          8.
7500  027602  004737  005232                  JSR     PC,STPLU        ;CLK PAST END OF MSG
7501  027606  000030                          24.
7502  027610  012737  000013  002356          MOV     #13,REGNUM      ;SET REG NO. = 13
7503  027616  004737  003650                  JSR     PC,READLU       ;READ REG 13
7504  027622  032737  000040  002342          BIT     #RTS,REDBYT     ;CHK FOR RTS STILL SET
7505  027630  001006                          BNE     3$              ;BR IF RTS SET
7506  027632  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7507                                  ;REPORT RTS NOT SET
7508  027636                                  ERRDF   60,EM60,ERR7
7509  027636  104455                                                  TRAP    C$ERDF
7510  027640  000074                                                  .WORD   60
7511  027642  014206                                                  .WORD   EM60
7512  027644  017446                                                  .WORD   ERR7
7513  027646  004737  006252          3$:     JSR     PC,ENDTRN       ;SET OC, MONITOR OCOR
7514  027652                          A6:
7515  027652  004737  003554                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
7516  027656                          ENDTST
7517  027656                                                          L10055:
7518  027656  104401                                                  TRAP    C$ETST
7519
7520
7521
7522
7523
7524                                  ;;*************************************************************************
7525                                  .SBTTL      TEST 26 - TX UNDERRUN SET AND CLEAR TEST - CHAR MODE
7526                                  ;*
7527                                  ;* IN THIS TEST, A TX UNDERRUN ERROR IS FORCED IN EACH OF 2 SITUATIONS,
7528                                  ;* AND THEN CLEARED DIFFERENTLY IN EACH.
7529                                  ;* IN THE FIRST, A MESSAGE IS INITIATED, A 000 CHAR IS SENT, AND THE TX
7530                                  ;* BUFFER IS NOT SERVICED IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG,
7531                                  ;* WHICH CAUSES UNRR TO SET IN REG 11. THEN, SOM IS SET TO CLEAR THE ERROR,
7532                                  ;* AND THIS IS VERIFIED.
7533                                  ;* IN THE SECOND SITUATION, A MSG IS INITIATED, A 000 CHAR IS SENT, AND THE TX
7534                                  ;* BUFFER IS NOT SERVICED IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG, WHICH
7535                                  ;* AGAIN CAUSES UNRR TO SET. THEN, A MASTER CLEAR IS DONE, AND THE UNRR BIT
7536                                  ;* IS CHECKED TO BE CLEARED.
7537                                  ;*************************************************************************
7538  027660                          BGNTST
```

```
7539  027660                                                                T26::
7540                                ;-------------------------------------------------------------------
7541                                ; CAUSE TX UNDERRUN, CHK UNRR = 1; SET SOM, CHK UNRR = 0
7542                                ;-------------------------------------------------------------------
7543  027660  012737  030216  002340          MOV     #A7,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
7544  027666  004737  005516                  JSR     PC,INITRN       ;DO MASTER CLEAR, LOAD 2 SOM'S
7545  027672  000226                          SYNCH
7546  027674  000011                          STRIP!DDCMP
7547  027676  004737  006100                  JSR     PC,TXCHAR       ;LOAD A 000 CHAR, TX FIRST SYNCH
7548  027702  000000                          000
7549  027704  100010                          CHPCHK!8.
7550  027706  004737  005232                  JSR     PC,STPLU        ;CLOCK THE TRANSMITTER UNTIL 7 BITS OF
7551  027712  000016                          14.                     ;    THE 000 CHAR HAVE BEEN TRANSMITTED
7552  027714  012737  000011  002356          MOV     #11,REGNUM      ;SET LU REG NO. = 11
7553  027722  004737  003650                  JSR     PC,READLU       ;READ REG 11
7554  027726  132737  000001  002342          BITB    #UNRR,REDBYT    ;CHK FOR UNRR = 0
7555  027734  001410                          BEQ     6$              ;BR IF UNRR = 0
7556  027736  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7557                              ;REPORT UNRR NOT CLEARED
7558  027742                                  ERRDF   16,EM16,ERR7
7559  027742  104455                                                                  TRAP    C$ERDF
7560  027744  000020                                                                  .WORD   16
7561  027746  012600                                                                  .WORD   EM16
7562  027750  017446                                                                  .WORD   ERR7
7563  027752  000137  030216                  JMP     A7              ;SKIP TO END OF TEST
7564  027756  004737  005232          6$:     JSR     PC,STPLU        ;CLOCK LAST BIT OF 000 CHAR
7565  027762  000003                          3
7566  027764  004737  003650                  JSR     PC,READLU       ;READ REG 11
7567  027770  132737  000001  002342          BITB    #UNRR,REDBYT    ;CHK FOR UNRR = 1
7568  027776  001010                          BNE     9$              ;BR IF UNRR = 1
7569  030000  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7570                              ;REPORT UNRR NOT SET
7571  030004                                  ERRDF   14,EM14,ERR7
7572  030004  104455                                                                  TRAP    C$ERDF
7573  030006  000016                                                                  .WORD   14
7574  030010  012534                                                                  .WORD   EM14
7575  030012  017446                                                                  .WORD   ERR7
7576  030014  000137  030216                  JMP     A7              ;SKIP TO END OF TEST
7577  030020  012737  000400  002400  9$:     MOV     #TXSOM,TXWORD   ;SET SOM CHAR TO BE WRITTEN
7578  030026  004737  005152                  JSR     PC,LDTXSI       ;LOAD SOM CHAR INTO TX SILO
7579  030032  004737  005124                  JSR     PC,WAIT50       ;WAIT FOR SILO DATA TO RIPPLE
7580  030036  004737  005232                  JSR     PC,STPLU        ;CLOCK LU FOR 2 CYCLES
7581  030042  000002                          2
7582  030044  004737  003650                  JSR     PC,READLU       ;READ REG 11
7583  030050  132737  000001  002342          BITB    #UNRR,REDBYT    ;CHK FOR UNRR = 0
7584  030056  001410                          BEQ     12$             ;BR IF UNRR = 0
7585  030060  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7586                              ;REPORT UNRR NOT CLEARED BY SOM
7587  030064                                  ERRDF   13,EM13,ERR7
7588  030064  104455                                                                  TRAP    C$ERDF
7589  030066  000015                                                                  .WORD   13
7590  030070  012504                                                                  .WORD   EM13
7591  030072  017446                                                                  .WORD   ERR7
7592  030074  000137  030216                  JMP     A7              ;SKIP TO END OF TEST
7593  030100                          12$:
7594                              ;-------------------------------------------------------------------
```

```
7595                                    ;CAUSE TX UNDERRUN, CHK UNRR = 1; DO MASTER CLR, CHK UNRR = 0
7596                                    ;-------------------------------------------------------------------------------
7597  030100  004737  005516            JSR     PC,INITRN       ;DO MASTER CLEAR, LOAD 2 SOM'S
7598  030104  000226                    SYNCH
7599  030106  000051                    IDLE!STRIP!DDCMP
7600  030110  004737  006100            JSR     PC,TXCHAR       ;LOAD A 000 CHAR, TX FIRST SYNCH
7601  030114  000000                    000
7602  030116  100010                    CHPCHK!8.
7603  030120  004737  005232            JSR     PC,STPLU        ;STEP THE LU UNTIL 000 HAS BEN TRANSMITTED
7604  030124  000021                    17.
7605  030126  004737  003650            JSR     PC,READLU       ;READ REG 11
7606  030132  132737  000001  002342    BITB    #UNRR,REDBYT    ;CHK FOR UNR = 1
7607  030140  001010                    BNE     16$             ;BR IF UNRR = 1
7608  030142  004737  004504            JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7609                                    ;REPORT UNRR NOT SET
7610  030146                            ERRDF   14,EM14,ERR7
7611  030146  104455                                                            TRAP    C$ERDF
7612  030150  000016                                                            .WORD   14
7613  030152  012534                                                            .WORD   EM14
7614  030154  017446                                                            .WORD   ERR7
7615  030156  000137  030216            JMP     A7              ;SKIP TO END OF TEST
7616  030162  004737  003554    16$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
7617  030166  004737  003650            JSR     PC,READLU       ;READ REG 11
7618  030172  132737  000001  002342    BITB    #UNRR,REDBYT    ;CHK FOR UNRR = 0
7619  030200  001406                    BEQ     A7              ;BR IF UNRR = 0
7620  030202  004737  004504            JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7621                                    ;REPORT UNRR NOT CLEARED BY OC
7622  030206                            ERRDF   15,EM15,ERR7
7623  030206  104455                                                            TRAP    C$ERDF
7624  030210  000017                                                            .WORD   15
7625  030212  012551                                                            .WORD   EM15
7626  030214  017446                                                            .WORD   ERR7
7627  030216  004737  003554    A7:     JSR     PC,MSTCLR       ;ISSUE CLEAN-UP MASTER CLEAR
7628  030222                            ENDTST
7629  030222                                                            L10056:
7630  030222  104401                                                            TRAP    C$ETST
7631
7632
7633
7634
7635
7636                                    ;;*****************************************************************************
7637                                    .SBTTL      TEST 27 - TRANSMIT CHAR LENGTH TIMING TEST - CHAR MODE, CRC
7638                                    ;*
7639                                    ;* THE LINE UNIT IS PLACED IN CHAR MODE (DDCMP) AND A MESSAGE IS INITIATED
7640                                    ;* WITH AN 8-BIT SYNCH AND A 5-BIT SYNCH CHAR. NEXT, A 000 CHAR IS SENT WITH
7641                                    ;* EACH OF THE FOLLOWING TX CHAR LENGTHS : 5 BITS, 6 BITS, 7 BITS, 8 BITS.
7642                                    ;* (FOR EXAMPLE, A 5-BIT CHAR REQUIRES 5 CLOCK CYCLES TO BE TRANSMITTED). TWO
7643                                    ;* TERMINATING SYNCHS ARE SENT AFTER THE DATA.
7644                                    ;;*****************************************************************************
7645  030224                            BGNTST
7646  030224                                                            T27::
7647  030224  012737  030432  002340            MOV     #A8,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
7648  030232  004737  005516                    JSR     PC,INITRN       ;DO MASTER CLR, LOAD 2 SOM'S
7649  030236  000000                            000
7650  030240  000041                            IDLE!DDCMP
```

CZDMRF.P11 03-NOV-81 10:29 TEST 27 - TRANSMIT CHAR LENGTH TIMING TEST - CHAR MODE, CRC

```
7651  030242  012737  000006  002360          MOV     #6,AXNUM        ;SET BYTE NO. = 6 FOR AX3
7652  030250  012737  000000  002352          MOV     #000,WAX15      ;SET DATA FOR AX3-15 = 0
7653  030256  012737  000240  002354          MOV     #TXLEN2!TXLEN0,WAX16 ;SET TX LENGTH = 5 FOR AX3-16
7654  030264  004737  004270                  JSR     PC,WRITAX       ;LOAD AX3-15,AX3-16
7655  030270  004737  006100                  JSR     PC,TXCHAR       ;LOAD 5-BIT 000 CHAR, TX 8-BIT SYNCH
7656  030274  000000                          000
7657  030276  100010                          CHPCHK!8.
7658  030300  004737  006100                  JSR     PC,TXCHAR       ;LOAD 6-BIT 000 CHAR, TX 5-BIT SYNCH
7659  030304  000000                          000
7660  030306  000005                          5
7661  030310  012737  000300  002354          MOV     #TXLEN2!TXLEN1,WAX16
7662  030316  004737  004270                  JSR     PC,WRITAX       ;SET TX CHAR LENGTH = 6
7663  030322  004737  006100                  JSR     PC,TXCHAR       ;LOAD 7-BIT 000 CHAR, TX 5-BIT 000 CHAR
7664  030326  000000                          000
7665  030330  000005                          5
7666  030332  012737  000340  002354          MOV     #TXLEN2!TXLEN1!TXLEN0,WAX16
7667  030340  004737  004270                  JSR     PC,WRITAX       ;SET TX CHAR LENGTH = 7
7668  030344  004737  006100                  JSR     PC,TXCHAR       ;LOAD 8-BIT 000 CHAR, TX 6-BIT 000 CHAR
7669  030350  000000                          000
7670  030352  000006                          6
7671  030354  012737  000000  002354          MOV     #000,WAX16
7672  030362  004737  004270                  JSR     PC,WRITAX       ;SET TX CHAR LENGTH = 8
7673  030366  004737  006100                  JSR     PC,TXCHAR       ;LOAD EOM, TX 7-BIT 000 CHAR
7674  030372  001000                          TXEOM
7675  030374  000007                          7
7676  030376  004737  006100                  JSR     PC,TXCHAR       ;LOAD EOM, TX 8-BIT 000 CHAR
7677  030402  001000                          TXEOM
7678  030404  000010                          8.
7679  030406  004737  006100                  JSR     PC,TXCHAR       ;LOAD EOM, TX CRC-16 CHAR
7680  030412  001000                          TXEOM
7681  030414  000020                          16.
7682  030416  004737  006100                  JSR     PC,TXCHAR       ;LOAD EOM, TX FIRST TERMINATING SYNCH
7683  030422  001000                          TXEOM
7684  030424  000010                          8.
7685  030426  004737  006252                  JSR     PC,ENDTRN       ;CLEAR TRANSMITTER
7686  030432                          A8:
7687  030432                          ENDTST
7688  030432                                                          L10057:
7689  030432  104401                                                          TRAP    C$ETST
7690
7691
7692
7693
7694
7695                                ;;*************************************************************************
7696                                .SBTTL      TEST 28 - TRANSMIT CHAR LENGTH TIMING TEST - BIT MODE, CRC
7697                                ;*
7698                                ;* THE LINE UNIT IS PLACED IN BIT MODE AND A MESSAGE IS INITIATED
7699                                ;* WITH 2 FLAG CHARS. NEXT, 2 8-BIT 000 CHARS ARE SENT, FOLLOWED BY 000 CHARS
7700                                ;* WITH EACH OF THE FOLLOWING TRANSMITTER CHAR LENGTHS:
7701                                ;*     1 BIT, 2 BITS, 3 BITS, 4 BITS, 5 BITS, 6 BITS, 7 BITS, AND 8 BITS.
7702                                ;* (FOR EXAMPLE, A 5-BIT CHAR REQUIRES 5 CLOCK CYCLES TO BE TRANSMITTED).
7703                                ;* TWO TERMINATING FLAGS ARE SENT AFTER THE DATA.
7704                                ;;*************************************************************************
7705  030434                          BGNTST
7706  030434                                                          T28::
```

```
 7707  030434  012737  030762  002340          MOV     #A9,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
 7708  030442  004737  005516                  JSR     PC,INITRN       ;DO MASTER CLR, LOAD 2 SOM'S
 7709  030446  000000                          000
 7710  030450  000000                          000
 7711  030452  004737  006100                  JSR     PC,TXCHAR       ;LOAD FIRST 8-BIT OCO CHAR, TX FIRST FLAG
 7712  030456  000000                          000
 7713  030460  100010                          CHPCHK!8.
 7714  030462  004737  006100                  JSR     PC,TXCHAR       ;LOAD 2ND 8-BIT 000 CHAR, TX 2ND FLAG
 7715  030466  000000                          000
 7716  030470  000010                          8.
 7717  030472  004737  006100                  JSR     PC,TXCHAR       ;LOAD 1-BIT 000 CHAR, TX FIRST 8-BIT 000 CHAR
 7718  030476  000000                          000
 7719  030500  000010                          8.
 7720  030502  012737  000006  002360          MOV     #6,AXNUM        ;SET BYTE NO. = 6 FOR AX3
 7721  030510  012737  000000  002352          MOV     #000,WAX15      ;SET DATA FOR AX3-15 = 0
 7722  030516  012737  000040  002354          MOV     #TXLEN0,WAX16   ;SET TX CHAR LENGTH = 1 FOR AX3-16
 7723  030524  004737  004270                  JSR     PC,WRITAX       ;LOAD AX3-15,AX3-16
 7724  030530  004737  006100                  JSR     PC,TXCHAR       ;LOAD 2-BIT 000 CHAR, TX 2ND 8-BIT 000 CHAR
 7725  030534  000000                          000
 7726  030536  000010                          8.
 7727  030540  012737  000100  002354          MOV     #TXLEN1,WAX16
 7728  030546  004737  004270                  JSR     PC,WRITAX       ;SET TX CHAR LENGTH = 2
 7729  030552  004737  006100                  JSR     PC,TXCHAR       ;LOAD 3-BIT 000 CHAR, TX 1-BIT 000 CHAR
 7730  030556  000000                          000
 7731  030560  000001                          1
 7732  030562  012737  000140  002354          MOV     #TXLEN1!TXLEN0,WAX16
 7733  030570  004737  004270                  JSR     PC,WRITAX       ;SET TX CHAR LENGTH = 3
 7734  030574  004737  006100                  JSR     PC,TXCHAR       ;LOAD 4-BIT 000 CHAR, TX 2-BIT 000 CHAR
 7735  030600  000000                          000
 7736  030602  000002                          2
 7737  030604  012737  000200  002354          MOV     #TXLEN2,WAX16
 7738  030612  004737  004270                  JSR     PC,WRITAX       ;SET TX CHAR LENGTH = 4
 7739  030616  004737  006100                  JSR     PC,TXCHAR       ;LOAD 5-BIT 000 CHAR, TX 3-BIT 000 CHAR
 7740  030622  000000                          000
 7741  030624  000003                          3
 7742  030626  012737  000240  002354          MOV     #TXLEN2!TXLEN0,WAX16
 7743  030634  004737  004270                  JSR     PC,WRITAX       ;SET TX CHAR LENGTH = 5
 7744  030640  004737  006100                  JSR     PC,TXCHAR       ;LOAD 6-BIT 000 CHAR, TX 4-BIT 000 CHAR
 7745  030644  000000                          000
 7746  030646  000004                          4
 7747  030650  012737  000300  002354          MOV     #TXLEN2!TXLEN1,WAX16
 7748  030656  004737  004270                  JSR     PC,WRITAX       ;SET TX CHAR LENGTH = 6
 7749  030662  004737  006100                  JSR     PC,TXCHAR       ;LOAD 7-BIT 000 CHAR, TX 5-BIT 000 CHAR
 7750  030666  000000                          000
 7751  030670  000005                          5
 7752  030672  012737  000340  002354          MOV     #TXLEN2!TXLEN1!TXLEN0,WAX16
 7753  030700  004737  004270                  JSR     PC,WRITAX       ;SET TX CHAR LENGTH = 7
 7754  030704  004737  006100                  JSR     PC,TXCHAR       ;LOAD 8-BIT 000 CHAR, TX 6-BIT 000 CHAR
 7755  030710  000000                          000
 7756  030712  000006                          6
 7757  030714  012737  000000  002354          MOV     #000,WAX16
 7758  030722  004737  004270                  JSR     PC,WRITAX       ;SET TX CHAR LENGTH = 8
 7759  030726  004737  006100                  JSR     PC,TXCHAR       ;LOAD EOM, TX 7-BIT 000 CHAR
 7760  030732  001000                          TXEOM
 7761  030734  000007                          7
 7762  030736  004737  006100                  JSR     PC,TXCHAR       ;LOAD EOM, TX 8-BIT 000 CHAR, CRC-CCITT-1 CHAR
```

```
 7763  030742  001000                          TXEOM
 7764  030744  000031                          25.
 7765  030746  004737  006100                  JSR     PC,TXCHAR       ;LOAD EOM, TX FIRST TERMINATING FLAG
 7766  030752  001000                          TXEOM
 7767  030754  000010                          8.
 7768  030756  004737  006252                  JSR     PC,ENDTRN       ;CLEAR TRANSMITTER
 7769  030762                          A9:
 7770  030762                          ENDTST
 7771  030762                                                                  L10060:
 7772  030762  104401                                                                  TRAP    C$ETST
 7773
 7774
 7775
 7776
 7777
 7778                                  ;;*************************************************************************
 7779                                  .SBTTL      TEST 29 - TXDATA BIT TEST - CHAR MODE, CRC
 7780                                  ;*
 7781                                  ;* THE LINE UNIT IS INITIALIZED AND A MSG IS INITIATED (USING STEPLU) WITH CRC-
 7782                                  ;* 16 SELECTED IN CHAR MODE. TWO SYNCHS, 000,125,252,377,000, AND 2 TERMINATING
 7783                                  ;* SYNCHS ARE THEN SENT. THE PROGRAM CHECKS EACH BIT OF THE TRANSMITTED
 7784                                  ;* DATA CHARS, BY MONITORING TXDATA (REG 17) AS THE DATA IS CLOCKED OUT OF
 7785                                  ;* THE USYRT TRANSMITTER.
 7786                                  ;;*************************************************************************
 7787  030764                          BGNTST
 7788  030764                                                                  T29::
 7789  030764  004737  005516                  JSR     PC,INITRN       ;DO MASTER CLR, LOAD 2 SOM'S
 7790  030770  000226                          SYNCH
 7791  030772  000011                          STRIP!DDCMP
 7792  030774  012701  003202                  MOV     #MSG1+4,R1      ;GET POINTER TO DATA
 7793  031000  010103                          MOV     R1,R3
 7794  031002  012137  002400          3$:     MOV     (R1)+,TXWORD
 7795  031006  004737  005152                  JSR     PC,LDTXSI       ;LOAD A DATA CHAR INTO TX SILO
 7796  031012  020127  003220                  CMP     R1,#MSG1+18.    ;SEE IF ALL CHARS LOADED YET
 7797  031016  103771                          BLO     3$              ;BR IF NOT YET
 7798  031020  004737  005124                  JSR     PC,WAIT50       ;WAIT FOR SILO TO RIPPLE
 7799  031024  004737  005232                  JSR     PC,STPLU        ;CLOCK LU UNTIL SYNCHS ARE TX'D
 7800  031030  100020                          CHPCHK!16.
 7801  031032  011337  031042          6$:     MOV     (R3),8$         ;GET EXPECTED DATA CHAR
 7802  031036  004737  011154                  JSR     PC,CKTBIT       ;CHECK TXDATA FOR CHAR BITS
 7803  031042  000000          8$:     .WORD   0                       ;EXPECTED CHAR GOES HERE
 7804  031044  062703  000002                  ADD     #2,R3           ;INCR PATTERN POINTER
 7805  031050  020327  003214                  CMP     R3,#MSG1+14.    ;SEE IF ALL CHARS CHECKED YET
 7806  031054  103766                          BLO     6$              ;BR IF NOT YET
 7807  031056  004737  003554          16$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
 7808  031062                          ENDTST
 7809  031062                                                                  L10061:
 7810  031062  104401                                                                  TRAP    C$ETST
 7811
 7812
 7813
 7814
 7815
 7816                                  ;;*************************************************************************
 7817                                  .SBTTL      TEST 30 - USYRT RECEIVER MSG TEST - CHAR MODE, CRC
 7818                                  ;*
```

```
                        ;* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
                        ;* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH CRC-16
                        ;* SELECTED. TWO SYNCHS, 000,125,252,377,000, AND FOUR TERMINATING SYNCHS ARE
                        ;* SENT. THE PROGRAM MONITORS IACT, AND THE RCV'D CHARS AND CRC BYTES ARE READ
                        ;* FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR IACT
                        ;* STILL SET AFTER THE MESSAGE.
                        ;;*****************************************************************************
031064                  BGNTST
031064                                                                          T30::
031064  012737  031426  002340          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
031072  004737  003554                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
031076  004737  010616                  JSR     PC,SETUP        ;PROGRAM THE USYRT
031102  000226                          SYNCH
031104  000013                          STRIP!IERR!DDCMP
031106  000000                          000
031110  000000                          000
031112  012737  000012  002356          MOV     #12,REGNUM      ;SET LU REG NO. = 12
031120  005037  002360                  CLR     AXNUM           ;SET AX BYTE NO. = 0 FOR AX0
031124  112737  000040  002344          MOVB    #LULP,WRIBYT
031132  004737  003726                  JSR     PC,WRITLU       ;SET LULP IN REG 12
031136  012701  003176                  MOV     #MSG1,R1        ;GET POINTER TO MSG DATA TABLE
031142  012137  002400          3$:     MOV     (R1)+,TXWORD    ;GET CHAR TO BE LOADED
031146  004737  005152                  JSR     PC,LDTXSI       ;LOAD CHAR INTO TX SILO
031152  020127  003224                  CMP     R1,#MSG1+22.    ;SEE IF ALL MSG CHARS LOADED YET
031156  103771                          BLO     3$              ;BR IF NOT YET
031160  004737  005124                  JSR     PC,WAIT50       ;ALLOW DATA TO RIPPLE IN SILO
031164  004737  005232                  JSR     PC,STPLU        ;CLOCK LU FOR 40 CYCLES (UNTIL FIRST
031170  000050                          40.                     ;   DATA CHAR IS ABOUT TO BE RECEIVED)
031172  004737  006672                  JSR     PC,IACTIV       ;CHK IACT = 0
031176  000000                          0
031200  004737  005232                  JSR     PC,STPLU        ;CLOCK LU UNTIL 1ST DATA CHAR IS RCV'D
031204  000006                          6
031206  012701  003202                  MOV     #MSG1+4,R1
031212  004737  006672          10$:    JSR     PC,IACTIV       ;CHK IACT = 1
031216  000001                          1
031220  004737  004102                  JSR     PC,READAX       ;READ AX0
031224  023721  002346                  CMP     RAX15,(R1)+     ;COMPARE RCV'D CHAR TO EXPECTED
031230  001415                          BEQ     12$             ;BR IF RCV'D DATA OK
031232  016137  177776  002362          MOV     -2(R1),GOODAT   ;GET EXPECTED DATA
031240  013737  002346  002364          MOV     RAX15,BADDAT    ;GET ACTUAL DATA
031246  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                        ;REPORT INCORRECT DATA CHAR RCV'D
031252                                  ERRDF   26,EM26,ERR3
031252  104455                                                                  TRAP    C$ERDF
031254  000032                                                                  .WORD   26
031256  013045                                                                  .WORD   EM26
031260  015116                                                                  .WORD   ERR3
031262  000461                          BR      24$
031264  004737  005232          12$:    JSR     PC,STPLU        ;CLOCK LU 8 CYCLES
031270  000010                          8.
031272  020127  003214                  CMP     R1,#MSG1+14.    ;SEE IF CHECKING HI CRC BYTE YET
031276  103745                          BLO     10$             ;BR IF NOT YET
031300  004737  006672                  JSR     PC,IACTIV       ;CHK IACT = 1
031304  000001                          1
031306  004737  004102                  JSR     PC,READAX       ;READ AX0
031312  123727  002346  000160          CMPB    RAX15,#160      ;CMP RCV'D CHAR TO EXPECTED HI CRC BYTE
```

```
7875  031320  001415                          BEQ     16$             ;BR IF HI CRC BYTE RCV'D OK
7876  031322  012737  000160  002362          MOV     #160,GOODAT     ;GET EXPECTED DATA
7877  031330  013737  002346  002364  14$:    MOV     RAX15,BADDAT    ;SET ACTUAL DATA
7878  031336  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7879                                  ;REPORT INCORRECT CRC BYTE RCV'D
7880  031342                                  ERRDF   27,EM27,ERR3
7881  031342  104455                                                                  TRAP    C$ERDF
7882  031344  000033                                                                  .WORD   27
7883  031346  013077                                                                  .WORD   EM27
7884  031350  015116                                                                  .WORD   ERR3
7885  031352  000425                          BR      24$
7886  031354  004737  005232          16$:    JSR     PC,STPLU        ;CLOCK LU FOR 8 CYCLES
7887  031360  000010                          8.
7888  031362  004737  006672                  JSR     PC,IACTIV       ;CHK IACT = 1
7889  031366  000001                          1
7890  031370  012737  000034  002362          MOV     #034,GOODAT     ;GET EXPECTED LO CRC BYTE
7891  031376  004737  004102                  JSR     PC,READAX       ;READ AX0
7892  031402  123727  002346  000034          CMPB    RAX15,#034      ;CMP RCV'D CHAR TO EXPECTED LO CRC BYTE
7893  031410  001347                          BNE     14$             ;BR IF LO CRC INCORRECT
7894  031412  004737  005232                  JSR     PC,STPLU        ;CLOCK LU 8 CYCLES
7895  031416  000010                          8.
7896  031420  004737  006672                  JSR     PC,IACTIV       ;CHK IACT STILL = 1
7897  031424  000001                          1
7898  031426  004737  003554          24$:    JSR     PC,MSTCLR       ;ISSUE CLEAN-UP MASTER CLEAR
7899  031432                                  ENDTST
7900  031432                                                                  L10062:
7901  031432  104401                                                                  TRAP    C$ETST
7902
7903
7904
7905
7906
7907                                  ;;**************************************************************************************
7908                                  .SBTTL      TEST 31 - USYRT RECEIVER MSG TEST - BIT MODE, CRC
7909                                  ;*
7910                                  ;* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
7911                                  ;* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH CRC-
7912                                  ;* CCITT-1. TWO FLAGS, 000,125,252,377,000, AND TWO TERMINATING FLAGS ARE THEN
7913                                  ;* SENT. THE PROGRAM MONITORS IACT, RSOM, AND THE RCV'D CHARS ARE READ
7914                                  ;* FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR
7915                                  ;* IACT = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT STILL = 0.
7916                                  ;;**************************************************************************************
7917  031434                          BGNTST
7918  031434                                                                  T31::
7919  031434  012737  031770  002340          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
7920  031442  004737  003554                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
7921  031446  004737  010616                  JSR     PC,SETUP        ;PROGRAM THE USYRT
7922  031452  000000                          000
7923  031454  000002                          IERR
7924  031456  000000                          000
7925  031460  000000                          000
7926  031462  012737  000012  002356          MOV     #12,REGNUM      ;SET LU REG NO. = 12
7927  031470  005037  002360                  CLR     AXNUM           ;SET AX BYTE NO. = 0 FOR AX0
7928  031474  112737  000040  002344          MOVB    #LULP,WRIBYT
7929  031502  004737  003726                  JSR     PC,WRITLU       ;SET LULP IN REG 12
7930  031506  012701  003176                  MOV     #MSG1,R1        ;GET POINTER TO MSG DATA TABLE
```

```
7931  031512  012137  002400          3$:     MOV     (R1)+,TXWORD    ;GET CHAR TO BE LOADED
7932  031516  004737  005152                  JSR     PC,LDTXSI       ;LOAD CHAR INTO TX SILO
7933  031522  020127  003220                  CMP     R1,#MSG1+18.    ;SEE IF ALL MSG CHARS LOADED YET
7934  031526  103771                          BLO     3$              ;BR IF NOT YET
7935  031530  004737  005124                  JSR     PC,WAIT50       ;ALLOW DATA TO RIPPLE IN SILO
7936  031534  004737  005232                  JSR     PC,STPLU        ;CLOCK LU FOR 50 CYCLES (UNTIL FIRST
7937  031540  000062                          50.                     ;   DATA CHAR IS ABOUT TO BE RECEIVED)
7938  031542  004737  006672                  JSR     PC,IACTIV       ;CHK IACT = 0
7939  031546  000000                          0
7940  031550  004737  007060                  JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0
7941  031554  000000                          0
7942  031556  004737  005232                  JSR     PC,STPLU        ;CLOCK LU UNTIL 1ST DATA CHAR IS RCV'D
7943  031562  000006                          6
7944  031564  012701  003202                  MOV     #MSG1+4,R1
7945  031570  020127  003202          5$:     CMP     R1,#MSG1+4      ;SEE IF 1ST CHAR RCV'D
7946  031574  001007                          BNE     6$              ;BR IF NO
7947  031576  004737  006672                  JSR     PC,IACTIV       ;CHK IACT = 1
7948  031602  000001                          1
7949  031604  004737  007060                  JSR     PC,RSEOM        ;CHK RSOM = 1, REOM = 0
7950  031610  000001                          1
7951  031612  000420                          BR      9$
7952  031614  020127  003212          6$:     CMP     R1,#MSG1+12.    ;SEE IF LAST CHAR RCV'D
7953  031620  001007                          BNE     8$              ;BR IF NO
7954  031622  004737  006672                  JSR     PC,IACTIV       ;CHK FOR IACT = 0
7955  031626  000000                          0
7956  031630  004737  007060                  JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0
7957  031634  000000                          0
7958  031636  000406                          BR      9$
7959  031640  004737  006672          8$:     JSR     PC,IACTIV       ;CHK FOR IACT = 1
7960  031644  000001                          1
7961  031646  004737  007060                  JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0
7962  031652  000000                          0
7963  031654  004737  004102          9$:     JSR     PC,READAX       ;READ AX0
7964  031660  023721  002346                  CMP     RAX15,(R1)+     ;COMPARE RCV'D CHAR TO EXPECTED
7965  031664  001415                          BEQ     12$             ;BR IF RCV'D DATA OK
7966  031666  016137  177776  002362          MOV     -2(R1),GOODAT   ;GET EXPECTED DATA
7967  031674  013737  002346  002364          MOV     RAX15,BADDAT    ;GET ACTUAL DATA
7968  031702  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7969                                  ;REPORT INCORRECT DATA CHAR RCV'D
7970  031706                                  ERRDF   26,EM26,ERR3
7971  031706  104455                                                                  TRAP    C$ERDF
7972  031710  000032                                                                  .WORD   26
7973  031712  013045                                                                  .WORD   EM26
7974  031714  015116                                                                  .WORD   ERR3
7975  031716  000424                          BR      24$
7976  031720  004737  005232          12$:    JSR     PC,STPLU        ;CLOCK LU 8 CYCLES
7977  031724  000010                          8.
7978  031726  020127  003214                  CMP     R1,#MSG1+14.    ;SEE IF ALL DATA CHARS CHECKED YET
7979  031732  103716                          BLO     5$              ;BR IF NOT YET
7980  031734  004737  006672                  JSR     PC,IACTIV       ;CHK IACT = 0
7981  031740  000000                          0
7982  031742  004737  007060                  JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0
7983  031746  000000                          0
7984  031750  012737  000200  002344          MOV     #IC,WRIBYT
7985  031756  004737  003726                  JSR     PC,WRITLU       ;SET IC (INPUT CLEAR) IN REG 12
7986  031762  004737  006672                  JSR     PC,IACTIV       ;CHK IACT = 0
```

```
                                        0
                24$:    JSR     PC,MSTCLR       ;ISSUE CLEAN-UP MASTER CLEAR
                ENDTST
                                                                        L10063:
                                                                                TRAP    C$ETST
                ;;*******************************************************************************
                .SBTTL      TEST 32 - USYRT RECEIVER MSG TEST - CHAR MODE, NO CRC
                ;*
                ;* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
                ;* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH NO
                ;* ERROR DETECTION. TWO SYNCHS, 000,125,252,377,000, AND TWO SYNCHS ARE
                ;* THEN SENT. THE PROGRAM MONITORS IACT, AND THE RECEIVED CHARS ARE READ FROM
                ;* AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR IACT
                ;* STILL = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT = 0.
                ;;*******************************************************************************
                BGNTST
                                                                        T32::
                        MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
                        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
                        JSR     PC,SETUP        ;PROGRAM THE USYRT
                        SYNCH
                        CRC2!CRC1!STRIP!IERR!DDCMP
                        000
                        000
                        MOV     #12,REGNUM      ;SET LU REG NO. = 12
                        CLR     AXNUM           ;SET AX BYTE NO. = 0 FOR AX0
                        MOVB    #LULP,WRIBYT
                        JSR     PC,WRITLU       ;SET LULP IN REG 12
                        MOV     #MSG1,R1        ;GET POINTER TO MSG DATA TABLE
                3$:     MOV     (R1)+,TXWORD    ;GET CHAR TO BE LOADED
                        JSR     PC,LDTXSI       ;LOAD CHAR INTO TX SILO
                        CMP     R1,#MSG1+18.    ;SEE IF ALL MSG CHARS LOADED YET
                        BLO     3$              ;BR IF NOT YET
                        JSR     PC,WAIT50       ;ALLOW DATA TO RIPPLE IN SILO
                        JSR     PC,STPLU        ;CLOCK LU FOR 24 CYCLES (UNTIL FIRST
                        24.                     ;   DATA CHAR IS ABOUT TO BE RECEIVED)
                        JSR     PC,IACTIV       ;CHK IACT = 0
                        0
                        JSR     PC,STPLU        ;CLOCK LU UNTIL 1ST DATA CHAR IS RCV'D
                        6
                        MOV     #MSG1+4,R1
                10$:    JSR     PC,IACTIV       ;CHK IACT = 1
                        1
                        JSR     PC,READAX       ;READ AX0
                        CMP     RAX15,(R1)+     ;COMPARE RCV'D CHAR TO EXPECTED
                        BEQ     12$             ;BR IF RCV'D DATA OK
                        MOV     -2(R1),GOODAT   ;GET EXPECTED DATA
                        MOV     RAX15,BADDAT    ;GET ACTUAL DATA
                        JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                ;REPORT INCORRECT DATA CHAR RCV'D
                        ERRDF   26,EM26,ERR3
```

```
8043  032164  104455                                                          TRAP    C$ERDF
8044  032166  000032                                                          .WORD   26
8045  032170  013045                                                          .WORD   EM26
8046  032172  015116                                                          .WORD   ERR3
8047  032174  000421                      BR      24$
8048  032176  004737  005232      12$:     JSR     PC,STPLU        ;CLOCK LU 8 CYCLES
8049  032202  000010                      8.
8050  032204  020127  003214              CMP     R1,#MSG1+14.    ;SEE IF ALL 5 DATA CHARS RCV'D YET
8051  032210  103745                      BLO     10$             ;BR IF NOT YET
8052  032212  004737  006672              JSR     PC,IACTIV       ;CHK FOR IACT STILL = 1
8053  032216  000001                      1
8054  032220  012737  000200  002344      MOV     #IC,WRIBYT
8055  032226  004737  003726              JSR     PC,WRITLU       ;SET IC (INPUT CLEAR) IN REG 12
8056  032232  004737  006672              JSR     PC,IACTIV       ;CHK IACT = 0
8057  032236  000000                      0
8058  032240  004737  003554      24$:     JSR     PC,MSTCLR       ;ISSUE CLEAN-UP MASTER CLEAR
8059  032244                      ENDTST
8060  032244                                                                  L10064:
8061  032244  104401                                                          TRAP    C$ETST
8062
8063
8064
8065
8066
8067                              ;;*************************************************************************
8068                              .SBTTL      TEST 33 - USYRT RECEIVER MSG TEST - BIT MODE, NO CRC
8069                              ;*
8070                              ;* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
8071                              ;* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH ERROR
8072                              ;* DETECTION INHIBITED. TWO FLAGS, 000,125,252,377,000, AND TWO TERMINATING FLAGS
8073                              ;* ARE THEN SENT. THE PROGRAM MONITORS IACT, RSOM, AND THE RCV'D CHARS ARE
8074                              ;* READ FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR
8075                              ;* IACT = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT STILL = 0.
8076                              ;;*************************************************************************
8077  032246                      BGNTST
8078  032246                                                                  T33::
8079  032246  012737  032602  002340          MOV     #24$,RETADR
8080  032254  004737  003554              JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
8081  032260  004737  010616              JSR     PC,SETUP        ;PROGRAM THE USYRT
8082  032264  000000                      000
8083  032266  000302                      CRC2!CRC1!IERR
8084  032270  000000                      000
8085  032272  000000                      000
8086  032274  012737  000012  002356      MOV     #12,REGNUM      ;SET LU REG NO. = 12
8087  032302  005037  002360              CLR     AXNUM           ;SET AX BYTE NO. = 0 FOR AX0
8088  032306  112737  000040  002344      MOVB    #LULP,WRIBYT
8089  032314  004737  003726              JSR     PC,WRITLU       ;SET LULP IN REG 12
8090  032320  012701  003176              MOV     #MSG1,R1        ;GET POINTER TO MSG DATA TABLE
8091  032324  012137  002400      3$:     MOV     (R1)+,TXWORD    ;GET CHAR TO BE LOADED
8092  032330  004737  005152              JSR     PC,LDTXSI       ;LOAD CHAR INTO TX SILO
8093  032334  020127  003220              CMP     R1,#MSG1+18.    ;SEE IF ALL MSG CHARS LOADED YET
8094  032340  103771                      BLO     3$              ;BR IF NOT YET
8095  032342  004737  005124              JSR     PC,WAIT50       ;ALLOW DATA TO RIPPLE IN SILO
8096  032346  004737  005232              JSR     PC,STPLU        ;CLOCK LU FOR 33 CYCLES (UNTIL FIRST
8097  032352  000041                      33.                     ;   DATA CHAR IS ABOUT TO BE RECEIVED)
8098  032354  004737  006672              JSR     PC,IACTIV       ;CHK IACT = 0
```

```
8099  032360  000000                          0
8100  032362  004737  007060                JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0
8101  032366  000000                          0
8102  032370  004737  005232                JSR     PC,STPLU        ;CLOCK LU UNTIL 1ST DATA CHAR IS RCV'D
8103  032374  000006                          6
8104  032376  012701  003202                MOV     #MSG1+4,R1
8105  032402  020127  003202        5$:     CMP     R1,#MSG1+4      ;SEE IF 1ST CHAR RCV'D
8106  032406  001007                        BNE     6$              ;BR IF NO
8107  032410  004737  006672                JSR     PC,IACTIV       ;CHK IACT = 1
8108  032414  000001                          1
8109  032416  004737  007060                JSR     PC,RSEOM        ;CHK RSOM = 1, REOM = 0
8110  032422  000001                          1
8111  032424  000420                        BR      9$
8112  032426  020127  003212        6$:     CMP     R1,#MSG1+12.    ;SEE IF LAST CHAR RCV'D
8113  032432  001007                        BNE     8$              ;BR IF NO
8114  032434  004737  006672                JSR     PC,IACTIV       ;CHK FOR IACT = 0
8115  032440  000000                          0
8116  032442  004737  007060                JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0
8117  032446  000000                          0
8118  032450  000406                        BR      9$
8119  032452  004737  006672        8$:     JSR     PC,IACTIV       ;CHK FOR IACT = 1
8120  032456  000001                          1
8121  032460  004737  007060                JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0
8122  032464  000000                          0
8123  032466  004737  004102        9$:     JSR     PC,READAX       ;READ AX0
8124  032472  023721  002346                CMP     RAX15,(R1)+     ;COMPARE RCV'D CHAR TO EXPECTED
8125  032476  001415                        BEQ     12$             ;BR IF RCV'D DATA OK
8126  032500  016137  177776  002362        MOV     -2(R1),GOODAT   ;GET EXPECTED DATA
8127  032506  013737  002346  002364        MOV     RAX15,BADDAT    ;GET ACTUAL DATA
8128  032514  004737  004504                JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
8129                                ;REPORT INCORRECT DATA CHAR RCV'D
8130  032520                                ERRDF   26,EM26,ERR3
8131  032520  104455                                                                TRAP    C$ERDF
8132  032522  000032                                                                .WORD   26
8133  032524  013045                                                                .WORD   EM26
8134  032526  015116                                                                .WORD   ERR3
8135  032530  000424                        BR      24$
8136  032532  004737  005232        12$:    JSR     PC,STPLU        ;CLOCK LU 8 CYCLES
8137  032536  000010                          8.
8138  032540  020127  003214                CMP     R1,#MSG1+14.    ;SEE IF ALL DATA CHARS CHECKED YET
8139  032544  103716                        BLO     5$              ;BR IF NOT YET
8140  032546  004737  006672                JSR     PC,IACTIV       ;CHK IACT = 0
8141  032552  000000                          0
8142  032554  004737  007060                JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0
8143  032560  000000                          0
8144  032562  012737  000200  002344        MOV     #IC,WRIBYT
8145  032570  004737  003726                JSR     PC,WRITLU       ;SET IC (INPUT CLEAR) IN REG 12
8146  032574  004737  006672                JSR     PC,IACTIV       ;CHK IACT = 0
8147  032600  000000                          0
8148  032602  004737  003554        24$:    JSR     PC,MSTCLR       ;ISSUE CLEAN-UP MASTER CLEAR
8149  032606                                ENDTST
8150  032606                                                                  L10065:
8151  032606  104401                                                                TRAP    C$ETST
8152
8153
8154
```

```
                                ;;*************************************************************************************
                                .SBTTL      TEST 34 - SILO-DISABLED TRANSMITTER LOAD TEST
                                ;*
                                ;* THIS TEST DISABLES THE SILOS, LOADS A 125 CHARACTER INTO THE TX SILO, AND
                                ;* READS AX1-15 AND CHECKS THAT THE DATA DID NOT GET LOADED INTO THE USYRT TX
                                ;* BUFFER.
                                ;;*************************************************************************************
032610                          BGNTST
                                                                                        T34::
032610  004737  003554                        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
032614  012737  000014  002356                MOV     #14,REGNUM      ;SET REG NO. = 14
032622  012737  000040  002344                MOV     #DISSI,WRIBYT   ;SET DISSI BIT
032630  004737  003726                        JSR     PC,WRITLU
032634  012737  000040  002404                MOV     #DISSI,DISILO   ;SET DISABLE SILO FLAG
032642  012737  000125  002400                MOV     #125,TXWORD     ;LOAD 125 INTO TX SILO
032650  004737  005152                        JSR     PC,LDTXSI
032654  012737  000002  002360                MOV     #2,AXNUM        ;SET REG NO. FOR AX1
032662  004737  004102                        JSR     PC,READAX       ;READ AX1-15, AX1-16
032666  123727  002346  000000                CMPB    RAX15,#000      ;CHECK FOR AX1-15 UNCHANGED
032674  001414                              BEQ     3$              ;BR IF UNCHANGED
032676  012737  000000  002362                MOV     #000,GOODAT     ;GET EXPECTED DATA
032704  013737  002346  002364                MOV     RAX15,BADDAT    ;GET ACTUAL DATA
032712  004737  004504                        JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                ;REPORT REG MISCOMPARE
032716                                      ERRDF   3,EM3,ERR3
032716  104455                                                                  TRAP    C$ERDF
032720  000003                                                                  .WORD   3
032722  012176                                                                  .WORD   EM3
032724  015116                                                                  .WORD   ERR3
032726  005037  002404          3$:         CLR     DISILO          ;CLEAR DISABLE SILO FLAG
032732  004737  003554                        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
032736                          ENDTST
032736                                                                      L10066:
032736  104401                                                                  TRAP    C$ETST
                                ;;*************************************************************************************
                                .SBTTL      TEST 35 - SILO-DISABLED MESSAGE TEST - BIT MODE, NO CRC
                                ;*
                                ;* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU)
                                ;* WITH LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, WITH SILO
                                ;* DISABLE SET, AND WITH NO ERROR DETECTION. TWO FLAGS, 000,125,252, AND
                                ;* TERMINATING FLAGS ARE THEN SENT BY LOADING THE TRANSMITTED CHARS INTO
                                ;* REG AX1. THE PROGRAM MONITORS OACT, IACT, RSOM, REOM, ORDY, OCOR, ICIR,
                                ;* IRDY, AND THE RECEIVED CHARS ARE READ FROM AX0 AND COMPARED TO EXPECTED
                                ;* VALUES.
                                ;;*************************************************************************************
032740                          BGNTST
                                                                                        T35::
032740  012737  033640  002340                MOV     #18$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
032746  004737  005516                        JSR     PC,INITRN       ;FIND OUT WHICH USYRT CHIP
```

```
8211  032752  000000                          0
8212  032754  000000                          0
8213  032756  004737  003554                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
8214  032762  004737  010616                  JSR     PC,SETUP        ;PROGRAM THE USYRT
8215  032766  000000                          000
8216  032770  000302                          CRC2!CRC1!IERR
8217  032772  000000                          000
8218  032774  000000                          000
8219  032776  012737  000014  002356          MOV     #14,REGNUM      ;SET REG NO. = 14
8220  033004  012737  000140  002344          MOV     #TXEN!DISSI,WRIBYT
8221  033012  004737  003726                  JSR     PC,WRITLU       ;SET TXEN AND DISSI IN REG 14
8222  033016  012737  000140  002404          MOV     #TXEN!DISSI,DISILO      ;SET DISABLE SILO FLAG
8223  033024  012737  000012  002356          MOV     #12,REGNUM      ;SET LU REG NO. = 12
8224  033032  112737  000040  002344          MOVB    #LULP,WRIBYT
8225  033040  004737  003726                  JSR     PC,WRITLU       ;SET LULP IN REG 12
8226  033044  012701  003176                  MOV     #MSG1,R1        ;GET POINTER TO MSG
8227  033050  004737  004640                  JSR     PC,OSIRDY       ;CHK ORDY = 1
8228  033054  000001                          1
8229  033056  012737  000002  002360          MOV     #2,AXNUM        ;SET AX BYTE NO. FOR AX1
8230  033064  112137  002352                  MOVB    (R1)+,WAX15     ;GET A CHAR
8231  033070  112137  002354                  MOVB    (R1)+,WAX16
8232  033074  004737  004270                  JSR     PC,WRITAX       ;LOAD CHAR INTO USYRT TX BUFFER
8233  033100  004737  004640                  JSR     PC,OSIRDY       ;CHK ORDY = 0
8234  033104  000000                          0
8235  033106  004737  005330                  JSR     PC,OACTIV       ;CHK OACT = 0
8236  033112  000000                          0
8237  033114  004737  005232                  JSR     PC,STPLU        ;CLOCK LU FOR 3 CYCLES
8238  033120  000003                          3
8239  033122  004737  005330                  JSR     PC,OACTIV       ;CHK OACT = 1
8240  033126  000001                          1
8241  033130  012703  000004                  MOV     #4,R3           ;INIT COUNTER
8242  033134  112137  002352          4$:     MOVB    (R1)+,WAX15     ;GET ANOTHER CHAR
8243  033140  112137  002354                  MOVB    (R1)+,WAX16
8244  033144  020327  000001                  CMP     R3,#1           ;SEE IF LOADING LAST DATA CHAR YET
8245  033150  001006                          BNE     5$              ;BR IF NOT
8246  033152  005737  002406                  TST     CHPTYP          ;SEE IF SIG USYRT
8247  033156  001403                          BEQ     5$              ;BR IF YES
8248  033160  112737  000002  002354          MOVB    #TEOM,WAX16     ;SET TEOM WITH LAST DATA CHAR
8249  033166  004737  004640          5$:     JSR     PC,OSIRDY       ;CHK ORDY = 1
8250  033172  000001                          1
8251  033174  004737  004270                  JSR     PC,WRITAX       ;LOAD ANOTHER CHAR INTO USYRT TX BUFFER
8252  033200  004737  004640                  JSR     PC,OSIRDY       ;CHK ORDY = 0
8253  033204  000000                          0
8254  033206  004737  006672                  JSR     PC,IACTIV       ;CHK IACT = 0
8255  033212  000000                          0
8256  033214  004737  007060                  JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0
8257  033220  000000                          0
8258  033222  004737  005232                  JSR     PC,STPLU        ;CLOCK LU FOR 8 CYCLES
8259  033226  000010                          8.
8260  033230  004737  005330                  JSR     PC,OACTIV       ;CHK OACT = 1
8261  033234  000001                          1
8262  033236  004737  004640                  JSR     PC,OSIRDY       ;CHK ORDY = 1
8263  033242  000001                          1
8264  033244  005303                          DEC     R3              ;DECR COUNTER
8265  033246  001332                          BNE     4$              ;BR IF NOT DONE YET
3266  033250  004737  006406                  JSR     PC,ISIRDY       ;CHK IRDY = 0
```

```
 8267  033254  000000                          0
 8268  033256  005737  002406                  TST     CHPTYP          ;SEE IF SIG USYRT
 8269  033262  001007                          BNE     11$             ;BR IF NOT
 8270  033264  105037  002352                  CLRB    WAX15
 8271  033270  112737  000002  002354          MOVB    #TEOM,WAX16     ;LOAD EOM CHAR
 8272  033276  004737  004270                  JSR     PC,WRITAX       ;LOAD ANOTHER CHAR INTO USYRT TX BUFFER
 8273  033302  004737  005232          11$:    JSR     PC,STPLU        ;CLOCK LU FOR 3 CYCLES
 8274  033306  000003                          3
 8275  033310  004737  006406                  JSR     PC,ISIRDY       ;CHK IRDY = 1
 8276  033314  000002                          2
 8277  033316  004737  005330                  JSR     PC,OACTIV       ;CHK OACT = 1
 8278  033322  000001                          1
 8279  033324  004737  006672                  JSR     PC,IACTIV       ;CHK IACT = 1
 8280  033330  000001                          1
 8281  033332  004737  007060                  JSR     PC,RSEOM        ;CHK RSOM = 1, REOM = 0
 8282  033336  000001                          1
 8283  033340  004737  006406                  JSR     PC,ISIRDY       ;CHK IRDY = 0
 8284  033344  000000                          0
 8285  033346  012737  000000  002360          MOV     #0,AXNUM        ;SET AX BYTE NO. FOR AX0
 8286  033354  123727  002346  000000          CMPB    RAX15,#000      ;COMPARE RCV'D CHAR TO 000
 8287  033362  001415                          BEQ     9$              ;BR IF MATCH
 8288  033364  012737  000000  002362          MOV     #0,GOODAT       ;SET EXPECTED DATA
 8289  033372  013737  002346  002364  6$:     MOV     RAX15,BADDAT    ;SET ACTUAL DATA
 8290  033400  004737  004504                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 8291                                  ;REPORT INCORRECT DATA CHAR RCV'D
 8292  033404                                  ERRDF   26,EM26,ERR3
 8293  033404  104455                                                                  TRAP    C$ERDF
 8294  033406  000032                                                                  .WORD   26
 8295  033410  013045                                                                  .WORD   EM26
 8296  033412  015116                                                                  .WORD   ERR3
 8297  033414  000511                          BR      18$
 8298  033416  004737  005232          9$:     JSR     PC,STPLU        ;CLOCK LU FOR 8 CYCLES
 8299  033422  000010                          8.
 8300  033424  004737  006406                  JSR     PC,ISIRDY       ;CHK IRDY = 1
 8301  033430  000002                          2
 8302  033432  004737  005330                  JSR     PC,OACTIV       ;CHK OACT = 1
 8303  033436  000001                          1
 8304  033440  004737  006672                  JSR     PC,IACTIV       ;CHK IACT = 1
 8305  033444  000001                          1
 8306  033446  004737  007060                  JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0
 8307  033452  000000                          0
 8308  033454  004737  006406                  JSR     PC,ISIRDY       ;CHK IRDY = 0
 8309  033460  000000                          0
 8310  033462  123727  002346  000125          CMPB    RAX15,#125      ;COMPARE 2ND RCV'D CHAR TO 125
 8311  033470  001404                          BEQ     12$             ;BR IF MATCH
 8312  033472  012737  000125  002362          MOV     #125,GOODAT     ;SET EXPECTED DATA
 8313  033500  000734                          BR      6$              ;BR TO REPORT ERROR
 8314  033502  004737  005232          12$:    JSR     PC,STPLU        ;CLOCK LU FOR 8 CYCLES
 8315  033506  000010                          8.
 8316  033510  004737  006406                  JSR     PC,ISIRDY       ;CHK IRDY = 1
 8317  033514  000002                          2
 8318  033516  004737  005330                  JSR     PC,OACTIV       ;CHK OACT = 1
 8319  033522  000001                          1
 8320  033524  004737  006672                  JSR     PC,IACTIV       ;CHK IACT = 0
 8321  033530  000000                          0
 8322  033532  004737  007060                  JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 1
```

```
 8323  033536  000002                          2
 8324  033540  004737  006406                  JSR     PC,ISIRDY       ;CHK IRDY = 0
 8325  033544  000000                          0
 8326  033546  123727  002346  000252          CMPB    RAX15,#252      ;COMPARE 3RD RCV'D CHAR TO 252
 8327  033554  001404                          BEQ     14$             ;BR IF MATCH
 8328  033556  012737  000252  002362          MOV     #252,GOODAT     ;SET EXPECTED DATA
 8329  033564  000702                          BR      6$              ;BR TO REPORT ERROR
 8330  033566  012737  000014  002356  14$:    MOV     #14,REGNUM      ;SET REG NO. = 14
 8331  033574  012737  000040  002344          MOV     #DISSI,WRIBYT
 8332  033602  004737  003726                  JSR     PC,WRITLU       ;CLEAR TX ENABLE
 8333  033606  012737  000011  002356          MOV     #11,REGNUM      ;SET REG NO. = 11
 8334  033614  012737  000200  002344          MOV     #OC,WRIBYT
 8335  033622  004737  003726                  JSR     PC,WRITLU       ;SET OC TO SHUT DOWN TRANSMITTER
 8336  033626  004737  005124                  JSR     PC,WAIT50       ;WAIT FOR SHUTDOWN
 8337  033632  004737  005330                  JSR     PC,OACTIV       ;CHK FOR OACT = 0
 8338  033636  000000                          0
 8339  033640  005037  002404          18$:    CLR     DISILO          ;CLEAR DISABLE SILO FLAG
 8340  033644  004737  003554                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
 8341  033650                          ENDTST
 8342  033650                                                                  L10067:
 8343  033650  104401                                                  ,               TRAP    C$ETST
 8344
 8345
 8346
 8347
 8348
 8349                                  ;;*************************************************************************************
 8350                                  .SBTTL      TEST 36 - RECEIVER BUFFER TEST - CHAR MODE, CRC
 8351                                  ;*
 8352                                  ;* FIRST, A MASTER CLEAR IS DONE AND THE PROGRAM CHECKS FOR ICIR = 1 AND IRDY
 8353                                  ;* = 0. THEN, 2 SOM CHARS ARE LOADED AND CLOCKED INTO THE USYRT, AND 64
 8354                                  ;* BYTES OF A 256-BYTE BINARY COUNT DATA PATTERN (000-377) ARE LOADED INTO
 8355                                  ;* THE TX SILO.
 8356                                  ;* THE LINE UNIT IS THEN CLOCKED UNTIL IRDY = 1, AND THE PROGRAM CHECKS FOR
 8357                                  ;* THIS TO OCCUR WITHIN 40-43 CYCLES. THE PROGRAM READS THE RCV SILO, CHECKS THE
 8358                                  ;* CHAR FOR 000, AND CHECKS FOR IRDY = 0 AGAIN.
 8359                                  ;* THE LINE UNIT IS THEN CLOCKED IN GROUPS OF 8 CYCLES, AND AFTER EACH, THE
 8360                                  ;* PROGRAM CHECKS FOR ICIR = 1, IRDY = 1, UNTIL THE 64TH GROUP, AFTER WHICH
 8361                                  ;* IT CHECKS FOR ICIR = 0, IRDY = 1. THE SECOND DATA CHAR IS READ FROM THE
 8362                                  ;* RECEIVER SILO AND COMPARED TO 001. THEN, THE PROGRAM CHECKS FOR ICIR = 1,
 8363                                  ;* IRDY = 1 AGAIN.
 8364                                  ;* THE REST OF THE BINARY COUNT DATA BYTES ARE CYCLED 8 CLOCKS AND READ AND
 8365                                  ;* COMPARED A BYTE AT A TIME.
 8366                                  ;*************************************************************************************
 8367  033652                          BGNTST
 8368  033652                                                                  T36::
 8369  033652  012737  034126  002340          MOV     #A10,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
 8370                                  ;-------------------------------------------------------------------------
 8371                                  ; DO MASTER CLR, CHK FOR ICIR = 1, IRDY = 0
 8372                                  ;-------------------------------------------------------------------------
 8373  033660  004737  003554                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 8374  033664  004737  006406                  JSR     PC,ISIRDY       ;CHK ICIR = 1, IRDY = 0
 8375  033670  000001                          1
 8376                                  ;-------------------------------------------------------------------------
 8377                                  ; LOAD AND CLOCK 2 SOM'S, LOAD 64 BYTES OF BINARY COUNT PATTERN INTO TX SILO,
 8378                                  ; CLOCK LINE UNIT, CHK FOR IRDY = 1 WITHIN 40-43 CYCLES
```

CZDMRF.P11 03-NOV-81 10:29 TEST 36 - RECEIVER BUFFER TEST - CHAR MODE, CRC SEQ 164

```
8379
8380  033672  004737  005516          ;----------------------------------------------------------------------
8381  033676  000226                  JSR     PC,INITRN       ;LOAD 2 SOM'S, CLOCK THEM INTO USYRT
8382  033700  000011                  SYNCH
8383  033702  005003                  STRIP!DDCMP
8384  033704  010337  002400          CLR     R3              ;INIT BINARY COUNT DATA FOR WRITING
8385  033710  004737  005152  2$:     MOV     R3,TXWORD
8386  033714  005203                  JSR     PC,LDTXSI       ;LOAD A DATA BYTE INTO TX SILO
8387  033716  020327  000100          INC     R3              ;INCR DATA
8388  033722  002770                  CMP     R3,#64.         ;SEE IF 64 BYTES LOADED YET
8389  033724  004737  007412          BLT     2$              ;BR IF NOT YE
8390  033730  000050                  JSR     PC,RCV1ST       ;RECEIVE AND TIME FIRST CHARACTER
8391                                  40.
8392                                  ;----------------------------------------------------------------
8393                                  ; READ RCV SILO, COMPARE FIRST CHAR TO 000, CHK FOR ICIR = 1, IRDY = 0
8394  033732  005004                  ;----------------------------------------------------------------
8395  033734  004737  007726  9$:     CLR     R4              ;INIT PATTERN FOR READING
8396  033740  000000                  JSR     PC,CKDATA       ;READ RCV SILO, COMPARE DATA
8397  033742  000000                  0                       ;EXPECTED DATA = 000
8398  033744  005204                  0                       ;DON'T CLOCK LINE UNIT
8399  033746  004737  006406  16$:    INC     R4              ;INCR DATA FOR READING
8400  033752  000001                  JSR     PC,ISIRDY       ;CHK FOR ICIR = 1, IRDY = 0
8401                                  1
8402                                  ;----------------------------------------------------------------
8403                                  ; CLOCK 63 CHARS INTO RCV SILO, CHK ICIR = 1, IRDY = 1
8404  033754  004737  005232          ;----------------------------------------------------------------
8405  033760  000010          18$:    JSR     PC,STPLU        ;CLOCK LU FOR 8 CYCLES
8406  033762  010337  002400          8.
8407  033766  004737  005152          MOV     R3,TXWORD
8408  033772  005203                  JSR     PC,LDTXSI       ;LOAD ANOTHER WORD INTO TX SILO
8409  033774  004737  006406          INC     R3              ;INCR PATTERN FOR WRITING
8410  034000  000003                  JSR     PC,ISIRDY       ;CHK ICIR = 1, IRDY = 1
8411  034002  020327  000177          3
8412  034006  002762                  CMP     R3,#127.        ;SEE IF 63 MORE CHARS CLOCKED YET
8413                                  BLT     18$             ;BR IF NOT YET
8414                                  ;----------------------------------------------------------------
8415                                  ; CLOCK 1 MORE CHAR INTO RCV SILO, CHK ICIR = 0, IRDY = 1
8416  034010  004737  005232          ;----------------------------------------------------------------
8417  034014  000010                  JSR     PC,STPLU        ;CLOCK LU FOR 8 CYCLES
8418  034016  004737  006406          8.
8419  034022  000002                  JSR     PC,ISIRDY       ;CHK ICIR = 0, IRDY = 1
8420                                  2
8421                                  ;----------------------------------------------------------------
8422                                  ;READ, COMPARE, CLOCK REST OF DATA CHARS
8423  034024  010437  034034          ;----------------------------------------------------------------
8424  034030  004737  007726  20$:    MOV     R4,21$          ;SET EXPECTED DATA
8425  034034  000000                  JSR     PC,CKDATA       ;READ AND COMPARE DATA
8426  034036  000000          21$:    0                       ;EXPECTED SILO ENTRY GOES HERE
8427  034040  005204                  0                       ;DON'T CLOCK LINE UNIT
8428  034042  020427  000400  22$:    INC     R4              ;INCR DATA PATTERN FOR READS
8429  034046  001427                  CMP     R4,#400         ;SEE IF ALL DONE READING YET
8430  034050  004737  006406          BEQ     32$             ;BR IF DONE READING
8431  034054  000003                  JSR     PC,ISIRDY       ;CHK ICIR = 1, IRDY = 1
8432  034056  004737  005232          3
8433  034062  000010                  JSR     PC,STPLU        ;CLOCK LU FOR 8 CYCLES
8434  034064  020327  000377          8.
                                      CMP     R3,#377         ;SEE IF ALL CHARS LOADED INTO TX SILO YET
```

```
                                        BGT     24$             ;BR IF YES
                                        MOV     R3,TXWORD
                                        JSR     PC,LDTXSI       ;LOAD ANOTHER CHAR INTO TX SILO
                                        INC     R3              ;INCR DATA PATTERN FOR WRITING
                                        JMP     20$
                                24$:    MOV     #TXEOM,TXWORD
                                        JSR     PC,LDTXSI       ;LOAD EOM INTO TX SILO
                                        JMP     20$
                                32$:
                                A10:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
                                ENDTST
                                                                                L10070:
                                                                                        TRAP    C$ETST
;;*************************************************************************************
.SBTTL      TEST 37 - RECEIVER CHAR LENGTH TIMING TEST - CHAR MODE, NO CRC
;*
;* THE LINE UNIT IS PLACED IN CHAR MODE, WITH NO ERROR DETECTION, AND A MSG IS
;* INITIATED WITH 2 SYNCH CHARS. NEXT, FIFTEEN 000 CHARS ARE LOADED INTO THE
;* TRANSMITTER SILO. THE LINE UNIT IS THEN CLOCKED USING STEPLU WITH LULOOP
;* SET, WHILE THE RECEIVER CHAR LENGTH IS SET TO THE FOLLOWING VALUES : 5,6,7,8.
;* FOR EACH RCV CHAR LENGTH, THE PROGRAM CHECKS TO MAKE SURE THET USYRT RECEIVER
;* FLAGS OCCUR THE PROPER NO. OF CYCLES APART, FOR EACH RCV CHAR LENGTH.
;* (FOR EXAMPLE A 5-BIT CHAR TAKES 5 CLOCK CYCLES TO BE RECEIVED). A MASTER
;* CLEAR IS THEN DONE TO TERMINATE THE OPERATION.
;;*************************************************************************************
BGNTST
                                                                                T37::
                                        MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
                                        JSR     PC,INITRN       ;DO MASTER CLR, LOAD 2 SOM'S
                                        000
                                        CRC2!CRC1!IDLE!DDCMP
                                        MOV     #15.,R1         ;INIT COUNTER
                                        CLR     TXWORD
                                3$:     JSR     PC,LDTXSI       ;LOAD A 000 CHAR INTO TX SILO
                                        DEC     R1              ;DECR COUNTER
                                        BNE     3$              ;BR IF NOT DONE LOADING YET
                                        JSR     PC,WAIT50       ;WAIT FOR SILO TO RIPPLE
                                        MOV     #6,AXNUM        ;SET BYTE NO. = 6 FOR AX3
                                        MOV     #000,WAX15      ;SET DATA FOR AX3-15 = 0
                                        MOV     #RXLEN2!RXLEN0,WAX16 ;SET RCV LEN = 5
                                        JSR     PC,WRITAX       ;LOAD AX3
                                        JSR     PC,STPLU        ;CLK LU UNTIL TX'ING 1ST DATA CHAR
                                        CHPCHK!10.
                                        MOV     #RXLEN2!RXLEN1,WAX16 ;SET RCV LEN = 6
                                        JSR     PC,WRITAX       ;LOAD AX3
                                        JSR     PC,RCV1ST       ;CLOCK 5-BIT DATA CHAR
                                        5
                                        MOV     #RXLEN2!RXLEN1!RXLEN0,WAX16 ;SET RCV LEN = 7
                                        JSR     PC,WRITAX       ;LOAD AX3
                                        JSR     PC,RXCHAR       ;RCV 5-BIT DATA CHAR, CLK 6-BIT
                                        6
```

```
 8491  034272  012737  000000  002354          MOV     #0,WAX16        ;SET RCV LEN = 8
 8492  034300  004737  004270                  JSR     PC,WRITAX       ;LOAD AX3
 8493  034304  004737  011052                  JSR     PC,RXCHAR       ;RCV 6-BIT DATA CHAR, CLK 7-BIT
 8494  034310  000007                          7
 8495  034312  004737  011052                  JSR     PC,RXCHAR       ;RCV 7-BIT DATA CHAR, CLK 8-BIT
 8496  034316  000010                          8.
 8497  034320  004737  011052                  JSR     PC,RXCHAR       ;RCV 8-BIT DATA CHAR
 8498  034324  000010                          8.
 8499  034326  004737  003554          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
 8500  034332                          ENDTST
 8501  034332                                                                  L10071:
 8502  034332  104401                                                                  TRAP    C$ETST
 8503
 8504
 8505
 8506
 8507
 8508                                  ;;*************************************************************************
 8509                                  .SBTTL      TEST 38 - RECEIVER CHAR LENGTH TIMING TEST - BIT MODE, NO CRC
 8510                                  ;*
 8511                                  ;* THE LINE UNIT IS PLACED IN BIT MODE WITH NO ERROR DETECTION, AND A MESSAGE IS
 8512                                  ;* INITIATED  WITH 2 FLAG CHARS. NEXT, FIFTEEN 000 CHARS ARE LOADED INTO THE
 8513                                  ;* TRANSMITTER SILO. THE LINE UNIT IS THEN CLOCKED USING STEPLU WITH LULOOP
 8514                                  ;* SET. WHILE THE RCV CHAR LENGTH IS SET TO THE FOLLOWING VALUES : 8,8,8,7,6,5,
 8515                                  ;* 4,3,2,1. FOR EACH RCV CHAR LENGTH, THE PROGRAM CHECKS TO MAKE SURE THAT THE
 8516                                  ;* USYRT RECEIVER FLAGS OCCUR THE PROPER NO. OF CYCLES APART, FOR EACH RCV
 8517                                  ;* CHAR LENGTH. (FOR EXAMPLE, A 5 BIT CHAR TAKES 5 CLOCK CYCLES TO BE RECEIVED).
 8518                                  ;* A MASTER CLEAR IS THEN DONE TO TERMINATE THE OPERATION.
 8519                                  ;;*************************************************************************
 8520  034334                          BGNTST
 8521  034334                                                                  T38::
 8522  034334  012737  034626  002340          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
 8523  034342  004737  005516                  JSR     PC,INITRN       ;DO MASTER CLR, LOAD 2 SOM'S
 8524  034346  000000                          000
 8525  034350  000300                          CRC2!CRC1
 8526  034352  012701  000017                  MOV     #15.,R1         ;INIT COUNTER
 8527  034356  005037  002400                  CLR     TXWORD
 8528  034362  004737  005152          3$:     JSR     PC,LDTXSI       ;LOAD A 000 CHAR INTO TX SILO
 8529  034366  005301                          DEC     R1              ;DECR COUNTER
 8530  034370  001374                          BNE     3$              ;BR IF NOT DONE LOADING YET
 8531  034372  004737  005124                  JSR     PC,WAIT50       ;WAIT FOR SILO TO RIPPLE
 8532  034376  012737  000006  002360          MOV     #6,AXNUM        ;SET BYTE NO. = 6 FOR AX3
 8533  034404  012737  000000  002352          MOV     #000,WAX15      ;SET DATA FOR AX3-15 = 0
 8534  034412  004737  007412                  JSR     PC,RCV1ST       ;CLOCK FIRST 8-BIT DATA CHAR
 8535  034416  000040                          32.
 8536  034420  012737  000000  002354          MOV     #0,WAX16        ;SET RCV LEN = 8
 8537  034426  004737  004270                  JSR     PC,WRITAX       ;LOAD AX3
 8538  034432  004737  011052                  JSR     PC,RXCHAR       ;RCV FIRST 8-BIT DATA CHAR, CLK SECOND 8-BIT
 8539  034436  000010                          8.
 8540  034440  012737  000007  002354          MOV     #RXLEN2!RXLEN1!RXLEN0,WAX16 ;SET RCV LEN = 7
 8541  034446  004737  004270                  JSR     PC,WRITAX       ;LOAD AX3
 8542  034452  004737  011052                  JSR     PC,RXCHAR       ;RCV SECOND 8-BIT DATA CHAR, CLK 3RD 8-BIT
 8543  034456  000010                          8.
 8544  034460  012737  000006  002354          MOV     #RXLEN2!RXLEN1,WAX16    ;SET RCV LEN = 6
 8545  034466  004737  004270                  JSR     PC,WRITAX       ;LOAD AX3
 8546  034472  004737  011052                  JSR     PC,RXCHAR       ;RCV 3RD 8-BIT DATA CHAR, CLK 7-BIT
```

```
8547  034476  000007                          7
8548  034500  012737  000005  002354          MOV     #RXLEN2!RXLEN0,WAX16    ;SET RCV LEN = 5
8549  034506  004737  004270                  JSR     PC,WRITAX       ;LOAD AX3
8550  034512  004737  011052                  JSR     PC,RXCHAR       ;RCV 7-BIT DATA CHAR, CLK 6-BIT
8551  034516  000006                          6
8552  034520  012737  000004  002354          MOV     #RXLEN2,WAX16   ;SET RCV LEN = 4
8553  034526  004737  004270                  JSR     PC,WRITAX       ;LOAD AX3
8554  034532  004737  011052                  JSR     PC,RXCHAR       ;RCV 6-BIT DATA CHAR, CLK 5-BIT
8555  034536  000005                          5
8556  034540  012737  000003  002354          MOV     #RXLEN1!RXLEN0,WAX16    ;SET RCV LEN = 3
8557  034546  004737  004270                  JSR     PC,WRITAX       ;LOAD AX3
8558  034552  004737  011052                  JSR     PC,RXCHAR       ;RCV 5-BIT DATA CHAR, CLR 4-BIT
8559  034556  000004                          4
8560  034560  012737  000002  002354          MOV     #RXLEN1,WAX16   ;SET RCV LEN = 2
8561  034566  004737  004270                  JSR     PC,WRITAX       ;LOAD AX3
8562  034572  004737  011052                  JSR     PC,RXCHAR       ;RCV 4-BIT DATA CHAR, CLK 3-BIT
8563  034576  000003                          3
8564  034600  012737  000001  002354          MOV     #RXLEN0,WAX16   ;SET RCV LEN = 1
8565  034606  004737  004270                  JSR     PC,WRITAX       ;LOAD AX3
8566  034612  004737  011052                  JSR     PC,RXCHAR       ;RCV 3-BIT DATA CHAR, CLK 2-BIT
8567  034616  000002                          2
8568  034620  004737  011052                  JSR     PC,RXCHAR       ;RCV 2-BIT DATA CHAR, CLK 1-BIT
8569  034624  000001                          1
8570  034626  004737  003554          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
8571  034632                          ENDTST
8572  034632                                                                  L10072:
8573  034632  104401                                                                  TRAP    C$ETST
8574
8575
8576
8577
8578
8579                                  ;;*************************************************************************
8580                                  .SBTTL      TEST 39 - TRANSMITTER UNDERRUN ERROR, IDLE MARKING, CHAR MODE,NO CRC
8581                                  ;*
8582                                  ;* THE LINE UNIT IS PLACED IN CHAR MODE, AND THE IDLE BIT IS SET. THEN, A
8583                                  ;* MSG IS INITIATED, A 000 CHAR IS SENT, AND THE TX BUFFER IS NOT SERVICED
8584                                  ;* IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG, WHICH CAUSES A TX UNDERRUN
8585                                  ;* ERROR. THEN, THE RECEIVER IS CLOCKED AND CHECKED FOR TWO 377 CHARS TO BE
8586                                  ;* RECEIVED (LINE MARKING) BEFORE SHUTTING DOWN WITH A MASTER CLEAR.
8587                                  ;;*************************************************************************
8588  034634                          BGNTST
8589  034634                                                                  T39::
8590  034634  012737  034716  002340          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
8591  034642  004737  005516                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
8592  034646  000226                          SYNCH
8593  034650  000341                          CRC2!CRC1!IDLE!DDCMP
8594  034652  012737  000000  002400          MOV     #000,TXWORD
8595  034660  004737  005152                  JSR     PC,LDTXSI       ;LOAD 000 CHAR INTO TX SILO
8596  034664  004737  005232                  JSR     PC,STPLU        ;CLOCK LINE UNIT UNTIL LINE GOES MARKING
8597  034670  000063                          51.
8598  034672  004737  007332                  JSR     PC,RDRXSI       ;READ 000 CHAR
8599  034676  004737  007726                  JSR     PC,CKDATA       ;READ AND CHECK FOR MARK CHAR (377)
8600  034702  000377                          377
8601  034704  000000                          000
8602  034706  004737  007726                  JSR     PC,CKDATA       ;READ AND CHECK FOR ANOTHER MARK CHAR
```

```
8603  034712  000377                          377
8604  034714  000000                          000
8605  034716  004737  003554          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
8606  034722                          ENDTST
8607  034722                                                                  L10073:
8608  034722  104401                                                                  TRAP    C$ETST
8609
8610
8611
8612
8613
8614                                  ;;*************************************************************************
8615                                  .SBTTL      TEST 40 - MSG TERMINATION WITH GA CHARS - BIT MODE,NO CRC
8616                                  ;*
8617                                  ;* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
8618                                  ;* INITIATED IN BIT MODE.
8619                                  ;* 2 FLAG CHARACTERS ARE SENT, FOLLOWED BY
8620                                  ;* THE FOLLOWING DATA CHARACTERS : 000, 125, 252, 377, 000. THEN THE LOOP
8621                                  ;* MODE BIT (STRIP) IS SET AND 2 TERMINATING GO-AHEAD CHARACTERS ARE
8622                                  ;* SENT. EACH USYRT RCV FLAG IS TIMED AS IT IS RECEIVED, AND THE 5 DATA
8623                                  ;* WORDS ARE READ AND COMPARED TO EXPECTED VALUES.
8624                                  ;* ALSO, THE FIRST GA CHAR IS CHECKED BY SCANNING THE TXDATA BIT AS THE GA
8625                                  ;* IS BEING TRANSMITTED (GA = 376 OCTAL).
8626                                  ;* THE TEST ALSO CHECKS FOR SETTING OF RAB AND EBLK
8627                                  ;* IN LOOP MODE.
8628                                  ;;*************************************************************************
8629  034724                          BGNTST
8630  034724                                                                  T40::
8631  034724  012737  035060  002340          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
8632  034732  004737  005516                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
8633  034736  000000                          000
8634  034740  000310                          CRC2!CRC1!STRIP
8635  034742  004737  010774                  JSR     PC,LODMSG       ;LOAD DATA CHARS INTO TX SILO
8636  034746  003202                          MSG1+4
8637  034750  000005                          5
8638  034752  012737  005000  002400          MOV     #TXGOA!TXEOM,TXWORD
8639  034760  004737  005152                  JSR     PC,LDTXSI       ;LOAD A GA CHAR INTO TX SILO
8640  034764  004737  005152                  JSR     PC,LDTXSI       ;LOAD ANOTHER GA
8641  034770  004737  005124                  JSR     PC,WAIT50       ;ALLOW SILO TO RIPPLE
8642  034774  004737  005232                  JSR     PC,STPLU        ;CLOCK LU UNTIL GA CHAR IS TX'ING
8643  035000  100071                          CHPCHK!57.
8644  035002  004737  011154                  JSR     PC,CKTBIT       ;SCAN TXDATA BIT FOR GA CHAR
8645  035006  000376                          376
8646  035010  004737  007726                  JSR     PC,CKDATA       ;RCV 000 CHAR, CLK 125
8647  035014  000000                          000
8648  035016  000000                          0
8649  035020  004737  007726                  JSR     PC,CKDATA       ;RCV 125 CHAR, CLK 252
8650  035024  000125                          125
8651  035026  000000                          0
8652  035030  004737  007726                  JSR     PC,CKDATA       ;RCV 252 CHAR, CLK 377
8653  035034  000252                          252
8654  035036  000000                          0
8655  035040  004737  007726                  JSR     PC,CKDATA       ;RCV 377 CHAR
8656  035044  000377                          377
8657  035046  000010                          8.
8658  035050  004737  007726                  JSR     PC,CKDATA       ;RCV 000 CHAR, CHK RAB = 1, EBLK = 1
```

```
 8659  035054  003000                        3000
 8660  035056  000010                        8.
 8661  035060  004737  003554        24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 8662  035064                        ENDTST
 8663  035064                                                                L10074:
 8664  035064  104401                                                              TRAP    C$ETST
 8665
 8666
 8667
 8668
 8669
 8670                                ;;**************************************************************************
 8671                                .SBTTL      TEST 41 - IDLE SYNCHS TEST - CHAR MODE
 8672                                ;*
 8673                                ;* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
 8674                                ;* INITIATED IN CHAR MODE. 24(DEC) SYNCHS ARE SENT.
 8675                                ;* EACH SYNCH IS TIMED AS IT IS RECEIVED, AND THE BITS ARE CHECKED
 8676                                ;* FOR A VALID SYNCH CHAR FOR EACH OF THE 22 SYNCHS WHICH FOLLOW
 8677                                ;* THE FIRST TWO (THESE PERFORM SYNCHRONIZATION, AND ARE NOT READ).
 8678                                ;* WHILE THE LAST SYNCH IS BEING TRANSMITTED, OC IS SET, AND THE
 8679                                ;* NEXT CHAR RCV'D AFTER THE SYNCH IS CHECKED TO BE 377 (LINE MARKING).
 8680                                ;* THEN, A MASTER CLEAR IS ISSUED.
 8681                                ;* THE SYNCH CHAR USED IS 226 (OCTAL).
 8682                                ;;**************************************************************************
 8683  035066                        BGNTST
 8684  035066                                                                T41::
 8685  035066  012737  035226  002340        MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
 8686  035074  004737  005516                JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
 8687  035100  000226                        SYNCH
 8688  035102  000341                        CRC2!CRC1!IDLE!DDCMP
 8689  035104  012701  000026                MOV     #22.,R1         ;INIT COUNTER
 8690  035110  012737  000226  002400        MOV     #SYNCH,TXWORD
 8691  035116  004737  005152        6$:     JSR     PC,LDTXSI       ;LOAD AN SOM INTO TX SILO
 8692  035122  005301                        DEC     R1              ;DECR COUNTER
 8693  035124  001374                        BNE     6$              ;BR IF MORE TO LOAD
 8694  035126  004737  005124                JSR     PC,WAIT50       ;ALLOW SILO TO RIPPLE
 8695  035132  004737  007412                JSR     PC,RCV1ST       ;CLK LU UNTIL 3RD SYNCH RCV'D (RCVR IS ACTIVE)
 8696  035136  000030                        24.
 8697  035140  012701  000025                MOV     #21.,R1         ;INIT COUNTER
 8698  035144  004737  007726        8$:     JSR     PC,CKDATA       ;READ A SYNCH, CLK NEXT ONE
 8699  035150  000226                        SYNCH
 8700  035152  000010                        8.
 8701  035154  005301                        DEC     R1              ;DERC COUNTER
 8702  035156  001372                        BNE     8$              ;BR IF NOT ALL CHECKED
 8703  035160  004737  007726                JSR     PC,CKDATA       ;CHECK LAST SYNCH
 8704  035164  000226                        SYNCH
 8705  035166  000004                        4
 8706  035170  012737  000011  002356        MOV     #11,REGNUM      ;SET REG NO. = 11
 8707  035176  112737  000200  002344        MOVB    #OC,WRIBYT
 8708  035204  004737  003726                JSR     PC,WRITLU       ;SET OC IN REG 11
 8709  035210  004737  005232                JSR     PC,STPLU        ;FINISH CLOCKING CHAR, AND THEN SOME
 8710  035214  000014                        12.
 8711  035216  004737  007726                JSR     PC,CKDATA       ;RCV A MARK CHAR, CHK IT
 8712  035222  000377                        377
 8713  035224  000000                        0
 8714  035226  004737  003554        24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
```

```
 8715  035232                          ENDTST
 8716  035232                                                                  L10075:
 8717  035232  104401                                                                  TRAP    C$ETST
 8718
 8719
 8720
 8721
 8722
 8723                                  ;:*****************************************************************************
 8724                                  .SBTTL      TEST 42 - STRIP SYNCH TEST
 8725                                  ;*
 8726                                  ;* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
 8727                                  ;* INITIATED IN CHAR MODE AND WITH THE STRIP SYNCH
 8728                                  ;* BIT SET. THEN 24 (DEC) SYNCHS ARE SENT
 8729                                  ;* FOLLOWED BY THE FOLLOWING DATA CHARACTERS : 377, 000, 125, 252,
 8730                                  ;* AND 2 TERMINATING SYNCHS.
 8731                                  ;* EACH OF THE 23 SYNCHS AFTER THE FIRST ARE CHECKED AT THE TRANSMITTER OUTPUT,
 8732                                  ;* BY SCANNING THE TXDATA BIT.
 8733                                  ;* EACH USYRT RCV FLAG IS TIMED AS IT IS RECEIVED, AND THE 4 DATA WORDS
 8734                                  ;* ARE READ AND COMPARED TO EXPECTED VALUES.
 8735                                  ;* FINALLY, THE LINE UNIT IS CLOCKED FOR SEVERAL CHAR TIMES, AND A CHECK
 8736                                  ;* IS MADE FOR OACT = 0 (TEOM SHOULD CAUSE TX ENABLE TO DROP).
 8737                                  ;* THE ABOVE TEST IS REPEATED FOR EACH OF THE FOLLOWING SYNCH CHAR DATA
 8738                                  ;* PATTERNS : 226,000,125,252,376,177.
 8739                                  ;:*****************************************************************************
 8740  035234                          BGNTST
 8741  035234                                                                          T42::
 8742  035234  012737  035454  002340          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
 8743  035242  012701  035462          MOV     #SYNPAT,R1      ;GET POINTER TO DATA
 8744  035246  011137  035256  2$:     MOV     (R1),3$         ;GET A SYNCH PATTERN
 8745  035252  004737  005516          JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
 8746  035256  000000          3$:     .WORD   0               ;SYNCH PATTERN GOES HERE
 8747  035260  000311          CRC2!CRC1!STRIP!DDCMP
 8748  035262  012737  000400  002400          MOV     #TXSOM,TXWORD
 8749  035270  012702  000026          MOV     #22.,R2         ;LOAD 22 SOM'S INTO TX SILO
 8750  035274  004737  005152  6$:     JSR     PC,LDTXSI
 8751  035300  005302          DEC     R2
 8752  035302  001374          BNE     6$
 8753  035304  004737  010774          JSR     PC,LODMSG       ;LOAD DATA CHARS INTO TX SILO
 8754  035310  003224          MSG4
 8755  035312  000006          6
 8756  035314  012737  001000  002400          MOV     #TXEOM,TXWORD
 8757  035322  004737  005152          JSR     PC,LDTXSI       ;LOAD A TEOM
 8758  035326  004737  005152          JSR     PC,LDTXSI       ;LOAD ANOTHER TEOM
 8759  035332  004737  005124          JSR     PC,WAIT50       ;ALLOW SILO TO RIPPLE
 8760  035336  011137  035360          MOV     (R1),16$        ;GET CURRENT SYNCH PATTERN
 8761  035342  012702  000027          MOV     #23.,R2         ;INIT COUNTER
 8762  035346  004737  005232          JSR     PC,STPLU        ;CLOCK OUT FIRST SYNCH
 8763  035352  100010          CHPCHK!8.
 8764  035354  004737  011154  14$:     JSR     PC,CKTBIT       ;CHECK TX'D SYNCH
 8765  035360  000000          16$:    .WORD   0               ;SYNCH PATTERN GOES HERE
 8766  035362  005302          DEC     R2              ;DECR COUNTER
 8767  035364  001373          BNE     14$             ;BR IF NOT DONE CHECKING LAST 23 SYNCHS
 8768  035366  004737  007412          JSR     PC,RCV1ST       ;CLOCK UNTIL 000 CHAR RCV'D
 8769  035372  000010          8.
 8770  035374  004737  007726          JSR     PC,CKDATA       ;RCV 377, CLOCK 000
```

```
 8771  035400  000377                        377
 8772  035402  000010                        8.
 8773  035404  004737  007726                JSR     PC,CKDATA       ;RCV 000, CLK 125
 8774  035410  000000                        000
 8775  035412  000010                        8.
 8776  035414  004737  007726                JSR     PC,CKDATA       ;RCV 125, CLK 252
 8777  035420  000125                        125
 8778  035422  000010                        8.
 8779  035424  004737  007726                JSR     PC,CKDATA       ;RCV 252, CLK END OF MSG
 8780  035430  000252                        252
 8781  035432  000040                        32.
 8782  035434  004737  005330                JSR     PC,OACTIV       ;CHK FOR OACT = 0
 8783  035440  000000                        0
 8784  035442  062701  000002                ADD     #2,R1           ;INIT SYNCH PATTERN POINTER
 8785  035446  020127  035476                CMP     R1,#SYNPAT+12.  ;SEE IF ALL PATTERNS CHECKED YET
 8786  035452  103675                        BLO     2$              ;BR IF NOT YET
 8787  035454  004737  003554        24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
 8788  035460                        ENDTST
 8789  035460                                                                L10076:
 8790  035460  104401                                                              TRAP    C$ETST
 8791
 8792  035462  000226                SYNPAT: 226
 8793  035464  000000                        000
 8794  035466  000125                        125
 8795  035470  000252                        252
 8796  035472  000376                        376
 8797  035474  000177                        177
 8798
 8799
 8800
 8801
 8802
```

```
 8803                                   .SBTTL  HARDWARE PARAMETER CODING SECTION
 8804
 8805
 8806
 8807                                   ;////////////////////////////////////////////////////////////////////////
 8808                                   ;/ THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
 8809                                   ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
 8810                                   ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
 8811                                   ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
 8812                                   ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
 8813                                   ;/ WITH THE OPERATOR.
 8814                                   ;////////////////////////////////////////////////////////////////////////
 8815
 8816  035476                                   BGNHRD
 8817  035476  000011                                                          .WORD L10077-L$HARD/2
 8818  035500                                                         L$HARD::
 8819
 8820  035500                                   GPRMA   ADDRES,0,0,160000,177776,YES
 8821  035500  000031                                                          .WORD   T$CODE
 8822  035502  035522                                                          .WORD   ADDRES
 8823  035504  160000                                                          .WORD   T$LOLIM
 8824  035506  177776                                                          .WORD   T$HILIM
 8825  035510                                   GPRMD   ISRUN,2,0,7,0,7,YES
 8826  035510  001032                                                          .WORD   T$CODE
 8827  035512  035550                                                          .WORD   ISRUN
 8828  035514  000007                                                          .WORD   7
 8829  035516  000000                                                          .WORD   T$LOLIM
 8830  035520  000007                                                          .WORD   T$HILIM
 8831
 8832  035522                                   ENDHRD
 8833                                                                    .EVEN
 8834  035522                                                         L10077:
 8835
 8836  035522  042504  044526  042503  ADDRES: .ASCIZ  /DEVICE CSR ADDRESS : /
 8837  035530  041440  051123  040440
 8838  035536  042104  042522  051523
 8839  035544  035040  000040
 8840  035550  034115  030062  020067  ISRUN:  .ASCIZ  /M8207 RUN SWITCH (E28 SW7) - TYPE 0 IF OFF, 1 IF ON : /
 8841  035556  052522  020116  053523
 8842  035564  052111  044103  024040
 8843  035572  031105  020070  053523
 8844  035600  024467  026440  052040
 8845  035606  050131  020105  020060
 8846  035614  043111  047440  043106
 8847  035622  020054  020061  043111
 8848  035630  047440  020116  020072
 8849  035636     000
 8850          035640                   .EVEN
 8851
 8852
 8853
 8854
 8855
 8856
```

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

035640                                  BGNSFT
035640  000000                                                          .WORD L10100-L$SOFT/2
035642                                                          L$SOFT::

035642                                  ENDSFT
                                                                        .EVEN
035642                                                          L10100:

                                .EVEN


                                ;****** PATCH AREA FOR DEBUG ******
035642                          PATCH:
        036042                          .=.+200
036042  000240                          NOP
036044  000240                          NOP
036046  000240                          NOP
                                ;************************************


036050                                  ENDMOD

036050                                  LASTAD
                                                                        .EVEN
036050  000000                                                          .WORD   0
036052  000000                                                          .WORD   0
036054                          L$LAST::

        000001                  .END
```

```
ABORT = 000004          2466#
ADDRE  035522           4862    4878    4953    5070    5172    5297    5399    5508    5618    8822    8836#
ADR   = 000020 G        2414#
ANBITS  002537          2864#   6784    6785    6851    6852    6914    6915
APA   = 000200          2644#
ASBC0 = 000020          2601#
ASBC1 = 000040          2600#
ASBC2 = 000100          2599#
ASSEMB= 000010          2098
AXNUM   002360          2785#   3443    3481    3494    3519    3521    3527*   3533*   3534    3536*   3538    3753    3761*
                        3804*   4028    4034*   4079*   4085*   4366    4368*   4376*   4385*   6598*   6616*   6620*   6636
                        6654*   6671*   6702*   6709*   6711    6729*   6781*   6802*   6848*   6869*   6911*   6932*   6987*
                        7011*   7057*   7161*   7178*   7294*   7651*   7720*   7836*   7927*   8017*   8087*   8173*   8229*
                        8285*   8477*   8532*
AX0.15= 002300          2714#   3526    5028    5137    5255    5364    5466    5583
AX0.16= 002302          2715#   5027    5136    5254    5363    5465    5582
AX1   = 000001          2497#   2569#
AX1.15= 002304          2716#   5026    5135    5253    5362    5464    5581
AX1.16= 002306          2717#   5025    5134    5252    5361    5463    5580
AX2   = 000002          2496#   2568#
AX2.15= 002310          2718#   5045    5154    5272    5381    5483    5600
AX2.16= 002312          2719#   5044    5153    5271    5380    5482    5599
AX3.15= 002314          2720#   5043    5152    5270    5379    5481    5598
AX3.16= 002316          2721#   5042    5151    5269    5378    5480    5597
AX315U= 000372          2664#   6638    6713    6994
A1      024144          6484    6519    6547    6556    6575#
A10     034126          8369    8444#
A2      026552          7144    7197#
A3      027232          7227    7343    7347#
A4      027346          7372    7398#
A5      027456          7424    7447#
A6      027652          7475    7514#
A7      030216          7543    7563    7576    7592    7615    7619    7627#
A8      030432          7647    7686#
A9      030762          7707    7769#
BADDAT  002364          2787#   4216*   4217*   4246*   4247*   4899    4974    5201    5529    5971*   6013*   6053*   6094*
                        6137*   6186*   6247*   6293*   6337*   6381*   6425*   6504*   6532*   6560*   6643*   6660*   6718*
                        6734*   6791*   6807*   6858*   6873*   6921*   6936*   7000*   7016*   7076*   7109*   7167*   7183*
                        7299*   7328*   7858*   7877*   7967*   8039*   8127*   8178*   8289*
BCC   = 000001          2545#   4250    4252    4262
BCCCHK= 100000          2729#   4248
BIT0  = 000001 G        2387#   2447    2459    2468    2497    2509    2521    2533    2545    2557    2569    2581    2593
                        2605    2617    2626    2638    2651    2663    2674    2746    3521    3562    3707    3939    3988
                        4036    4460    5774
BIT00 = 000001 G        2376#   2387
BIT01 = 000002 G        2375#   2386
BIT02 = 000004 G        2374#   2385
BIT03 = 000010 G        2373#   2384
BIT04 = 000020 G        2372#   2383
BIT05 = 000040 G        2371#   2382
BIT06 = 000100 G        2370#   2381
BIT07 = 000200 G        2369#   2380
BIT08 = 000400 G        2368#   2379
BIT09 = 001000 G        2367#   2378
BIT1  = 000002 G        2386#   2446    2458    2467    2486    2496    2508    2520    2532    2544    2556    2568    2580
                        2592    2604    2616    2625    2637    2650    2662    2673    2747    3586    3915    4058
```

```
BIT10 = 002000 G   2366# 2684 2696
BIT11 = 004000 G   2365# 2683 2695
BIT12 = 010000 G   2364#
BIT13 = 020000 G   2363#
BIT14 = 040000 G   2362#
BIT15 = 100000 G   2361# 2727 2729 2730 3675 3837
BIT2  = 000004 G   2385# 2445 2457 2466 2485 2495 2507 2519 2531 2543 2555 2567 2579
                   2591 2603 2615 2624 2636 2649 2661 2672
BIT3  = 000010 G   2384# 2444 2456 2465 2484 2494 2506 2518 2530 2542 2554 2566 2578
                   2590 2602 2614 2623 2635 2648 2660
BIT4  = 000020 G   2383# 2443 2455 2483 2493 2505 2517 2529 2541 2553 2565 2577 2589
                   2601 2613 2634 2647 2659
BIT5  = 000040 G   2382# 2454 2475 2482 2492 2504 2516 2528 2540 2552 2564 2576 2588
                   2600 2612 2633 2646 2658 2671
BIT6  = 000100 G   2381# 2442 2453 2474 2481 2491 2503 2515 2527 2539 2551 2563 2575
                   2587 2599 2611 2632 2645 2657 2670
BIT7  = 000200 G   2380# 2441 2452 2464 2473 2480 2502 2514 2526 2538 2550 2562 2574
                   2586 2598 2610 2622 2631 2644 2656 2669 4210 4211
BIT8  = 000400 G   2379# 2686 2698
BIT9  = 001000 G   2378# 2685 2697
BOE   = 000400 G   2418#
BPOLL = 000100     2474#
BSEL1   002426     2806# 3307* 3309* 3310* 3324* 3325* 3330* 3422* 3424* 3525* 3679* 3680* 3682*
                   3780* 3781* 3786* 3799* 5792* 5793* 5794  6083* 6981*
BSEL2   002430     2807# 5794* 5795* 6497  6526  6554
BSEL4   002432     2808# 3360  3381* 6129*
CARR  = 000001     2557#
CHPCHK= 100000     2727# 7378 7430 7484 7549 7602 7657 7713 7800 8482 8643 8763
CHPTYP  002406     2796# 3676 3794* 3798* 3838 5731* 8246 8268
CKDATA  007726     4201# 8395 8424 8599 8602 8646 8649 8652 8655 8658 8698 8703 8711
                   8770  8773 8776 8779
CKTBIT  011154     4456# 7802 8644 8764
CRCCHK= 100000     2730# 4519
CRCTY0= 000001     2651#
CRCTY1= 000002     2650#
CRCTY2= 000004     2649#
CRC1  = 000100     2503# 7478 8013 8083 8216 8470 8525 8593 8634 8688 8747
CRC2  = 000200     2502# 7478 8013 8083 8216 8470 8525 8593 8634 8688 8747
CS    = 000004     2555#
C$AU  = 000052     2098# 5908
C$AUTO= 000061     2098# 5836
C$BRK = 000022     2098#
C$BSEG= 000004     2098# 6004 6285 6329 6373 6417 6780 6847 6910 6986 7066
C$BSUB= 000002     2098# 7063 7100 7147
C$CEFG= 000045     2098#
C$CLCK= 000062     2098#
C$CLEA= 000012     2098# 5855
C$CLOS= 000035     2098#
C$CLP1= 000006     2098#
C$CVEC= 000036     2098#
C$DCLN= 000044     2098#
C$DODU= 000051     2098# 5831
C$DRPT= 000024     2098#
C$DU  = 000053     2098# 5883
C$EDIT= 000003     2098# 2180
C$ERDF= 000055     2098# 3569 3579 3593 3603 3714 3724 3886 3922 3932 3946 3956 3995
```

```
                    4005  4043  4053  4065  4075  4225  4234  4257  4267  4280  4290  4303  4313
                    4326  4336  4472  4482  5942  5975  6017  6057  6098  6141  6190  6251  6297
                    6341  6385  6429  6508  6536  6564  6609  6629  6647  6664  6722  6738  6795
                    6811  6862  6877  6925  6940  7004  7020  7080  7113  7171  7187  7303  7332
                    7509  7559  7572  7588  7611  7623  7862  7881  7971  8043  8131  8182  8293
C$ERHR= 000056       2098#
C$ERRO= 000060       2098#
C$ERSF= 000054       2098#
C$ERSO= 000057       2098#
C$ESCA= 000010       2098#  6195  6256  6614  6634  6652  6669  6727  6743  6800  6867  6930  7009
                     7176  7192  7308  7337
C$ESEG= 000005       2098#  6024  6304  6348  6392  6436  6818  6884  6947  7027  7087
C$ESUB= 000003       2098#  7094  7120  7200
C$ETST= 000001       2098#  5950  5982  6030  6064  6105  6148  6203  6266  6310  6354  6398  6442
                     6579  6677  6753  6824  6890  6953  7034  7123  7203  7350  7402  7451  7518
                     7630  7689  7772  7810  7901  7991  8061  8151  8190  8343  8447  8502  8573
                     8608  8664  8717  8790
C$EXIT= 000032       2098#
C$GETB= 000026       2098#
C$GETW= 000027       2098#
C$GMAN= 000043       2098#
C$GPHR= 000042       2098#  5783
C$GPLO= 000030       2098#
C$GPRI= 000040       2098#
C$INIT= 000011       2098#  5804
C$INLP= 000020       2098#
C$MANI= 000050       2098#
C$MEM = 000031       2098#
C$MSG = 000023       2098#  4870  4943  5053  5162  5280  5389  5491  5608  5675
C$OPEN= 000034       2098#
C$PNTB= 000014       2098#  4866  4882  4888  4896  4904  4957  4963  4971  4979  5066  5074  5080
                     5088  5176  5184  5190  5198  5206  5293  5301  5307  5315  5403  5409  5417
                     5504  5512  5518  5526  5534  5622  5628  5636
C$PNTF= 000017       2098#  5879  6482  6517  6545  6573
C$PNTS= 000016       2098#
C$PNTX= 000015       2098#  4912  4922  4929  4939  4987  4997  5004  5014  5022  5032  5039  5049
                     5096  5106  5113  5123  5131  5141  5148  5158  5214  5224  5231  5241  5249
                     5259  5266  5276  5323  5333  5340  5350  5358  5368  5375  5385  5425  5435
                     5442  5452  5460  5470  5477  5487  5542  5552  5559  5569  5577  5587  5594
                     5604  5644  5654  5661  5671
C$QIO = 000377       2098#
C$RDBU= 000007       2098#
C$REFG= 000047       2098#  5745  5751  5757  5763
C$RESE= 000033       2098#  5872  6082
C$REVI= 000003       2098#  2179
C$RFLA= 000021       2098#
C$RPT = 000025       2098#  5694
C$SEFG= 000046       2098#
C$SPRI= 000041       2098#  5822  5933
C$SVEC= 000037       2098#
C$TPRI= 000013       2098#
C32BCC= 000040       2658#  2664  6995
C32ENB= 000004       2661#
DDC   = 000100       2645#
DDCMP = 000001       2509#  2581#  7375  7478  7546  7599  7650  7791  7832  8013  8382  8470  8593
                     8688  8747
```

CZDMRF.P11 03-NOV-81 10:29 CROSS REFERENCE TABLE -- USER SYMBOLS

```
DEVMAP  002412          2798#   5770*   5789*
DEVPTR  002414          2799#   5774*   5787*   5789    5790*
DFPTBL  002252 G        2305#
DH1     014242          4813#   4892    5084    5413    5522    5632
DH2     014264          4816#   4908    4983    5092    5210    5319    5421    5538    5640
DH3     014313          4820#   4907    4982    5091    5209    5318    5420    5537    5639
DH4     014351          4826#   4925    5000    5109    5227    5336    5438    5555    5657
DH5     014413          3520    4832#
DH6     014416          3523    4833#
DH7     014421          4834#   5018    5127    5245    5354    5456    5573
DH8     014453          4839#   5017    5126    5244    5353    5455    5572
DH9     014512          4845#   5035    5144    5262    5371    5473    5590
DIAGMC= 000000          2098
DISILO  002404          2795#   3446    3483    3497    5730*   8170*   8186*   8222*   8339*
DISSI = 000040          2492#   2564#   8168    8170    8220    8222    8331
DTR   = 000100          2481#   2551#
EBLK  = 000002          2544#   4273    4275    4285
EF.CON= 000036 G        2394#   5762
EF.NEW= 000035 G        2395#   5756
EF.PWR= 000034 G        2396#
EF.RES= 000037 G        2393#   5750
EF.STA= 000040 G        2392#   5744
EM1     012103          4597#   5944
EM10    012425          3605    4638#
EM11    012446          3716    4641#
EM12    012463          3726    4644#
EM13    012504          4647#   7590
EM14    012534          4651#   7574    7613
EM15    012551          4654#   7625
EM16    012600          4658#   7561
EM17    012621          3924    4661#
EM18    012636          3934    4664#
EM19    012657          3948    4667#
EM2     012137          4602#   5977    6059    6192    6649    6666    7115
EM20    012674          3958    4670#
EM21    012715          3997    4673#
EM22    012732          4007    4676#
EM23    012753          4679#
EM24    012774          4682#
EM25    013011          4685#
EM26    013045          4690#   7864    7973    8045    8133    8295
EM27    013077          4695#   7883
EM28    013130          4055    4700#
EM29    013151          4045    4703#
EM3     012176          4608#   6019    6143    6253    6299    6343    6387    6431    6724    6740    6797    6813    6864
                        6879    6927    6942    7006    7022    7082    7173    7189    7305    7334    8184
EM30    013166          4077    4706#
EM31    013207          4067    4709#
EM32    013224          4474    4712#
EM33    013253          4484    4716#
EM34    013276          4236    4720#
EM35    013324          4259    4724#
EM36    013344          4269    4727#
EM37    013360          4282    4729#
EM38    013401          4292    4732#
EM39    013416          4305    4735#
```

```
EM4     012215           4611#  6100
EM40    013436           4315   4738#
EM41    013452           4328   4740#
EM42    013473           4338   4743#
EM43    013510           4746#
EM44    013535           4750#
EM45    013562           4754#
EM46    013607           4758#
EM47    013642           4763#
EM48    013675           4768#
EM49    013730           4773#
EM5     012270           4619#  6510   6566
EM50    013767           4779#
EM51    014023           4784#
EM52    014063           4790#  6611
EM53    014124           4796#  6631
EM54    014104           4227   4802#
EM6     012321           4624#  6538
EM60    014206           4805#  7511
EM65    014222           3888   4807#
EM7     012352           3571   4629#
EM8     012367           3581   4632#
EM9     012410           3595   4635#
ENAX  = 000004           2495#  2567#  3445   3496
ENDIT   021410           5765   5801#
ENDPAT  003176           3204#
ENDTRN  006252           3867#  7397   7446   7513   7685   7768
EOM   = 000002           2467#
ERRFLG  002334           2775#
ERROR1  002376           2792#  3441*  3454*  3479*  3505*  5732*  6603  6623
ERR1    014556 G         4859#  5945
ERR2    014610 G         4875#  5978   6020   6060   6101   6144   6193  6254  6300  6344  6388  6432  6511
                         6539   6567
ERR3    015116 G         4950#  6650   6667   6725   6741   6798   6814  6865  6880  6928  6943  7007  7023
                         7116   7174   7190   7306   7335   7865   7884  7974  8046  8134  8185  8296
ERR4    015600 G         3572   3582   3596   3606   3717   3727   3889  3925  3935  3949  3959  3998  4008
                         4228   4475   4485   5060#
ERR5    016256 G         5169#  7083
ERR6    016770 G         4046   4056   4068   4078   5287#
ERR7    017446 G         5396#  7512   7562   7575   7591   7614   7626
ERR8    020100 G         4237   4260   4270   4283   4293   4306   4316  4329  4339  5498#
ERR9    020606 G         5615#  6612   6632
EVL   = 000004 G         2412#
E$END = 002100           2098#
E$LOAD= 000035           2098#  2203
FMT1    011432           4538#  4863   4879   4954   5071   5173   5298  5400  5509  5619
FMT10   011675           4569#  5063   5290   5501
FMT11   011726           4574#  5181
FMT19   011765           4580#  6479
FMT2    011442           4540#  4885   4960   5077   5187   5304   5400  5515  5625
FMT24   012016           4585#  6514   6542   6570
FMT27   021524           5876   5885#
FMT3    011464           4543#  4901   4976   5203   5531
FMT4    011526           4549#  4909   4984   5019   5093   5128   5211  5246  5320  5355  5422  5457  5539
                         5574   5641
FMT5    011541           4551#  4919   4994   5029   5103   5138   5221  5256  5330  5365  5432  5467  5549
```

```
                    5584  5651
FMT6     011571     4556# 4936  5011  5046  5120  5155  5238  5273  5347  5382  5449  5484  5566
                    5601  5668
FMT7     011624     4561# 4893  5085  5414  5523  5633
FMT8     011634     4563# 4968  5195  5312
FMT9     011670     4568# 4926  5001  5036  5110  5145  5228  5263  5337  5372  5439  5474  5556
                    5591  5658
FRSTIM   002370     2789# 5734  5741*
F$AU   = 000015     2098# 5905  5907
F$AUTO= 000020      2098# 5818  5835
F$BGN  = 000040     2098# 2104  4859  4875  4950  5060  5169  5287  5396  5498  5615  5690  5708
                    5726  5818  5850  5869  5905  5929  5949  5963  5991  5998  6004  6029  6042
                    6063  6076  6104  6122  6147  6166  6195  6202  6219  6256  6265  6279  6285
                    6309  6323  6329  6353  6367  6373  6397  6411  6417  6441  6470  6578  6596
                    6614  6634  6652  6669  6676  6694  6727  6743  6752  6775  6780  6800  6823
                    6842  6847  6867  6889  6905  6910  6930  6952  6979  6986  7009  7033  7052
                    7062  7066  7093  7099  7119  7122  7142  7146  7176  7192  7199  7202  7226
                    7308  7337  7349  7371  7401  7423  7450  7474  7517  7539  7629  7646  7688
                    7706  7771  7788  7809  7827  7900  7918  7990  8008  8060  8078  8150  8165
                    8189  8208  8342  8368  8446  8466  8501  8521  8572  8589  8607  8630  8663
                    8684  8716  8741  8789  8817  8870  8898
F$CLEA= 000007       2098# 5850  5854
F$DU   = 000016     2098# 5869  5882
F$END  = 000041     2098# 2104  4871  4944  5054  5163  5281  5390  5492  5609  5676  5695  5805
                    5837  5856  5884  5909  5929  5949  5951  5963  5981  5983  5998  6025  6029
                    6031  6042  6063  6065  6076  6104  6106  6122  6147  6149  6166  6195  6202
                    6204  6219  6256  6265  6267  6279  6305  6309  6311  6323  6349  6353  6355
                    6367  6393  6397  6399  6411  6437  6441  6443  6470  6578  6580  6596  6614
                    6634  6652  6669  6676  6678  6694  6727  6743  6752  6754  6775  6800  6819
                    6823  6825  6842  6867  6885  6889  6891  6905  6930  6948  6952  6954  6979
                    7009  7028  7033  7035  7052  7062  7088  7093  7095  7099  7119  7121  7122
                    7124  7142  7146  7176  7192  7199  7201  7202  7204  7226  7308  7337  7349
                    7351  7371  7401  7403  7423  7450  7452  7474  7517  7519  7539  7629  7631
                    7646  7688  7690  7706  7771  7773  7788  7809  7811  7827  7900  7902  7918
                    7990  7992  8008  8060  8062  8078  8150  8152  8165  8189  8191  8208  8342
                    8344  8368  8446  8448  8466  8501  8503  8521  8572  8574  8589  8607  8609
                    8630  8663  8665  8684  8716  8718  8741  8789  8791  8835  8877  8898
F$HARD= 000004       2098# 8817  8833
F$HW   = 000013     2098# 2303  2311
F$INIT= 000006      2098# 5726  5803
F$JMP  = 000050     2098#
F$MOD  = 000000     2098# 2104  8898
F$MSG  = 000011     2098# 4859  4869  4875  4942  4950  5052  5060  5161  5169  5279  5287  5388
                    5396  5490  5498  5607  5615  5674
F$PROT= 000021      2098# 5708  5713
F$PWR  = 000017     2098#
F$RPT  = 000012     2098# 5690  5693
F$SEG  = 000003     2098# 6004  6023  6285  6303  6329  6347  6373  6391  6417  6435  6780  6817
                    6847  6883  6910  6946  6986  7026  7066  7086
F$SOFT= 000005      2098# 8870  8875
F$SRV  = 000010     2098#
F$SUB  = 000002     2098# 7063  7093  7100  7119  7147  7199
F$SW   = 000014     2098# 2325  2331
F$TEST= 000001      2098# 5930  5949  5964  5981  5999  6029  6043  6063  6077  6104  6123  6147
                    6167  6202  6220  6265  6280  6309  6324  6353  6368  6397  6412  6441  6471
                    6578  6597  6676  6695  6752  6776  6823  6843  6889  6906  6952  6980  7033
```

```
                        7053    7122    7143    7202    7227    7349    7372    7401    7424    7450    7475    7517    7540
                        7629    7647    7688    7707    7771    7789    7809    7828    7900    7919    7990    8009    8060
                        8079    8150    8166    8189    8209    8342    8369    8446    8467    8501    8522    8572    8590
                        8607    8631    8663    8685    8716    8742    8789
GETALL  004504          3518#   3566    3576    3590    3600    3711    3721    3883    3919    3929    3943    3953    3992
                        4002    4040    4050    4062    4072    4222    4231    4254    4264    4277    4287    4300    4310
                        4323    4333    4469    4479    6644    6661    6719    6735    6792    6808    6859    6874    6922
                        6937    7001    7017    7077    7110    7168    7184    7300    7329    7506    7556    7569    7585
                        7608    7620    7859    7878    7968    8040    8128    8179    8290
GETPRM  021264          5766    5777#   5788
GETREG  003774          3395#   3524    5972    6014    6054    6095    6138    6187    6248    6294    6338    6382    6426
                        6505    6533    6561    6606    6626
GOAH   = 000010         2465#
GOODAT  002362          2786#   4214*   4215*   4244*   4245*   4900    4975    5202    5530    5969*   5970*   6012*   6051*
                        6052*   6092*   6093*   6135*   6136*   6184*   6185*   6245*   6246*   6292*   6336*   6380*   6424*
                        6503*   6531*   6559*   6641*   6642*   6658*   6659*   6716*   6717*   6732*   6733*   6790*   6805*
                        6806*   6857*   6872*   6920*   6935*   6998*   6999*   7014*   7015*   7074*   7075*   7107*   7108*
                        7165*   7166*   7181*   7182*   7298*   7327*   7857*   7876*   7890*   7966*   8038*   8126*   8177*
                        8288*   8312*   8328*
G$CNTO= 000200          2098#
G$DELM= 000372          2098#
G$DISP= 000003          2098#
G$EXCP= 000400          2098#
G$HILI= 000002          2098#
G$LOLI= 000001          2098#
G$NO   = 000000         2098#
G$OFFS= 000400          2098#   8821    8826
G$OFSI= 000376          2098#   8821    8826
G$PRMA= 000001          2098#   8821
G$PRMD= 000002          2098#   8826
G$PRML= 000000          2098#
G$RADA= 000140          2098#
G$RADB= 000000          2098#
G$RADD= 000040          2098#
G$RADL= 000120          2098#
G$RADO= 000020          2098#   8821    8826
G$XFER= 000004          2098#
G$YES  = 000010         2098#   8821    8826
HDX    = 000020         2483#   2553#
HELP   = 000001         2089#   2128    2221    2285    3283
HOE    = 100000 G       2425#
IACT   = 000100         2539#   3990    4000
IACTIV  006672          3980#   4131    4152    7847    7852    7871    7888    7896    7938    7947    7954    7959    7980
                        7986    8028    8033    8052    8056    8098    8107    8114    8119    8140    8146    8254    8279
                        8304    8320
IBE    = 010000 G       2422#
IC     = 000200         2473#   2538#   7984    8054    8144
ICIR   = 000010         2578#   3941    3951
IDL    = 000010         2648#
IDLE   = 000040         2504#   7599    7650    8470    8593    8688
IDU    = 000040 G       2415#
IER    = 020000 G       2423#
IERR   = 000002         2508#   4176    4182    7832    7923    8013    8083    8216
INITRN  005516          3751#   7373    7425    7476    7544    7597    7648    7708    7789    8210    8380    8468    8523
                        8591    8632    8686    8745
INTFLG  002332          2773#
```

```
INTGRL= 000010          2660#   2664
IRDY  = 000020          2541#   3917    3927    4142
ISIRDY  006406          3907#   4133    4146    4150    4154    4432    4438    8266    8275    8283    8300    8308    8316
                        8324    8374    8399    8409    8418    8430
ISR   = 000100 G        2416#
ISRUN   035550          8827    8840#
IXE   = 004000 G        2421#
I$AU  = 000041          2098#   5905#   5909#
I$AUTO= 000041          2098#   5818#   5837#
I$CLN = 000041          2098#   5850#   5856#
I$DU  = 000041          2098#   5869#   5884#
I$HRD = 000041          8817#   8835#
I$INIT= 000041          2098#   5726#   5805#
I$MOD = 000041          2098#   2104#   8898#
I$MSG = 000041          2098#   4859#   4871#   4875#   4944#   4950#   5054#   5060#   5163#   5169#   5281#   5287#   5390#
                        5396#   5492#   5498#   5609#   5615#   5676#
I$PROT= 000040          2098#   5708#
I$PTAB= 000041          2098#
I$PWR = 000041          2098#
I$RPT = 000041          2098#   5690#   5695#
I$SEG = 000041          2098#   5929    5963    5998    6004#   6025#   6042    6076    6122    6166    6219    6279    6285#
                        6305#   6323    6329#   6349#   6367    6373#   6393#   6411    6417#   6437#   6470    6596    6694
                        6775    6780#   6800    6819#   6842    6847#   6867    6885#   6905    6910#   6930    6948#   6979
                        6986#   7009    7028#   7052    7062    7066#   7088#   7099    7142    7146    7226    7371    7423
                        7474    7539    7646    7706    7788    7827    7918    8008    8078    8165    8208    8368    8466
                        8521    8589    8630    8684    8741
I$SETU= 000041          2098#
I$SFT = 000041          8870#   8877#
I$SRV = 000041          2098#
I$SUB = 000041          2098#   5929    5963    5998    6042    6076    6122    6166    6219    6279    6323    6367    6411
                        6470    6596    6694    6775    6842    6905    6979    7052    7062#   7093#   7095#   7099#   7119#
                        7121#   7142    7146#   7176    7192    7199#   7201#   7226    7371    7423    7474    7539    7646
                        7706    7788    7827    7918    8008    8078    8165    8208    8368    8466    8521    8589    8630
                        8684    8741
I$TST = 000041          2098#   5929#   5949#   5951#   5963#   5981#   5983#   5998#   6029#   6031#   6042#   6063#   6065#
                        6076#   6104#   6106#   6122#   6147#   6149#   6166#   6195    6202#   6204#   6219#   6256    6265#
                        6267#   6279#   6309#   6311#   6323#   6353#   6355#   6367#   6397#   6399#   6411#   6441#   6443#
                        6470#   6578#   6580#   6596#   6614    6634    6652    6669    6676#   6678#   6694#   6727    6743
                        6752#   6754#   6775#   6823#   6825#   6842#   6889#   6891#   6905#   6952#   6954#   6979#   7033#
                        7035#   7052#   7062    7099    7122#   7124#   7142#   7146    7202#   7204#   7226#   7308    7337
                        7349#   7351#   7371#   7401#   7403#   7423#   7450#   7452#   7474#   7517#   7519#   7539#   7629#
                        7631#   7646#   7688#   7690#   7706#   7771#   7773#   7788#   7809#   7811#   7827#   7900#   7902#
                        7918#   7990#   7992#   8008#   8060#   8062#   8078#   8150#   8152#   8165#   8189#   8191#   8208#
                        8342#   8344#   8368#   8446#   8448#   8466#   8501#   8503#   8521#   8572#   8574#   8589#   8607#
                        8609#   8630#   8663#   8665#   8684#   8716#   8718#   8741#   8789#   8791#
I422  = 000200          2656#   2664
J$JMP = 000167          2098#
LDMSG1  011352          4509#
LDTXSI  005152          3650#   3771    3772    3830    4405    7238    7239    7263    7317    7578    7795    7841    7932
                        8022    8092    8172    8385    8407    8437    8441    8473    8528    8595    8639    8640    8691
                        8750    8757    8758
LOADAT  002366          2788#   5179    7072*   7073*
LODMSG  010774          4398#   4511    8635    8753
LOE   = 040000 G        2424#
LOGDEV  002322          2769#   5772*   5778*   5779    5782    5830    5875
LOOPIN  004052          3421#   6490
```

```
LOT    = 000010 G      2413#
LULOOP= 000010         2444#   3330    3525    3679    3780    6981
LULP   = 000040        2475#   2540#   7837    7928    8018    8088    8224
LUREG   002260         2706    2707    2708    2709    2710    2711    2712    2713    2714    2715    2716    2717    2718
                       2719    2720    2721    2763#
LUR10 = 002260         2706#   3397    4918    4993    5102    5220    5329    5431    5548    5650
LUR11 = 002262         2707#   4917    4992    5101    5219    5328    5430    5547    5649
LUR12 = 002264         2708#   4916    4991    5100    5218    5327    5429    5546    5648
LUR13 = 002266         2709#   4915    4990    5099    5217    5326    5428    5545    5647
LUR14 = 002270         2710#   4935    5010    5119    5237    5346    5448    5565    5667
LUR15 = 002272         2711#   4934    5009    5118    5236    5345    5447    5564    5666
LUR16 = 002274         2712#   4933    5008    5117    5235    5344    5446    5563    5665
LUR17 = 002276         2713#   4932    5007    5116    5234    5343    5445    5562    5664
LUSWI1  002444         2814#
LUSWI2  002446         2815#
LUSWI3  002450         2816#
LU1MOD  002000 G       2104#
L$ACP   002110 G       2210#
L$APT   002036 G       2168#
L$AU    021556 G       2195    5905#
L$AUT   002070 G       2194#
L$AUTO  021412 G       2211    5818#
L$CCP   002106 G       2208#
L$CLEA  021472 G       2209    5850#
L$CO    002032 G       2164#
L$DEPO  002011 G       2146#
L$DESC  003446 G       2201    3268#
L$DESP  002076 G       2200#
L$DEVP  002060 G       2186#
L$DISP  002124 G       2171    2241#
L$DLY   002116 G       2216#
L$DTP   002040 G       2170#
L$DTYP  002034 G       2166#
L$DU    021474 G       2197    5869#
L$DUT   002072 G       2196#
L$DVTY  003440 G       2187    3260#
L$EF    002052 G       2181#
L$ENVI  002044 G       2174#
L$ETP   002102 G       2204#
L$EXP1  002046 G       2176#
L$EXP4  002064 G       2190#
L$EXP5  002066 G       2192#
L$HARD  035500 G       2153    8817    8818#
L$HIME  002120 G       2218#
L$HPCP  002016 G       2152#
L$HPTP  002022 G       2156#
L$HW    002252 G       2157    2303    2304#
L$ICP   002104 G       2206#
L$INIT  021074 G       2207    5726#
L$LADP  002026 G       2160#
L$LAST  036054 G       2161    8903#
L$LOAD  002100 G       2202#
L$LUN   002074 G       2198#
L$MREV  002050 G       2178#
L$NAME  002000 G       2135#
L$PRIO  002042 G       2172#
```

CZDMRF.P11 03-NOV-81 10:29 CROSS REFERENCE TABLE -- USER SYMBOLS

```
L$PROT  021066 G        2213    5708#
L$PRT   002112 G        2212#
L$REPP  002062 G        2188#
L$REV   002010 G        2144#
L$RPT   021064 G        5690#
L$SOFT  035642 G        8870    8871#
L$SPC   002056 G        2184#
L$SPCP  002020 G        2154#
L$SPTP  002024 G        2158#
L$STA   002030 G        2162#
L$SW    002260 G        2325    2326#
L$TEST  002114 G        2214#
L$TIML  002014 G        2150#
L$UNIT  002012 G        2148#   5779
L10000  002256          2303    2311#
L10001  002260          2325    2331#
L10002  014606          4869#
L10003  015114          4942#
L10004  015576          5052#
L10005  016254          5161#
L10006  016766          5279#
L10007  017444          5388#
L10010  020076          5490#
L10011  020604          5607#
L10012  021062          5674#
L10013  021064          5693#
L10015  021410          5803#
L10016  021470          5835#
L10017  021472          5854#
L10020  021522          5882#
L10021  021556          5907#
L10022  021640          5949#
L10023  021724          5981#
L10024  022050          6029#
L10025  022152          6063#
L10026  022300          6104#
L10027  022420          6147#
L10030  022574          6196    6202#
L10031  023030          6257    6265#
L10032  023146          6309#
L10033  023264          6353#
L10034  023402          6397#
L10035  023520          6441#
L10036  024150          6578#
L10037  024502          6615    6635    6653    6670    6676#
L10040  024772          6728    6744    6752#
L10041  025202          6823#
L10042  025406          6889#
L10043  025612          6952#
L10044  026050          7033#
L10045  026302          7122#
L10046  026206          7093#
L10047  026300          7119#
L10050  026554          7202#
L10051  026552          7177    7193    7199#
L10052  027232          7309    7338    7349#
```

```
L10053  027352          7401#
L10054  027462          7450#
L10055  027656          7517#
L10056  030222          7629#
L10057  030432          7688#
L10060  030762          7771#
L10061  031062          7809#
L10062  031432          7900#
L10063  031774          7990#
L10064  032244          8060#
L10065  032606          8150#
L10066  032736          8189#
L10067  033650          8342#
L10070  034132          8446#
L10071  034332          8501#
L10072  034632          8572#
L10073  034722          8607#
L10074  035064          8663#
L10075  035232          8716#
L10076  035460          8789#
L10077  035522          8817    8834#
L10100  035642          8870    8876#
MAINT1= 000010          2484#
MAINT2= 000004          2485#
MCLK   = 000002          2580#  6498    6503    6527    6555    6559
MCLR   = 000100          2442#  3324    3325
MODR   = 000010          2554#
MPCSR   002424          2805#  4861    4877    4952    5069    5171    5296    5398    5507    5617    5791*   5825    5936
MPIVEC  002436          2811#
MPOVEC  002440          2812#
MPRIOR  002442          2813#
MSG1    003176          3213#  4512    4514    4517    7792    7796    7905    7839    7842    7851    7869    7930    7933
                        7944   7945    7952    7978    8020    8023    8032    8050    8090    8093    8104    8105    8112
                        8138   8226    8636
MSG4    003224          3225#  8754
MSTCLR  003554          3321#  3756    5965    5999    6043    6047    6077    6123    6167    6175    6220    6280    6324
                        6368   6412    6471    6576    6597    6619    6695    6776    6843    6906    6980    7053    7103
                        7148   7231    7399    7448    7515    7616    7627    7807    7829    7898    7920    7988    8010
                        8058   8080    8148    8166    8187    8213    8340    8373    8444    8499    8570    8605    8661
                        8714   8787
MVIOX  = 021000          2739#  3356    6488
MVIXO  = 122000          2740#  3378
MVIXOX= 121000          2741#  6127
NEWST   021244          5759    5771#   5780
OACT   = 000100          2527#  3709    3719    3784
OACTIV  005330          3699#  3759    3776    3791    3841    3877    8235    8239    8260    8277    8302    8318    8337
                        8782
OC     = 000200          2464#  2526#   3872    8334    8707
OCOR   = 000020          2577#  3588    3598    3796    7251
OP     = 000002          2662#  2664
ORDY   = 000020          2529#  3564    3574    6241    6242    7283
OSIRDY  004640          3554#  3757    3774    3801    3845    3851    3875    7149    7159    7232    7241    7256    7270
                        7288   7319    7341    7344    8227    8233    8249    8252    8262
OVRR   = 000010          2542#  4319    4321    4331
O$APTS= 000000          2098#  2162
O$AU   = 000001          2098#  2127#   2194
```

```
O$BGNR= 000000          2098#   2188
O$BGNS= 000000          2098#   2154
O$DU  = 000001          2098#   2127#   2196
O$ERRT= 000000          2098#   2204
O$GNSW= 000000          2098#   2158
O$POIN= 000001          2098#   2127#   2220
O$SETU= 000000          2098#   2148    8901
PATA    002547          2875#   6001
PATB    002573          2898#   6026    6230
PATC    002603          2909#   6282
PATCH   035642          8888#
PATD    002606          2915#   6306    6326
PATE    002611          2921#   6350    6370
PATF    002617          2930#   6394    6414
PATG    002622          2936#   6169    6438
PATH    002632          2947#   6173    6599
PATI    002642          2958#   6617    6621    6696    6803    6806    7105    7108
PATJ    002652          2969#   6673    6699    6703    6704
PATK    002662          2980#   6777    6907
PATL    002772          3054#   6820    6844    6949
PATM    003000          3063#   5967    5970    6049    6052    6090    6093    6133    6136    6177    6222    6886
PATN    003010          3074#   6199    7143
PATO    003026          3091#   7054    7195
PATP    003044          3108#   7055    7090
PATU    003062          3125#   6983
PATV    003130          3165#   6982    7030
PNT   = 001000 G        2419#
POLL  = 000200          2480#
PRI   = 002000 G        2420#
PRIOR   002326          2771#
PRI00 = 000000 G        2408#
PRI01 = 000040 G        2407#
PRI02 = 000100 G        2406#
PRI03 = 000140 G        2405#
PRI04 = 000200 G        2404#
PRI05 = 000240 G        2403#
PRI06 = 000300 G        2402#
PRI07 = 000340 G        2401#   5821    5824    5932    5935
PSTACK  002324          2770#   3608    3729    3961    4010    4080    4348    4491    5728*
RAB   = 000004          2543#   4296    4298    4308
RABT  = 000004          2603#
RAX15   002346          2780#   3458*   3459*   3529    6638*   6639    6643    6713*   6714    6718    6788    6791    6855
                        6858    6918    6921    6994*   6995*   6996    7000    7163    7167    7296    7299    7325    7328
                        7855    7858    7874    7877    7892    7964    7967    8036    8039    8124    8127    8175    8178
                        8286    8289    8310    8326
RAX16   002350          2781#   3462*   3463*   3531    4038    4048    4060    4070    6656    6660    6730    6734    6803
                        6807    6870    6873    6933    6936    7012    7016    7070    7076    7105    7109    7179    7183
RCVBUF  003240          3237#   4515
RCV1ST  007412          4120#   8389    8485    8534    8695    8768
RDALL = 000004          2507#
RDAX  = 000020          2493#   2565#   3445
RDRXSI  007332          4096#   4207    4427    8598
READAX  004102          3439#   3528    4035    6622    6710    6787    6854    6917    6991    7069    7104    7162    7295
                        7324    7854    7873    7891    7963    8035    8123    8174
READLU  003650          3348#   3399    3449    3457    3461    3500    3561    3585    3706    3783    3795    3880    3914
                        3938    3987    4098    4102    4141    4219    4464    5966    6008    6048    6088    6132    6180
```

```
                    6237    6288    6332    6376    6420    7250    7282    7503    7553    7566    7582    7605    7617
READY = 000200      2562#   3450    3501
REDBYT  002342      2778#   3360*   3361*   3400    3450    3458    3462    3501    3564    3574    3588    3598    3709
                    3719    3784    3796    3881    3917    3927    3941    3951    3990    4000    4099    4103    4142
                    4220    4467    4477    5967    5971    6009*   6010    6013    6049    6053    6089*   6090    6094
                    6133    6137    6181*   6182    6186    6238*   6241*   6243    6247    6289*   6290    6293    6333*
                    6334    6337    6377*   6378    6381    6421*   6422    6425    6497*   6498    6504    6526*   6527
                    6532    6554*   6555    6560    7251    7283    7504    7554    7567    7583    7606    7618
REDDAT  002456      2821#   6221    6231*   6232*   6242*   6243    6246    6697    6705*   6706*   6714    6717    6730
                    6733
REGNUM  002356      2784#   3323    3331*   3334*   3350    3376    3396    3398*   3402*   3403    3405*   3440    3442*
                    3456*   3460*   3464*   3478    3480*   3485*   3489*   3493*   3506*   3554    3560*   3584*   3607*
                    3613*   3650    3652*   3655*   3658*   3699    3705*   3728*   3734*   3752    3767*   3779*   3793*
                    3805*   3868    3871*   3879*   3891*   3907    3913*   3937*   3960*   3966*   3980    3986*   4009*
                    4015*   4096    4097*   4101*   4104*   4122    4125*   4157*   4173    4174*   4185*   4202    4218*
                    4230*   4240*   4347*   4352*   4367    4372*   4384*   4456    4461*   4496*   4891    5083    5180
                    5412    5521    5631    5964*   6000*   6044*   6078*   6124*   6125    6168*   6172*   6176*   6197*
                    6228*   6229*   6235*   6236*   6239    6281*   6325*   6369*   6413*   6485*   6605*   6625*   7056*
                    7151*   7154*   7247*   7278*   7479*   7502*   7552*   7835*   7926*   8016*   8086*   8167*   8219*
                    8223*   8330*   8333*   8706*
REG0    002466      2831#   6496*   6500*   6525*   6529*   6553*   6557*
REG1    002470      2832#
REG2    002472      2833#
REG3    002474      2834#
REG4    002476      2835#
REG5    002500      2836#
REG6    002502      2837#
REG7    002504      2838#
REOM  = 000002      2604#   4060    4070
RERR  = 000200      2598#
RETADR  002340      2777#   3609    3730    3962    4011    4081    4349    4492    7144*   7227*   7372*   7424*   7475*
                    7543*   7647*   7707*   7828*   7919*   8009*   8079*   8209*   8369*   8467*   8522*   8590*   8631*
                    8685*   8742*
RING  = 000200      2550#
ROMI  = 000002      2446#   3307    3309    3310    3422    3424
ROMO  = 000004      2445#   3307    3309    3310    3422    3424
ROR   = 000010      2602#
RRDYTO= 000001      2746#   3441    3454    6623
RSEOM   007060      4028#   4135    7940    7949    7956    7961    7982    8100    8109    8116    8121    8142    8256
                    8281    8306    8322
RSOM  = 000001      2605#   4038    4048
RTS   = 000040      2552#   3881    7504
RUN   = 000200      2441#   3325    3424    6083
RUNINH  002454      2818#   5800*   6473
RXABT = 002000      2696#
RXBCC = 000400      2698#   4519
RXCHAR  011052      4424#   8489    8493    8495    8497    8538    8542    8546    8550    8554    8558    8562    8566
                    8568
RXEBL = 001000      2697#
RXLEN0= 000001      2674#   4208    8479    8487    8540    8548    8556    8564
PXLEN1= 000002      2673#   4208    8483    8487    8540    8544    8556    8560
RXLEN2= 000004      2672#   4208    8479    8483    8487    8540    8544    8548    8552
RXOVR = 004000      2695#
RXWORD  002402      2794#   4099*   4100*   4103*   4211*   4212    4217    4241    4247    4252    4262    4275    4285
                    4298    4308    4321    4331
RX0   = 000001      2521#   2593#
```

```
RX1    = 000002          2520#   2592#
RX2    = 000004          2519#   2591#
RX3    = 000010          2518#   2590#
RX4    = 000020          2517#   2589#
RX5    = 000040          2516#   2588#
RX6    = 000100          2515#   2587#
RX7    = 000200          2514#   2586#
R14NRW   002536          2861#   6006    6009
SAVE4    002372          2790#   5736*   5739    5832    5946
SAVE6    002374          2791#   5737*   5740    5833    5947
SAVLEN   002410          2797#   3336*   4208    4381*   5733*
SCRACH   002320          2768#
SEC    = 000020          2647#
SECA   = 000020          2505#
SELFR  = 000040          2482#
SELSBY= 000002           2486#
SEL4     002432          2809#   5796*   5797*
SEL6     002434          2810#   3308*   3423*   5798*   5799*
SETUP    010616          4366#   7830    7921    8011    8081    8214
SFPTBL   002260 G        2327#
SIGQ   = 000100          2575#
SIGR   = 000200          2574#
SOM    = 000001          2468#
STALL    005142          3637#   3681    3683    3782    3787    3800
STARES   002422          2802#   5768*   5773*   6475
STARST   021234          5747    5753    5767#
STBY   = 000002          2556#
STEPLU= 000020           2443#   3680    3682    3781    3786    3799
STEPMP= 000001           2447#   3309    3310
STPCLK   003516          3306#   3358    3382    6130
STPERR   007624          4173#
STPLU    005232          3671#   3847    4137    4180    4344    4434    4462    7248    7280    7322    7500    7550    7564
                         7580    7603    7799    7845    7849    7867    7886    7894    7936    7942    7976    8026    8030
                         8048    8096    8102    8136    8237    8258    8273    8298    8314    8404    8416    8432    8'81
                         8596    8642    8709    8762
STR    = 000040          2646#
STRIP  = 000010          2506#   7375    7478    7546    7599    7791    7832    8013    8382    8634    8747
SUBRPC   002330          2772#   3555    3556    3558*   3559*   3612*   3700    3701    3703*   3704*   3733*   3754*   3755*
                         3807*   3827*   3828*   3854*   3869*   3870*   3890*   3908    3909    3911*   3912*   3965*   3981
                         3982    3984*   3985*   4014*   4029    4030    4032*   4033*   4084*   4123*   4124*   4160*   4203*
                         4204*   4354*   4386*   4425*   4426*   4441*   4458*   4459*   4497*   5062    5289    5500    5729*
SYSGBL= 000000           2098#   2104    2111#   2135    2144    2146    2148    2150    2152    2154    2156    2158    2160
                         2162    2164    2166    2168    2170    2172    2174    2176    2178    2181    2184    2186    2188
                         2190    2192    2194    2196    2198    2200    2202    2204    2206    2208    2210    2212    2214
                         2216    2218    2241    2304    2305    2326    2327    3260    3268    4859    4875    4950    5060
                         5169    5287    5396    5498    5615    5690    5708    5726    5818    5850    5869    5905    8818
                         8871    8903#   8904
SVCINS= 000001           2098#   2108#   2136    2137    2138    2139    2140    2141    2142    2143    2145    2147    2149
                         2151    2153    2155    2157    2159    2161    2163    2165    2167    2169    2171    2173    2175
                         2177    2179    2180    2182    2183    2185    2187    2189    2191    2193    2195    2197    2199
                         2201    2203    2205    2207    2209    2211    2213    2215    2217    2219    2240    2242    2243
                         2244    2245    2246    2247    2248    2249    2250    2251    2252    2253    2254    2255    2256
                         2257    2258    2259    2260    2261    2262    2263    2264    2265    2266    2267    2268    2269
                         2270    2271    2272    2273    2274    2275    2276    2277    2278    2279    2280    2281    2282
                         2283    2303    2325    3261    3262    3269    3276    3569    3570    3571    3572    3579    3580
                         3581    3582    3593    3594    3595    3596    3603    3604    3605    3606    3714    3715    3716
```

3717	3724	3725	3726	3727	3886	3887	3888	3889	3922	3923	3924	3925
3932	3933	3934	3935	3946	3947	3948	3949	3956	3957	3958	3959	3995
3996	3997	3998	4005	4006	4007	4008	4043	4044	4045	4046	4053	4054
4055	4056	4065	4066	4067	4068	4075	4076	4077	4078	4225	4226	4227
4228	4234	4235	4236	4237	4257	4258	4259	4260	4267	4268	4269	4270
4280	4281	4282	4283	4290	4291	4292	4293	4303	4304	4305	4306	4313
4314	4315	4316	4326	4327	4328	4329	4336	4337	4338	4339	4472	4473
4474	4475	4482	4483	4484	4485	4861	4862	4863	4864	4865	4866	4867
4870	4877	4878	4879	4880	4881	4882	4883	4885	4886	4887	4888	4889
4891	4892	4893	4894	4895	4896	4897	4899	4900	4901	4902	4903	4904
4905	4907	4908	4909	4910	4911	4912	4913	4915	4916	4917	4918	4919
4920	4921	4922	4923	4925	4926	4927	4928	4929	4930	4932	4933	4934
4935	4936	4937	4938	4939	4940	4943	4952	4953	4954	4955	4956	4957
4958	4960	4961	4962	4963	4964	4966	4967	4968	4969	4970	4971	4972
4974	4975	4976	4977	4978	4979	4980	4982	4983	4984	4985	4986	4987
4988	4990	4991	4992	4993	4994	4995	4996	4997	4998	5000	5001	5002
5003	5004	5005	5007	5008	5009	5010	5011	5012	5013	5014	5015	5017
5018	5019	5020	5021	5022	5023	5025	5026	5027	5028	5029	5030	5031
5032	5033	5035	5036	5037	5038	5039	5040	5042	5043	5044	5045	5046
5047	5048	5049	5050	5053	5062	5063	5064	5065	5066	5067	5069	5070
5071	5072	5073	5074	5075	5077	5078	5079	5080	5081	5083	5084	5085
5086	5087	5088	5089	5091	5092	5093	5094	5095	5096	5097	5099	5100
5101	5102	5103	5104	5105	5106	5107	5109	5110	5111	5112	5113	5114
5116	5117	5118	5119	5120	5121	5122	5123	5124	5126	5127	5128	5129
5130	5131	5132	5134	5135	5136	5137	5138	5139	5140	5141	5142	5144
5145	5146	5147	5148	5149	5151	5152	5153	5154	5155	5156	5157	5158
5159	5162	5171	5172	5173	5174	5175	5176	5177	5179	5180	5181	5182
5183	5184	5185	5187	5188	5189	5190	5191	5193	5194	5195	5196	5197
5198	5199	5201	5202	5203	5204	5205	5206	5207	5209	5210	5211	5212
5213	5214	5215	5217	5218	5219	5220	5221	5222	5223	5224	5225	5227
5228	5229	5230	5231	5232	5234	5235	5236	5237	5238	5239	5240	5241
5242	5244	5245	5246	5247	5248	5249	5250	5252	5253	5254	5255	5256
5257	5258	5259	5260	5262	5263	5264	5265	5266	5267	5269	5270	5271
5272	5273	5274	5275	5276	5277	5280	5289	5290	5291	5292	5293	5294
5296	5297	5298	5299	5300	5301	5302	5304	5305	5306	5307	5308	5310
5311	5312	5313	5314	5315	5316	5318	5319	5320	5321	5322	5323	5324
5326	5327	5328	5329	5330	5331	5332	5333	5334	5336	5337	5338	5339
5340	5341	5343	5344	5345	5346	5347	5348	5349	5350	5351	5353	5354
5355	5356	5357	5358	5359	5361	5362	5363	5364	5365	5366	5367	5368
5369	5371	5372	5373	5374	5375	5376	5378	5379	5380	5381	5382	5383
5394	5385	5386	5389	5398	5399	5400	5401	5402	5403	5404	5406	5407
5408	5409	5410	5412	5413	5414	5415	5416	5417	5418	5420	5421	5422
5423	5424	5425	5426	5428	5429	5430	5431	5432	5433	5434	5435	5436
5438	5439	5440	5441	5442	5443	5445	5446	5447	5448	5449	5450	5451
5452	5453	5455	5456	5457	5458	5459	5460	5461	5463	5464	5465	5466
5467	5468	5469	5470	5471	5473	5474	5475	5476	5477	5478	5480	5481
5482	5483	5484	5485	5486	5487	5488	5491	5500	5501	5502	5503	5504
5505	5507	5508	5509	5510	5511	5512	5513	5515	5516	5517	5518	5519
5521	5522	5523	5524	5525	5526	5527	5529	5530	5531	5532	5533	5534
5535	5537	5538	5539	5540	5541	5542	5543	5545	5546	5547	5548	5549
5550	5551	5552	5553	5555	5556	5557	5558	5559	5560	5562	5563	5564
5565	5566	5567	5568	5569	5570	5572	5573	5574	5575	5576	5577	5578
5580	5581	5582	5583	5584	5585	5586	5587	5588	5590	5591	5592	5593
5594	5595	5597	5598	5599	5600	5601	5602	5603	5604	5605	5608	5617
5618	5619	5620	5621	5622	5623	5625	5626	5627	5628	5629	5631	5632
5633	5634	5635	5636	5637	5639	5640	5641	5642	5643	5644	5645	5647

```
                5648  5649  5650  5651  5652  5653  5654  5655  5657  5658  5659  5660  5661
                5662  5664  5665  5666  5667  5668  5669  5670  5671  5672  5675  5694  5744
                5745  5747  5750  5751  5753  5756  5757  5759  5762  5763  5765  5782  5783
                5784  5786  5804  5821  5822  5830  5831  5836  5855  5872  5875  5876  5877
                5878  5879  5880  5883  5908  5932  5933  5942  5943  5944  5945  5950  5975
                5976  5977  5978  5982  6004  6017  6018  6019  6020  6024  6030  6057  6058
                6059  6060  6064  6082  6098  6099  6100  6101  6105  6141  6142  6143  6144
                6148  6190  6191  6192  6193  6195  6196  6203  6251  6252  6253  6254  6256
                6257  6266  6285  6297  6298  6299  6300  6304  6310  6329  6341  6342  6343
                6344  6348  6354  6373  6385  6386  6387  6388  6392  6398  6417  6429  6430
                6431  6432  6436  6442  6478  6479  6480  6481  6482  6483  6508  6509  6510
                6511  6514  6515  6516  6517  6518  6536  6537  6538  6539  6542  6543  6544
                6545  6546  6564  6565  6566  6567  6570  6571  6572  6573  6574  6579  6609
                6610  6611  6612  6614  6615  6629  6630  6631  6632  6634  6635  6647  6648
                6649  6650  6652  6653  6664  6665  6666  6667  6669  6670  6677  6722  6723
                6724  6725  6727  6728  6738  6739  6740  6741  6743  6744  6753  6780  6795
                6796  6797  6798  6800  6801  6811  6812  6813  6814  6818  6824  6847  6862
                6863  6864  6865  6867  6868  6877  6878  6879  6880  6884  6890  6910  6925
                6926  6927  6928  6930  6931  6940  6941  6942  6943  6947  6953  6986  7004
                7005  7006  7007  7009  7010  7020  7021  7022  7023  7027  7034  7063  7066
                7080  7081  7082  7083  7087  7094  7100  7113  7114  7115  7116  7120  7123
                7147  7171  7172  7173  7174  7176  7177  7187  7188  7189  7190  7192  7193
                7200  7203  7303  7304  7305  7306  7308  7309  7332  7333  7334  7335  7337
                7338  7350  7402  7451  7509  7510  7511  7512  7518  7559  7560  7561  7562
                7572  7573  7574  7575  7588  7589  7590  7591  7611  7612  7613  7614  7623
                7624  7625  7626  7630  7689  7772  7810  7862  7863  7864  7865  7881  7882
                7883  7884  7901  7971  7972  7973  7974  7991  8043  8044  8045  8046  8061
                8131  8132  8133  8134  8151  8182  8183  8184  8185  8190  8293  8294  8295
                8296  8343  8447  8502  8573  8608  8664  8717  8790  8817  8821  8822  8823
                8824  8826  8827  8828  8829  8830  8833  8870  8875  8900  8901  8902
SVCSUB= 000001   2098#  2110#  7062  7099  7146
SVCTAG= 000001   2098#  2112#  2311  2331  4869  4942  5052  5161  5279  5388  5490  5607  5674
                5693  5803  5835  5854  5882  5907  5949  5981  6023  6029  6063  6104  6147
                6202  6265  6303  6309  6347  6353  6391  6397  6435  6441  6578  6676  6752
                6817  6823  6883  6889  6946  6952  7026  7033  7086  7093  7119  7122  7199
                7202  7349  7401  7450  7517  7629  7688  7771  7809  7900  7990  8060  8150
                8189  8342  8446  8501  8572  8607  8663  8716  8789  8834  8876
SVCTST= 000001   2098#  2109#  5929  5963  5998  6042  6076  6122  6166  6219  6279  6323  6367
                6411  6470  6596  6694  6775  6842  6905  6979  7052  7142  7226  7371  7423
                7474  7539  7646  7706  7788  7827  7918  8008  8078  8165  8208  8368  8466
                8521  8589  8630  8684  8741
SW0    = 000002  2532#
SW1    = 000004  2531#
SW2    = 000010  2530#
SW3    = 000040  2528#
SYNCH  = 000226  2639#  7374  7477  7545  7598  7790  7831  8012  8381  8592  8687  8690  8699
                8704
SYNPAT   035462  8743  8785  8792#
SYN0   = 000001  2638#
SYN1   = 000002  2637#
SYN2   = 000004  2636#
SYN3   = 000010  2635#
SYN4   = 000020  2634#
SYN5   = 000040  2633#
SYN6   = 000100  2632#
SYN7   = 000200  2631#
```

```
S$LSYM= 010000      2098#  2312#  2332#  4870#  4943#  5053#  5162#  5280#  5389#  5491#  5608#  5675#  5694#
                    5804#  5836#  5855#  5883#  5908#  5950#  5982#  6004#  6030#  6064#  6105#  6148#  6203#
                    6266#  6285#  6310#  6329#  6354#  6373#  6398#  6417#  6442#  6579#  6677#  6753#  6780#
                    6824#  6847#  6890#  6910#  6953#  6986#  7034#  7066#  7094#  7120#  7123#  7200#  7203#
                    7350#  7402#  7451#  7518#  7630#  7689#  7772#  7810#  7901#  7991#  8061#  8151#  8190#
                    8343#  8447#  8502#  8573#  8608#  8664#  8717#  8790#  8835#  8877#
TEOM  = 000002      2625#  8248   8271
TERR  = 000200      2622#
TEST  = 000001      2663#  6992
TESTMD= 000004      2579#
TIMFLG  002336      2776#
TMP0    002506      2841#  3520*  3523*  4966   5193   5310
TMP1    002510      2842#  3538*  3539*  4967   5194   5311
TMP2    002512      2843#
TMP3    002514      2844#
TMP4    002516      2845#
TMP5    002520      2846#
TMP6    002522      2847#
TMP7    002524      2848#
TSOM  = 000001      2626#
TSTCON  002452      2817#
TSTNUM  002420      2801#  6472*  6478
TXAB  = 000004      2624#
TXABT = 002000      2684#
TXCHAR  006100      3825#  7376   7379   7382   7385   7388   7391   7394   7428   7431   7434   7437   7440
                    7443   7482   7485   7488   7491   7494   7497   7547   7600   7655   7658   7663   7668
                    7673   7676   7679   7682   7711   7714   7717   7724   7729   7734   7739   7744   7749
                    7754   7759   7762   7765
TXDATA= 000040      2576#  4467   4477
TXEN  = 000100      2491#  2563#  6089   7480   8220   8222
TXEOM = 001000      2685#  3220   3221   3222   3223   3229   3230   7383   7386   7389   7392   7395   7435
                    7438   7441   7444   7489   7492   7495   7498   7674   7677   7680   7683   7760   7763
                    7766   8440   8638   8756
TXGA  = 000010      2623#
TXGOA = 004000      2683#  8638
TXLEN0= 000040      2671#  4208   7653   7666   7722   7732   7742   7752
TXLEN1= 000100      2670#  4208   7661   7666   7727   7732   7747   7752
TXLEN2= 000200      2669#  4208   7653   7661   7666   7737   7742   7747   7752
TXSOM = 000400      2686#  3213   3214   3763   7237   7577   8748
TXWORD  002400      2793#  3651*  3653   3656   3763*  3764*  3829*  4404*  7237*  7262*  7316*  7577*  7794*
                    7840*  7931*  8021*  8091*  8171*  8384*  8406*  8436*  8440*  8472*  8527*  8594*  8638*
                    8690*  8748*  8756*
TX0   = 000001      2459#  2617#
TX1   = 000002      2458#  2616#
TX2   = 000004      2457#  2615#
TX3   = 000010      2456#  2614#
TX4   = 000020      2455#  2613#
TX5   = 000040      2454#  2612#
TX6   = 000100      2453#  2611#
TX7   = 000200      2452#  2610#
T$ARGC= 000001      2136#  2137#  2138#  2139#  2140#  2141#  4861#  4867   4877#  4883   4885#  4889   4891#
                    4897   4899#  4905   4907#  4913   4915#  4923   4925#  4930   4932#  4940   4952#  4958
                    4960#  4964   4966#  4972   4974#  4980   4982#  4988   4990#  4998   5000#  5005   5007#
                    5015   5017#  5023   5025#  5033   5035#  5040   5042#  5050   5062#  5067   5069#  5075
                    5077#  5081   5083#  5089   5091#  5097   5099#  5107   5109#  5114   5116#  5124   5126#
                    5132   5134#  5142   5144#  5149   5151#  5159   5171#  5177   5179#  5185   5187#  5191
```

```
                    5193#  5199   5201#  5207   5209#  5215   5217#  5225   5227#  5232   5234#  5242   5244#
                    5250   5252#  5260   5262#  5267   5269#  5277   5289#  5294   5296#  5302   5304#  5308
                    5310#  5316   5318#  5324   5326#  5334   5336#  5341   5343#  5351   5353#  5359   5361#
                    5369   5371#  5376   5378#  5386   5398#  5404   5406#  5410   5412#  5418   5420#  5426
                    5428#  5436   5438#  5443   5445#  5453   5455#  5461   5463#  5471   5473#  5478   5480#
                    5488   5500#  5505   5507#  5513   5515#  5519   5521#  5527   5529#  5535   5537#  5543
                    5545#  5553   5555#  5560   5562#  5570   5572#  5578   5580#  5588   5590#  5595   5597#
                    5605   5617#  5623   5625#  5629   5631#  5637   5639#  5645   5647#  5655   5657#  5662
                    5664#  5672   5875#  5880   6478#  6483   6514#  6518   6542#  6546   6570#  6574
T$CODE= 001032       8821#  8826#
T$ERRN= 000032       2098#  3570#  3580#  3594#  3604#  3715#  3725#  3887#  3923#  3933#  3947#  3957#  3996#
                    4006#  4044#  4054#  4066#  4076#  4226#  4235#  4258#  4268#  4281#  4291#  4304#  4314#
                    4327#  4337#  4473#  4483#  5943#  5976#  6018#  6058#  6099#  6142#  6191#  6252#  6298#
                    6342#  6386#  6430#  6509#  6537#  6565#  6610#  6630#  6648#  6665#  6723#  6739#  6796#
                    6812#  6863#  6878#  6926#  6941#  7005#  7021#  7081#  7114#  7172#  7188#  7304#  7333#
                    7510#  7560#  7573#  7589#  7612#  7624#  7863#  7882#  7972#  8044#  8132#  8183#  8294#
T$EXCP= 000000       8821#  8825   8826#  8831
T$FLAG= 000040       6195#  6256#  6614#  6634#  6652#  6669#  6727#  6743#  6800#  6867#  6930#  7009#  7176#
                    7192#  7308#  7337#
T$GMAN= 000000       2098#
T$HILI= 000007       8821#  8824   8826#  8830
T$LAST= 000001       2098#  8901#
T$LOLI= 000000       8821#  8823   8826#  8829
T$LSYM= 010000       2098#  2312   2332   4870   4943   5053   5162   5280   5389   5491   5608   5675   5694
                    5804   5836   5855   5883   5908   5950   5982   6030   6064   6105   6148   6203   6266
                    6310   6354   6398   6442   6579   6677   6753   6824   6890   6953   7034   7094   7120
                    7123   7200   7203   7350   7402   7451   7518   7630   7689   7772   7810   7901   7991
                    8061   8151   8190   8343   8447   8502   8573   8608   8664   8717   8790   8835   8877
T$LTNO= 000052       8904#
T$NEST= 177777       2098#  2104#  2303#  2311#  2325#  2331#  4859#  4869#  4875#  4942#  4950#  5052#  5060#
                    5161#  5169#  5279#  5287#  5388#  5396#  5490#  5498#  5607#  5615#  5674#  5690#  5693#
                    5708#  5713#  5726#  5803#  5818#  5835#  5850#  5854#  5869#  5882#  5905#  5907#  5930#
                    5949#  5964#  5981#  5999#  6004#  6023#  6029#  6043#  6063#  6077#  6104#  6123#  6147#
                    6167#  6202#  6220#  6265#  6280#  6285#  6303#  6309#  6324#  6329#  6347#  6353#  6368#
                    6373#  6391#  6397#  6412#  6417#  6435#  6441#  6471#  6578#  6597#  6676#  6695#  6752#
                    6776#  6780#  6817#  6823#  6843#  6847#  6883#  6889#  6906#  6910#  6946#  6952#  6980#
                    6986#  7026#  7033#  7053#  7063#  7066#  7086#  7093#  7100#  7119#  7122#  7143#  7147#
                    7199#  7202#  7227#  7349#  7372#  7401#  7424#  7450#  7475#  7517#  7540#  7629#  7647#
                    7688#  7707#  7771#  7789#  7809#  7828#  7900#  7919#  7990#  8009#  8060#  8079#  8150#
                    8166#  8189#  8209#  8342#  8369#  8446#  8467#  8501#  8522#  8572#  8590#  8607#  8631#
                    8663#  8685#  8716#  8742#  8789#  8817#  8833#  8870#  8875#  8898#
T$NS0 = 000000       2104#  8898
T$NS1 = 000005       2303#  2311   2325#  2331   4859#  4869   4875#  4942   4950#  5052   5060#  5161   5169#
                    5279   5287#  5388   5396#  5490   5498#  5607   5615#  5674   5690#  5693   5708#  5713
                    5726#  5803   5818#  5835   5850#  5854   5869#  5882   5905#  5907   5930#  5949   5964#
                    5981   5999#  6029   6043#  6063   6077#  6104   6123#  6147   6167#  6202   6220#  6265
                    6280#  6309   6324#  6353   6368#  6397   6412#  6441   6471#  6578   6597#  6676   6695#
                    6752   6776#  6823   6843#  6889   6906#  6952   6980#  7033   7053#  7122   7143#  7202
                    7227#  7349   7372#  7401   7424#  7450   7475#  7517   7540#  7629   7647#  7688   7707#
                    7771   7789#  7809   7828#  7900   7919#  7990   8009#  8060   8079#  8150   8166#  8189
                    8209#  8342   8369#  8446   8467#  8501   8522#  8572   8590#  8607   8631#  8663   8685#
                    8716   8742#  8789   8817#  8833   8870#  8875
T$NS2 = 000002       6004#  6023   6285#  6303   6329#  6347   6373#  6391   6417#  6435   6780#  6817   6847#
                    6883   6910#  6946   6986#  7026   7063#  7093   7100#  7119   7147#  7199
T$NS3 = 000003       7066#  7086
T$PTNU= 000000       2098#
```

```
T$SAVL= 177777          2098#
T$SEGL= 177777          2098#  6004#  6023#  6025   6285#  6303#  6305   6329#  6347#  6349   6373#  6391#  6393
                        6417#  6435#  6437   6780#  6801   6817#  6819   6847#  6868   6883#  6885   6910#  6931
                        6946#  6948   6986#  7010   7026#  7028   7066#  7086#  7088
T$SEKO= 010000          6004#  6023   6285#  6303   6329#  6347   6373#  6391   6417#  6435   6780#  6801   6817
                        6847#  6868   6883   6910#  6931   6946   6986#  7010   7026   7066#  7086
T$SUBN= 000000          2098#  5929#  5963#  5998#  6042#  6076#  6122#  6166#  6219#  6279#  6323#  6367#  6411#
                        6470#  6596#  6694#  6775#  6842#  6905#  6979#  7052#  7062#  7099#  7142#  7146#  7226#
                        7371#  7423#  7474#  7539#  7646#  7706#  7788#  7827#  7918#  8008#  8078#  8165#  8208#
                        8368#  8466#  8521#  8589#  8630#  8684#  8741#
T$TAGL= 177777          2098#
T$TAGN= 010101          2098#  2303#  2325#  4859#  4875#  4950#  5060#  5169#  5287#  5396#  5498#  5615#  5690#
                        5708#  5726#  5818#  5850#  5869#  5905#  5930#  5964#  5999#  6043#  6077#  6123#  6167#
                        6220#  6280#  6324#  6368#  6412#  6471#  6597#  6695#  6776#  6843#  6906#  6980#  7053#
                        7063#  7100#  7143#  7147#  7227#  7372#  7424#  7475#  7540#  7647#  7707#  7789#  7828#
                        7919#  8009#  8079#  8166#  8209#  8369#  8467#  8522#  8590#  8631#  8685#  8742#  8817#
                        8870#
T$TEMP= 000000          2242#  2243#  2244#  2245#  2246#  2247#  2248#  2249#  2250#  2251#  2252#  2253#  2254#
                        2255#  2256#  2257#  2258#  2259#  2260#  2261#  2262#  2263#  2264#  2265#  2266#  2267#
                        2268#  2269#  2270#  2271#  2272#  2273#  2274#  2275#  2276#  2277#  2278#  2279#  2280#
                        2281#  2282#  2283#  2284#  2311#  2331#  4869#  4942#  5052#  5161#  5279#  5388#  5490#
                        5607#  5674#  5693#  5713#  5803#  5835#  5854#  5882#  5907#  5949#  5981#  6023#  6029#
                        6063#  6104#  6147#  6195#  6196   6202#  6256#  6257   6265#  6303#  6309#  6347#  6353#
                        6391#  6397#  6435#  6441#  6578#  6614#  6615   6634#  6635   6652#  6653   6669#  6670
                        6676#  6727#  6728   6743#  6744   6752#  6800#  6801#  6817#  6823#  6867#  6868#  6883#
                        6889#  6930#  6931#  6946#  6952#  7009#  7010#  7026#  7033#  7086#  7093#  7119#  7122#
                        7176#  7177   7192#  7193   7199#  7202#  7308#  7309   7337#  7338   7349#  7401#  7450#
                        7517#  7629#  7688#  7771#  7809#  7900#  7990#  8060#  8150#  8189#  8342#  8446#  8501#
                        8572#  8607#  8663#  8716#  8760#  8821#  8826#  8833#  8875#  8898#
T$TEST= 000052          2098#  5929#  5963#  5998#  6042#  6076#  6122#  6166#  6219#  6279#  6323#  6367#  6411#
                        6470#  6596#  6694#  6775#  6842#  6905#  6979#  7052#  7062   7099   7142#  7146   7226#
                        7371#  7423#  7474#  7539#  7646#  7706#  7788#  7827#  7918#  8008#  8078#  8165#  8208#
                        8368#  8466#  8521#  8589#  8630#  8684#  8741#  8904
T$TSTM= 177777          2098#  3569   3579   3593   3603   3714   3724   3886   3922   3932   3946   3956   3995
                        4005   4043   4053   4065   4075   4225   4234   4257   4267   4280   4290   4303   4313
                        4326   4336   4472   4482   4866   4870   4882   4888   4896   4904   4912   4922   4929
                        4939   4943   4957   4963   4971   4979   4987   4997   5004   5014   5022   5032   5039
                        5049   5053   5066   5074   5080   5088   5096   5106   5113   5123   5131   5141   5148
                        5158   5162   5176   5184   5190   5198   5206   5214   5224   5231   5241   5249   5259
                        5266   5276   5280   5293   5301   5307   5315   5323   5333   5340   5350   5358   5368
                        5375   5385   5389   5403   5409   5417   5425   5435   5442   5452   5460   5470   5477
                        5487   5491   5504   5512   5518   5526   5534   5542   5552   5559   5569   5577   5587
                        5594   5604   5608   5622   5628   5636   5644   5654   5661   5671   5675   5694   5745
                        5751   5757   5763   5783   5804   5822   5831   5836   5855   5872   5879   5883   5908
                        5933   5942   5950   5975   5982   6004   6017   6024   6030   6057   6064   6082   6098
                        6105   6141   6148   6190   6195   6203   6251   6256   6266   6285   6297   6304   6310
                        6329   6341   6348   6354   6373   6385   6392   6398   6417   6429   6436   6442   6482
                        6508   6517   6536   6545   6564   6573   6579   6609   6614   6629   6634   6647   6652
                        6664   6669   6677   6722   6727   6738   6743   6753   6780   6795   6800   6811   6818
                        6824   6847   6862   6867   6877   6884   6890   6910   6925   6930   6940   6947   6953
                        6986   7004   7009   7020   7027   7034   7063   7066   7080   7087   7094   7100   7113
                        7120   7123   7147   7171   7176   7187   7192   7200   7203   7303   7308   7332   7337
                        7350   7402   7451   7509   7518   7559   7572   7588   7611   7623   7630   7689   7772
                        7810   7862   7881   7901   7971   7991   8043   8061   8131   8151   8182   8190   8293
                        8343   8447   8502   8573   8608   8664   8717   8790
T$TSTS= 000001          2098#  5930#  5964#  5999#  6043#  6077#  6123#  6167#  6220#  6280#  6324#  6368#  6412#
```

```
                        6471#   6597#   6695#   6776#   6843#   6906#   6980#   7053#   7143#   7227#   7372#   7424#   7475#
                        7540#   7647#   7707#   7789#   7828#   7919#   8009#   8079#   8166#   8209#   8369#   8467#   8522#
                        8590#   8631#   8685#   8742#
T$$AU  = 010021         5905#   5907
T$$AUT= 010016          5818#   5835
T$$CLE= 010017          5850#   5854
T$$DU  = 010020         5869#   5882
T$$HAR= 010077          8817#   8834
T$$HW  = 010000         2303#   2311
T$$INI= 010015          5726#   5803
T$$MSG= 010012          4859#   4869    4875#   4942    4950#   5052    5060#   5161    5169#   5279    5287#   5388    5396#
                        5490    5498#   5607    5615#   5674
T$$PRO= 010014          5708#
T$$RPT= 010013          5690#   5693
T$$SEG= 010000          6004#   6023#   6285#   6303#   6329#   6347#   6373#   6391#   6417#   6435#   6780#   6800    6817#
                        6847#   6867    6883#   6910#   6930    6946#   6986#   7009    7026#   7066#   7086#
T$$SOF= 010100          8870#   8876
T$$SUB= 010051          7063#   7093    7100#   7119    7147#   7176    7192    7199
T$$SW  = 010001         2325#   2331
T$$TES= 010076          5930#   5949    5964#   5981    5999#   6029    6043#   6063    6077#   6104    6123#   6147    6167#
                        6195    6202    6220#   6256    6265    6280#   6309    6324#   6353    6368#   6397    6412#   6441
                        6471#   6578    6597#   6614    6634    6652    6669    6676    6695#   6727    6743    6752    6776#
                        6823    6843#   6889    6906#   6952    6980#   7033    7053#   7122    7143#   7202    7227#   7308
                        7337    7349    7372#   7401    7424#   7450    7475#   7517    7540#   7629    7647#   7688    7707#
                        7771    7789#   7809    7828#   7900    7919#   7990    8009#   8060    8079#   8150    8166#   8189
                        8209#   8342    8369#   8446    8467#   8501    8522#   8572    8590#   8607    8631#   8663    8685#
                        8716    8742#   8789
T1      021560 G        2242    5929#
T10     023150 G        2251    6323#
T11     023266 G        2252    6367#
T12     023404 G        2253    6411#
T13     023522 G        2254    6470#
T14     024152 G        2255    6596#
T15     024504 G        2256    6694#
T16     024774 G        2257    6775#
T17     025204 G        2258    6842#
T18     025410 G        2259    6905#
T19     025614 G        2260    6979#
T2      021642 G        2243    5963#
T20     026052 G        2261    7052#
T20.1   026102          7062#
T20.2   026210          7099#
T21     026304 G        2262    7142#
T21.1   026316          7146#
T22     026556 G        2263    7226#
T23     027234 G        2264    7371#
T24     027354 G        2265    7423#
T25     027464 G        2266    7474#
T26     027660 G        2267    7539#
T27     030224 G        2268    7646#
T28     030434 G        2269    7706#
T29     030764 G        2270    7788#
T3      021726 G        2244    5998#
T30     031064 G        2271    7827#
T31     031434 G        2272    7918#
T32     031776 G        2273    8008#
```

```
T33      032246 G          2274    8078#
T34      032610 G          2275    8165#
T35      032740 G          2276    8208#
T36      033652 G          2277    8368#
T37      034134 G          2278    8466#
T38      034334 G          2279    8521#
T39      034634 G          2280    8589#
T4       022052 G          2245    6042#
T40      034724 G          2281    8630#
T41      035066 G          2282    8684#
T42      035234 G          2283    8741#
T5       022154 G          2246    6076#
T6       022302 G          2247    6122#
T7       022422 G          2248    6166#
T8       022576 G          2249    6219#
T9       023032 G          2250    6279#
UAM    = 000200 G          2417#
UNIT     002416            2800#
UNRR   = 000001            2533#   4220    7554    7567    7583    7606    7618
UPBITS   002526            2851#   6178    6232    6238    6289    6333    6377    6421
V35    = 000020            2659#   2664
WAIT50   005124            3623#   3773    3831    3874    4139    4408    4428    7158    7240    7264    7579    7798    7844
                           7935    8025    8095    8336    8476    8531    8641    8694    8759
WAX    = 000010            2494#   2566#   3496
WAX15    002352            2782#   3486*   3487    3762*   3764    4369*   4378*   6600*   6703*   6705    6782*   6784*   6788
                           6790    6849*   6851*   6855    6857    6912*   6914*   6918    6920    6988*   6992    7652*   7721*
                           8230*   8242*   8270*   8478*   8533*
WAX16    002354            2783#   3490*   3491    3765*   4370*   4380*   4381    6601*   6704*   6706    6783*   6785*   6850*
                           6852*   6870    6872    6913*   6915*   6933    6935    6989*   7653*   7661*   7666*   7671*   7722*
                           7727*   7732*   7737*   7742*   7747*   7752*   7757*   8231*   8243*   8248*   8271*   8479*   8483*
                           8487*   8491*   8536*   8540*   8544*   8548*   8552*   8556*   8560*   8564*
WRDYTO=  000002            2747#   3479    3505    6603
WRIBYT   002344            2779#   3332*   3380*   3381    3443*   3444*   3445*   3446*   3481*   3482*   3483*   3487*   3491*
                           3494*   3495*   3496*   3497*   3653*   3656*   3769*   3872*   4175*   4176*   4182*   4374*   6005*
                           6006*   6010    6012    6045*   6079*   6170*   6230*   6231    6286*   6290    6292    6330*   6334
                           6336    6374*   6378    6380    6418*   6422    6424    7067*   7101*   7152*   7155*   7480*   7837*
                           7928*   7984*   8018*   8054*   8088*   8144*   8168*   8220*   8224*   8331*   8334*   8707*
WRITAX   004270            3477#   3766    4371    4382    6602    6707    6786    6853    6916    6990    7654    7662    7667
                           7672    7723    7728    7733    7738    7743    7748    7753    7758    8232    8251    8272    8480
                           8484    8488    8492    8537    8541    8545    8549    8553    8557    8561    8565
WRITLU   003726            3333    3374#   3447    3484    3488    3492    3498    3654    3657    3770    3873    4177    4183
                           4375    6007    6046    6080    6171    6233    6287    6331    6375    6419    7068    7102    7153
                           7156    7157    7481    7838    7929    7985    8019    8055    8089    8145    8169    8221    8225
                           8332    8335    8708
XYZ    = 000100            2657#   2664
X$ALWA=  000000            2098#
X$FALS=  000040            2098#
X$OFFS=  000400            2098#
X$TRUE=  000020            2098#
$LSTIN=  000001            2106#
$LSTTA=  000001            2107#
.      = 036054            2089#   2763#   3237#   3276#   4852#   5890#   6196    6257    6615    6635    6653    6670    6728
                           6744    6801    6868    6931    7010    7177    7193    7309    7338    8850#   8889#
```

```
BCOMPL      1#   2098#  5746  5752  5758  5764  5785
BERROR      1#   2098#
BGNAU       1#   2098#  5904
BGNAUT      1#   2098#  5817
BGNCLN      1#   2098#  5849
BGNDU       1#   2098#  5868
BGNHRD      1#   2098#  8816
BGNHW       1#   2098#  2302
BGNINI      1#   2098#  5725
BGNMOD      1#   2098#  2103
BGNMSG      1#   2098#  4858  4874  4949  5059  5168  5286  5395  5497  5614
BGNPRO      1#   2098#  5707
BGNPTA      1#   2098#
BGNRPT      1#   2098#  5689
BGNSEG      1#   2098#  6003  6284  6328  6372  6416  6779  6846  6909  6985  7065
BGNSET      1#   2098#
BGNSFT      1#   2098#  8869
BGNSRV      1#   2098#
BGNSUB      1#   2098#  7061  7098  7145
BGNSW       1#   2098#  2324
BGNTST      1#   2098#  5928  5962  5997  6041  6075  6121  6165  6218  6278  6322  6366  6410  6469
         6595    6693   6774  6841  6904  6978  7051  7141  7225  7370  7422  7473  7538  7645  7705
         7787    7826   7917  8007  8077  8164  8207  8367  8465  8520  8588  8629  8683  8740
BNCOMP      1#   2098#
BNERRO      1#   2098#
BREAK       1#   2098#
BRESET      1#   2098#  5871  6081
CKLOOP      1#   2098#
CLOCK       1#   2098#
CLOSE       1#   2098#
CLRVEC      1#   2098#
COMMEN      1#   2098#
DELAY       1#   2098#
DESCRI      1#   2098#  3267
DEVTYP      1#   2098#  3259
DISPAT      1#   2098#  2239
DISPLA      1#   2098#
DOCLN       1#   2098#
DODU        1#   2098#  5829
DORPT       1#   2098#
ENDAU       1#   2098#  5906
ENDAUT      1#   2098#  5834
ENDCLN      1#   2098#  5853
ENDCOM      1#   2098#
ENDDU       1#   2098#  5881
ENDHRD      1#   2098#  8832
ENDHW       1#   2098#  2310
ENDINI      1#   2098#  5802
ENDMOD      1#   2098#  8897
ENDMSG      1#   2098#  4868  4941  5051  5160  5278  5387  5489  5606  5673
ENDPRO      1#   2098#  5712
ENDPTA      1#   2098#
ENDRPT      1#   2098#  5692
ENDSEG      1#   2098#  6022  6302  6346  6390  6434  6816  6882  6945  7025  7085
ENDSET      1#   2098#
ENDSFT      1#   2098#  8874
```

```
ENDSRV      1#  2098#
ENDSUB      1#  2098#  7092  7118  7198
ENDSW       1#  2098#  2330
ENDTST      1#  2098#  5948  5980  6028  6062  6103  6146  6201  6264  6308  6352  6396  6440  6577
         6675   6751   6822  6888  6951  7032  7121  7201  7348  7400  7449  7516  7628  7687  7770
         7808   7899   7989  8059  8149  8188  8341  8445  8500  8571  8606  8662  8715  8788
EQUALS      1#  2098#  2357
ERRDF       1#  2098#  3568  3578  3592  3602  3713  3723  3885  3921  3931  3945  3955  3994  4004
         4042   4052   4064  4074  4224  4233  4256  4266  4279  4289  4302  4312  4325  4335  4471
         4481   5941   5974  6016  6056  6097  6140  6189  6250  6296  6340  6384  6428  6507  6535
         6563   6608   6628  6646  6663  6721  6737  6794  6810  6861  6876  6924  6939  7003  7019
         7079   7112   7170  7186  7302  7331  7508  7558  7571  7587  7610  7622  7861  7880  7970
         8042   8130   8181  8292
ERRHRD      1#  2098#
ERROR       1#  2098#
ERRSF       1#  2098#
ERRSOF      1#  2098#
ERRTBL      1#  2098#
ESCAPE      1#  2098#  6194  6255  6613  6633  6651  6668  6726  6742  6799  6866  6929  7008  7175
         7191   7307   7336
EXIT        1#  2098#
FEQUAL      1#  2098#
GETBYT      1#  2098#
GETPRI      1#  2098#
GETWOR      1#  2098#
GMANIA      1#  2098#
GMANID      1#  2098#
GMANIL      1#  2098#
GPHARD      1#  2098#  5781
GPRMA       1#  2098#  8820
GPRMD       1#  2098#  8825
GPRML       1#  2098#
HEADER      1#  2098#  2134
INLOOP      1#  2098#
IOSETU      1#  2098#
IOSTAR      1#  2098#
KT11        1#  2098#
LASTAD      1#  2098#  8899
MANUAL      1#  2098#
MEMORY      1#  2098#
MSBYTE      1#  2098#  2135#  2141  2142  2143
MSCHEC      1#  2098#
MSCNTO      1#  2098#  8821#  8826#
MSCOUN      1#  2098#  4861#  4877#  4885#  4891#  4899#  4907#  4915#  4925#  4932#  4952#  4960#  4966#  4974#
         4982#  4990#  5000#  5007#  5017#  5025#  5035#  5042#  5062#  5069#  5077#  5083#  5091#  5099#  5109#
         5116#  5126#  5134#  5144#  5151#  5171#  5179#  5187#  5193#  5201#  5209#  5217#  5227#  5234#  5244#
         5252#  5262#  5269#  5289#  5296#  5304#  5310#  5318#  5326#  5336#  5343#  5353#  5361#  5371#  5378#
         5398#  5406#  5412#  5420#  5428#  5438#  5445#  5455#  5463#  5473#  5480#  5500#  5507#  5515#  5521#
         5529#  5537#  5545#  5555#  5562#  5572#  5580#  5590#  5597#  5617#  5625#  5631#  5639#  5647#  5657#
         5664#  5875#  6478#  6514#  6542#  6570#
MSDATA      1#  2098#  2135#  2144  2146  2148  2150  2152  2154  2156  2158  2160  2162  2164  2166
         2168   2170   2172  2174#  2176  2178  2181  2184  2186  2188  2190  2192  2194  2196  2198
         2200   2202   2204  2206  2208  2210  2212  2214  2216  2218  3260#  3268#
MSDECR      1#  2098#  2311#  2331#  4869#  4942#  5052#  5161#  5279#  5388#  5490#  5607#  5674#  5693#  5713#
         5803#  5835#  5854#  5882#  5907#  5949#  5981#  6023#  6029#  6063#  6104#  6147#  6202#  6265#  6303#
         6309#  6347#  6353#  6391#  6397#  6435#  6441#  6578#  6676#  6752#  6817#  6823#  6883#  6889#  6946#
```

```
          6952#  7026#  7033#  7086#  7093#  7119#  7122#  7199#  7202#  7349#  7401#  7450#  7517#  7629#  7688#
          7771#  7809#  7900#  7990#  8060#  8150#  8189#  8342#  8446#  8501#  8572#  8607#  8663#  8716#  8789#
          8833#  8875#  8898#
M$DEFA       1#  2098#  8821#  8826#
M$ENDE       1#  2098#  2311#  2331#  4869#  4942#  5052#  5161#  5279#  5388#  5490#  5607#  5674#  5693#  5803#
          5835#  5854#  5882#  5907#  5949#  5981#  6023#  6029#  6063#  6104#  6147#  6202#  6265#  6303#  6309#
          6347#  6353#  6391#  6397#  6435#  6441#  6578#  6676#  6752#  6817#  6823#  6883#  6889#  6946#  6952#
          7026#  7033#  7086#  7093#  7119#  7122#  7199#  7202#  7349#  7401#  7450#  7517#  7629#  7688#  7771#
          7809#  7900#  7990#  8060#  8150#  8189#  8342#  8446#  8501#  8572#  8607#  8663#  8716#  8789#  8833#
          8875#  8898#
M$ERRI       1#  2098#  3569#  3579#  3593#  3603#  3714#  3724#  3886#  3922#  3932#  3946#  3956#  3995#  4005#
          4043#  4053#  4065#  4075#  4225#  4234#  4257#  4267#  4280#  4290#  4303#  4313#  4326#  4336#  4472#
          4482#  5942#  5975#  6017#  6057#  6098#  6141#  6190#  6251#  6297#  6341#  6385#  6429#  6508#  6536#
          6564#  6609#  6629#  6647#  6664#  6722#  6738#  6795#  6811#  6862#  6877#  6925#  6940#  7004#  7020#
          7080#  7113#  7171#  7187#  7303#  7332#  7509#  7559#  7572#  7588#  7611#  7623#  7862#  7881#  7971#
          8043#  8131#  8182#  8293#
M$ESCA       1#  2098#  6195#  6196   6256#  6257   6614#  6615   6634#  6635   6652#  6653   6669#  6670   6727#
          6728   6743#  6744   6800#  6867#  6930#  7009#  7176#  7177   7192#  7193   7308#  7309   7337#  7338
M$ESCS       1#  2098#  6195#  6256#  6614#  6634#  6652#  6669#  6727#  6743#  6800#  6801   6867#  6868   6930#
          6931   7009#  7010   7176#  7192#  7308#  7337#
M$EXCP       1#  2098#  8821#  8826#
M$EXIT       1#  2098#
M$EXSE       1#  2098#
M$EXTJ       1#  2098#
M$GEN        1#  2098#  2104#  2135#  2144#  2146#  2148#  2150#  2152#  2154#  2156#  2158#  2160#  2162#  2164#
          2166#  2168#  2170#  2172#  2174#  2176#  2178#  2181#  2184#  2186#  2188#  2190#  2192#  2194#  2196#
          2198#  2200#  2202#  2204#  2206#  2208#  2210#  2212#  2214#  2216#  2218#  2241#  2304#  2305#  2311#
          2326#  2327#  2331#  3260#  3268#  4859#  4869#  4875#  4942#  4950#  5052#  5060#  5161#  5169#  5279#
          5287#  5388#  5396#  5490#  5498#  5607#  5615#  5674#  5690#  5693#  5708#  5726#  5803#  5816#  5835#
          5850#  5854#  5869#  5882#  5905#  5907#  5929#  5949#  5963#  5981#  5998#  6023#  6029#  6042#  6063#
          6076#  6104#  6122#  6147#  6166#  6202#  6219#  6265#  6279#  6303#  6309#  6323#  6347#  6353#  6367#
          6391#  6397#  6411#  6435#  6441#  6470#  6578#  6596#  6676#  6694#  6752#  6775#  6817#  6823#  6842#
          6883#  6889#  6905#  6946#  6952#  6979#  7026#  7033#  7052#  7062#  7086#  7093#  7099#  7119#  7122#
          7142#  7146#  7199#  7202#  7226#  7349#  7371#  7401#  7423#  7450#  7474#  7517#  7539#  7629#  7646#
          7688#  7706#  7771#  7788#  7809#  7827#  7900#  7918#  7990#  8008#  8060#  8078#  8150#  8165#  8189#
          8208#  8342#  8368#  8446#  8466#  8501#  8521#  8572#  8589#  8607#  8630#  8663#  8684#  8716#  8741#
          8789#  8818#  8834#  8871#  8876#  8903#
M$GENB       1#  2098#
M$GETS       1#  2098#  2311#  2331#  4869#  4942#  5052#  5161#  5279#  5388#  5490#  5607#  5674#  5693#  5713#
          5803#  5835#  5854#  5882#  5907#  5949#  5981#  6023#  6029#  6063#  6104#  6147#  6202#  6265#  6303#
          6309#  6347#  6353#  6391#  6397#  6435#  6441#  6578#  6676#  6752#  6801#  6817#  6823#  6868#  6883#
          6889#  6931#  6946#  6952#  7010#  7026#  7033#  7086#  7093#  7119#  7122#  7199#  7202#  7349#  7401#
          7450#  7517#  7629#  7688#  7771#  7809#  7900#  7990#  8060#  8150#  8189#  8342#  8446#  8501#  8572#
          8607#  8663#  8716#  8789#  8833#  8875#  8898#
M$GETT       1#  2098#  6195#  6256#  6614#  6634#  6652#  6669#  6727#  6743#  6800#  6801   6867#  6868   6930#
          6931   7009#  7010   7176#  7192#  7308#  7337#
M$GNGB       1#  2098#  2104#  2135#  2144#  2146#  2148#  2150#  2152#  2154#  2156#  2158#  2160#  2162#  2164#
          2166#  2168#  2170#  2172#  2174#  2176#  2178#  2181#  2184#  2186#  2188#  2190#  2192#  2194#  2196#
          2198#  2200#  2202#  2204#  2206#  2208#  2210#  2212#  2214#  2216#  2218#  2240#  2241   2303#  2304
          2305   2325#  2326   2327   3260#  3268#  4859#  4875#  4950#  5060#  5169#  5287#  5396#  5498#  5615#
          5690#  5708#  5726#  5818#  5850#  5869#  5905#  8817#  8818   8870#  8871   8900#  8903
M$GNIN       1#  2098#  2135#  2136   2137   2138   2139   2140   2141#  2142#  2143#  2144#  2145   2146#  2147
          2148#  2149   2150#  2151   2152#  2153   2154#  2155   2156#  2157   2158#  2159   2160#  2161   2162#
          2163   2164#  2165   2166#  2167   2168#  2169   2170#  2171   2172#  2173   2174#  2175   2176#  2177
          2178#  2179   2180   2181#  2182   2183#  2184#  2185   2186#  2187   2188#  2189   2190#  2191   2192#
          2193   2194#  2195   2196#  2197   2198#  2199   2200#  2201   2202#  2203   2204#  2205   2206#  2207
```

2208# 2209 2210# 2211 2212# 2213 2214# 2215 2216# 2217 2218# 2219 2240# 2242# 2243#
2244# 2245# 2246# 2247# 2248# 2249# 2250# 2251# 2252# 2253# 2254# 2255# 2256# 2257# 2258#
2259# 2260# 2261# 2262# 2263# 2264# 2265# 2266# 2267# 2268# 2269# 2270# 2271# 2272# 2273#
2274# 2275# 2276# 2277# 2278# 2279# 2280# 2281# 2282# 2283# 2303# 2325# 3260# 3261 3262
3268# 3269 3276 3569# 3570# 3571# 3572# 3579# 3580# 3581# 3582# 3593# 3594# 3595# 3596#
3603# 3604# 3605# 3606# 3714# 3715# 3716# 3717# 3724# 3725# 3726# 3727# 3886# 3887# 3888#
3889# 3922# 3923# 3924# 3925# 3932# 3933# 3934# 3935# 3946# 3947# 3948# 3949# 3956# 3957#
3958# 3959# 3995# 3996# 3997# 3998# 4005# 4006# 4007# 4008# 4043# 4044# 4045# 4046# 4053#
4054# 4055# 4056# 4065# 4066# 4067# 4068# 4075# 4076# 4077# 4078# 4225# 4226# 4227# 4228#
4234# 4235# 4236# 4237# 4257# 4258# 4259# 4260# 4267# 4268# 4269# 4270# 4280# 4281# 4282#
4283# 4290# 4291# 4292# 4293# 4303# 4304# 4305# 4306# 4313# 4314# 4315# 4316# 4326# 4327#
4328# 4329# 4336# 4337# 4338# 4339# 4472# 4473# 4474# 4475# 4482# 4483# 4484# 4485# 4861#
4862# 4863# 4864# 4865 4866# 4867 4870# 4877# 4878# 4879# 4880# 4881 4882# 4883 4885#
4886# 4887 4888# 4889 4891# 4892# 4893# 4894# 4895 4896# 4897 4899# 4900# 4901# 4902#
4903 4904# 4905 4907# 4908# 4909# 4910# 4911 4912# 4913 4915# 4916# 4917# 4918# 4919#
4920# 4921 4922# 4923 4925# 4926# 4927# 4928 4929# 4930 4932# 4933# 4934# 4935# 4936#
4937# 4938 4939# 4940 4943# 4952# 4953# 4954# 4955# 4956 4957# 4958 4960# 4961# 4962
4963# 4964 4966# 4967# 4968# 4969# 4970 4971# 4972 4974# 4975# 4976# 4977# 4978 4979#
4980 4982# 4983# 4984# 4985# 4986 4987# 4988 4990# 4991# 4992# 4993# 4994# 4995# 4996
4997# 4998 5000# 5001# 5002# 5003 5004# 5005 5007# 5008# 5009# 5010# 5011# 5012# 5013
5014# 5015 5017# 5018# 5019# 5020# 5021 5022# 5023 5025# 5026# 5027# 5028# 5029# 5030#
5031 5032# 5033 5035# 5036# 5037# 5038 5039# 5040 5042# 5043# 5044# 5045# 5046# 5047#
5048 5049# 5050 5053# 5062# 5063# 5064# 5065 5066# 5067 5069# 5070# 5071# 5072# 5073
5074# 5075 5077# 5078# 5079 5080# 5081 5083# 5084# 5085# 5086# 5087 5088# 5089 5091#
5092# 5093# 5094# 5095 5096# 5097 5099# 5100# 5101# 5102# 5103# 5104# 5105 5106# 5107
5109# 5110# 5111# 5112 5113# 5114 5116# 5117# 5118# 5119# 5120# 5121# 5122 5123# 5124
5126# 5127# 5128# 5129# 5130 5131# 5132 5134# 5135# 5136# 5137# 5138# 5139# 5140 5141#
5142 5144# 5145# 5146# 5147 5148# 5149 5151# 5152# 5153# 5154# 5155# 5156# 5157 5158#
5159 5162# 5171# 5172# 5173# 5174# 5175 5176# 5177 5179# 5180# 5181# 5182# 5183 5184#
5185 5187# 5188# 5189 5190# 5191 5193# 5194# 5195# 5196# 5197 5198# 5199 5201# 5202#
5203# 5204# 5205 5206# 5207 5209# 5210# 5211# 5212# 5213 5214# 5215 5217# 5218# 5219#
5220# 5221# 5222# 5223 5224# 5225 5227# 5228# 5229# 5230 5231# 5232 5234# 5235# 5236#
5237# 5238# 5239# 5240 5241# 5242 5244# 5245# 5246# 5247# 5248 5249# 5250 5252# 5253#
5254# 5255# 5256# 5257# 5258 5259# 5260 5262# 5263# 5264# 5265 5266# 5267 5269# 5270#
5271# 5272# 5273# 5274# 5275 5276# 5277 5280# 5289# 5290# 5291# 5292 5293# 5294 5296#
5297# 5298# 5299# 5300 5301# 5302 5304# 5305# 5306 5307# 5308 5310# 5311# 5312# 5313#
5314 5315# 5316 5318# 5319# 5320# 5321# 5322 5323# 5324 5326# 5327# 5328# 5329# 5330#
5331# 5332 5333# 5334 5336# 5337# 5338# 5339 5340# 5341 5343# 5344# 5345# 5346# 5347#
5348# 5349 5350# 5351 5353# 5354# 5355# 5356# 5357 5358# 5359 5361# 5362# 5363# 5364#
5365# 5366# 5367 5368# 5369 5371# 5372# 5373# 5374 5375# 5376 5378# 5379# 5380# 5381#
5382# 5383# 5384 5385# 5386 5389# 5398# 5399# 5400# 5401# 5402 5403# 5404 5406# 5407#
5408 5409# 5410 5412# 5413# 5414# 5415# 5416 5417# 5418 5420# 5421# 5422# 5423# 5424
5425# 5426 5428# 5429# 5430# 5431# 5432# 5433# 5434 5435# 5436 5438# 5439# 5440# 5441
5442# 5443 5445# 5446# 5447# 5448# 5449# 5450# 5451 5452# 5453 5455# 5456# 5457# 5458#
5459 5460# 5461 5463# 5464# 5465# 5466# 5467# 5468# 5469 5470# 5471 5473# 5474# 5475#
5476 5477# 5478 5480# 5481# 5482# 5483# 5484# 5485# 5486 5487# 5488 5491# 5500# 5501#
5502# 5503 5504# 5505 5507# 5508# 5509# 5510# 5511 5512# 5513 5515# 5516# 5517 5518#
5519 5521# 5522# 5523# 5524# 5525 5526# 5527 5529# 5530# 5531# 5532# 5533 5534# 5535
5537# 5538# 5539# 5540# 5541 5542# 5543 5545# 5546# 5547# 5548# 5549# 5550# 5551 5552#
5553 5555# 5556# 5557# 5558 5559# 5560 5562# 5563# 5564# 5565# 5566# 5567# 5568 5569#
5570 5572# 5573# 5574# 5575# 5576 5577# 5578 5580# 5581# 5582# 5583# 5584# 5585# 5586
5587# 5588 5590# 5591# 5592# 5593 5594# 5595 5597# 5598# 5599# 5600# 5601# 5602# 5603
5604# 5605 5608# 5617# 5618# 5619# 5620# 5621 5622# 5623 5625# 5626# 5627 5628# 5629
5631# 5632# 5633# 5634# 5635 5636# 5637 5639# 5640# 5641# 5642# 5643 5644# 5645 5647#
5648# 5649# 5650# 5651# 5652# 5653 5654# 5655 5657# 5658# 5659# 5660 5661# 5662 5664#
5665# 5666# 5667# 5668# 5669# 5670 5671# 5672 5675# 5694# 5744# 5745# 5747# 5750# 5751#

```
          5753#  5756#  5757#  5759#  5762#  5763#  5765#  5782#  5783#  5784#  5786#  5804#  5821#  5822#  5830#
          5831#  5836#  5855#  5872#  5875#  5876#  5877#  5878   5879#  5880   5883#  5908#  5932#  5933#  5942#
          5943#  5944#  5945#  5950#  5975#  5976#  5977#  5978#  5982#  6004#  6017#  6018#  6019#  6020#  6024#
          6030#  6057#  6058#  6059#  6060#  6064#  6082#  6098#  6099#  6100#  6101#  6105#  6141#  6142#  6143#
          6144#  6148#  6190#  6191#  6192#  6193#  6195#  6196#  6203#  6251#  6252#  6253#  6254#  6256#  6257#
          6266#  6285#  6297#  6298#  6299#  6300#  6304#  6310#  6329#  6341#  6342#  6343#  6344#  6348#  6354#
          6373#  6385#  6386#  6387#  6388#  6392#  6398#  6417#  6429#  6430#  6431#  6432#  6436#  6442#  6478#
          6479#  6480#  6481   6482#  6483   6508#  6509#  6510#  6511#  6514#  6515#  6516   6517#  6518   6536#
          6537#  6538#  6539#  6542#  6543#  6544   6545#  6546   6564#  6565#  6566#  6567#  6570#  6571#  6572
          6573#  6574   6579#  6609#  6610#  6611#  6612#  6614#  6615#  6629#  6630#  6631#  6632#  6634#  6635#
          6647#  6648#  6649#  6650#  6652#  6653#  6664#  6665#  6666#  6667#  6669#  6670#  6677#  6722#  6723#
          6724#  6725#  6727#  6728#  6738#  6739#  6740#  6741#  6743#  6744#  6753#  6780#  6795#  6796#  6797#
          6798#  6800#  6801#  6811#  6812#  6813#  6814#  6818#  6824#  6847#  6862#  6863#  6864#  6865#  6867#
          6868#  6877#  6878#  6879#  6880#  6884#  6890#  6910#  6925#  6926#  6927#  6928#  6930#  6931#  6940#
          6941#  6942#  6943#  6947#  6953#  6986#  7004#  7005#  7006#  7007#  7009#  7010#  7020#  7021#  7022#
          7023#  7027#  7034#  7063#  7066#  7080#  7081#  7082#  7083#  7087#  7094#  7100#  7113#  7114#  7115#
          7116#  7120#  7123#  7147#  7171#  7172#  7173#  7174#  7176#  7177#  7187#  7188#  7189#  7190#  7192#
          7193#  7200#  7203#  7303#  7304#  7305#  7306#  7308#  7309#  7332#  7333#  7334#  7335#  7337#  7338#
          7350#  7402#  7451#  7509#  7510#  7511#  7512#  7518#  7559#  7560#  7561#  7562#  7572#  7573#  7574#
          7575#  7588#  7589#  7590#  7591#  7611#  7612#  7613#  7614#  7623#  7624#  7625#  7626#  7630#  7689#
          7772#  7810#  7862#  7863#  7864#  7865#  7881#  7882#  7883#  7884#  7901#  7971#  7972#  7973#  7974#
          7991#  8043#  8044#  8045#  8046#  8061#  8131#  8132#  8133#  8134#  8151#  8182#  8183#  8184#  8185#
          8190#  8293#  8294#  8295#  8296#  8343#  8447#  8502#  8573#  8608#  8664#  8717#  8790#  8817#  8821#
          8822   8823   8824   8826#  8827   8828   8829   8830   8833#  8870#  8875#  8900#  8901#  8902#
M$GNLS       1#  2098#  6023#  6303#  6347#  6391#  6435#  6817#  6883#  6946#  7026#  7086#
M$GNSU       1#  2098#  7062#  7099#  7146#
M$GNTA       1#  2098#  2311#  2331#  4869#  4942#  5052#  5161#  5279#  5388#  5490#  5607#  5674#  5693#  5803#
          5835#  5854#  5882#  5907#  5949#  5981#  6029#  6063#  6104#  6147#  6202#  6265#  6309#  6353#  6397#
          6441#  6578#  6676#  6752#  6823#  6889#  6952#  7033#  7093#  7119#  7122#  7199#  7202#  7349#  7401#
          7450#  7517#  7629#  7688#  7771#  7809#  7900#  7990#  8060#  8150#  8189#  8342#  8446#  8501#  8572#
          8607#  8663#  8716#  8789#  8833#  8834   8875#  8876
M$GNTE       1#  2098#  5929#  5963#  5998#  6042#  6076#  6122#  6166#  6219#  6279#  6323#  6367#  6411#  6470#
          6596#  6694#  6775#  6842#  6905#  6979#  7052#  7142#  7226#  7371#  7423#  7474#  7539#  7646#  7706#
          7788#  7827#  7918#  8008#  8078#  8165#  8208#  8368#  8466#  8521#  8589#  8630#  8684#  8741#
M$HAPT       1#  2098#  2135#
M$HNAP       1#  2098#  2135#  2174
M$INCR       1#  2098#  2104#  2303#  2325#  3569#  3579#  3593#  3603#  3714#  3724#  3886#  3922#  3932#  3946#
          3956#  3995#  4005#  4043#  4053#  4065#  4075#  4225#  4234#  4257#  4267#  4280#  4290#  4303#  4313#
          4326#  4336#  4472#  4482#  4859#  4866#  4870#  4875#  4882#  4888#  4896#  4904#  4912#  4922#  4929#
          4939#  4943#  4950#  4957#  4963#  4971#  4979#  4987#  4997#  5004#  5014#  5022#  5032#  5039#  5049#
          5053#  5060#  5066#  5074#  5080#  5088#  5096#  5106#  5113#  5123#  5131#  5141#  5148#  5158#  5162#
          5169#  5176#  5184#  5190#  5198#  5206#  5214#  5224#  5231#  5241#  5249#  5259#  5266#  5276#  5280#
          5287#  5293#  5301#  5307#  5315#  5323#  5333#  5340#  5350#  5358#  5368#  5375#  5385#  5389#  5396#
          5403#  5409#  5417#  5425#  5435#  5442#  5452#  5460#  5470#  5477#  5487#  5491#  5498#  5504#  5512#
          5518#  5526#  5534#  5542#  5552#  5559#  5569#  5577#  5587#  5594#  5604#  5608#  5615#  5622#  5628#
          5636#  5644#  5654#  5661#  5671#  5675#  5690#  5694#  5708#  5726#  5745#  5751#  5757#  5763#  5783#
          5804#  5818#  5822#  5831#  5836#  5850#  5855#  5869#  5872#  5879#  5883#  5905#  5908#  5929#  5930#
          5933#  5942#  5950#  5963#  5964#  5975#  5982#  5998#  5999#  6004#  6017#  6024#  6030#  6042#  6043#
          6057#  6064#  6076#  6077#  6082#  6098#  6105#  6122#  6123#  6141#  6148#  6166#  6167#  6190#  6195#
          6203#  6219#  6220#  6251#  6256#  6266#  6279#  6280#  6285#  6297#  6304#  6310#  6323#  6324#  6329#
          6341#  6348#  6354#  6367#  6368#  6373#  6385#  6392#  6398#  6411#  6412#  6417#  6429#  6436#  6442#
          6470#  6471#  6482#  6508#  6517#  6536#  6545#  6564#  6573#  6579#  6596#  6597#  6609#  6614#  6629#
          6634#  6647#  6652#  6664#  6669#  6677#  6694#  6695#  6722#  6727#  6738#  6743#  6753#  6775#  6776#
          6780#  6795#  6800#  6811#  6818#  6824#  6842#  6843#  6847#  6862#  6867#  6877#  6884#  6890#  6905#
          6906#  6910#  6925#  6930#  6940#  6947#  6953#  6979#  6980#  6986#  7004#  7009#  7020#  7027#  7034#
          7052#  7053#  7062#  7063#  7066#  7080#  7087#  7094#  7099#  7100#  7113#  7120#  7123#  7142#  7143#
```

```
                7146#  7147#  7171#  7176#  7187#  7192#  7200#  7203#  7226#  7227#  7303#  7308#  7332#  7337#  7350#
                7371#  7372#  7402#  7423#  7424#  7451#  7474#  7475#  7509#  7518#  7539#  7540#  7559#  7572#  7588#
                7611#  7623#  7630#  7646#  7647#  7689#  7706#  7707#  7772#  7788#  7789#  7810#  7827#  7828#  7862#
                7881#  7901#  7918#  7919#  7971#  7991#  8008#  8009#  8043#  8061#  8078#  8079#  8131#  8151#  8165#
                8166#  8182#  8190#  8208#  8209#  8293#  8343#  8368#  8369#  8447#  8466#  8467#  8502#  8521#  8522#
                8573#  8589#  8590#  8608#  8630#  8631#  8664#  8684#  8685#  8717#  8741#  8742#  8790#  8817#  8870#
M$IOSE             1#  2098#
M$LDRO             1#  2098#  5744#  5750#  5756#  5762#  5782#  5821#  5830#  5932#
M$MASK             1#  2098#
M$MCHI             1#  2098#
M$MCLO             1#  2098#
M$MSK1             1#  2098#
M$POP              1#  2098#  2311#  2331#  4869#  4942#  5052#  5161#  5279#  5388#  5490#  5607#  5674#  5693#  5713#
                5803#  5835#  5854#  5882#  5907#  5949#  5981#  6023#  6029#  6063#  6104#  6147#  6202#  6265#  6303#
                6309#  6347#  6353#  6391#  6397#  6435#  6441#  6578#  6676#  6752#  6817#  6823#  6883#  6889#  6946#
                6952#  7026#  7033#  7086#  7093#  7119#  7122#  7199#  7202#  7349#  7401#  7450#  7517#  7629#  7688#
                7771#  7809#  7900#  7990#  8060#  8150#  8189#  8342#  8446#  8501#  8572#  8607#  8663#  8716#  8789#
                8833#  8875#  8898#
M$PRIN             1#  2098#  4861#  4877#  4885#  4891#  4899#  4907#  4915#  4925#  4932#  4952#  4960#  4966#  4974#
                4982#  4990#  5000#  5007#  5017#  5025#  5035#  5042#  5062#  5069#  5077#  5083#  5091#  5099#  5109#
                5116#  5126#  5134#  5144#  5151#  5171#  5179#  5187#  5193#  5201#  5209#  5217#  5227#  5234#  5244#
                5252#  5262#  5269#  5289#  5296#  5304#  5310#  5318#  5326#  5336#  5343#  5353#  5361#  5371#  5378#
                5398#  5406#  5412#  5420#  5428#  5438#  5445#  5455#  5463#  5473#  5480#  5500#  5507#  5515#  5521#
                5529#  5537#  5545#  5555#  5562#  5572#  5580#  5590#  5597#  5617#  5625#  5631#  5639#  5647#  5657#
                5664#  5875#  6478#  6514#  6542#  6570#
M$PUSH             1#  2098#  2104#  2303#  2325#  4859#  4875#  4950#  5060#  5169#  5287#  5396#  5498#  5615#  5690#
                5708#  5726#  5818#  5850#  5869#  5905#  5929#  5930   5963#  5964   5998#  5999   6004#  6042#  6043
                6076#  6077   6122#  6123   6166#  6167   6219#  6220   6279#  6280   6285#  6323#  6324   6329#  6367#
                6368   6373#  6411#  6412   6417#  6470#  6471   6596#  6597   6694#  6695   6775#  6776   6780#  6842#
                6843   6847#  6905#  6906   6910#  6979#  6980   6966#  7052#  7053   7062#  7063   7066#  7099#  7100
                7142#  7143   7146#  7147   7226#  7227   7371#  7372   7423#  7424   7474#  7475   7539#  7540   7646#
                7647   7706#  7707   7788#  7789   7827#  7828   7918#  7919   8008#  8009   8078#  8079   8165#  8166
                8208#  8209   8368#  8369   8466#  8467   8521#  8522   8589#  8590   8630#  8631   8684#  8685   8741#
                8742   8817#  8870#
M$PUT              1#  2098#  4861#  4877#  4885#  4891#  4899#  4907#  4915#  4925#  4932#  4952#  4960#  4966#  4974#
                4982#  4990#  5000#  5007#  5017#  5025#  5035#  5042#  5062#  5069#  5077#  5083#  5091#  5099#  5109#
                5116#  5126#  5134#  5144#  5151#  5171#  5179#  5187#  5193#  5201#  5209#  5217#  5227#  5234#  5244#
                5252#  5262#  5269#  5289#  5296#  5304#  5310#  5318#  5326#  5336#  5343#  5353#  5361#  5371#  5378#
                5398#  5406#  5412#  5420#  5428#  5438#  5445#  5455#  5463#  5473#  5480#  5500#  5507#  5515#  5521#
                5529#  5537#  5545#  5555#  5562#  5572#  5580#  5590#  5597#  5617#  5625#  5631#  5639#  5647#  5657#
                5664#  5875#  6478#  6514#  6542#  6570#
M$PUT1             1#  2098#  4861#  4862   4863   4864   4877#  4878   4879   4880   4885#  4886   4891#  4892   4893
                4894   4899#  4900   4901   4902   4907#  4908   4909   4910   4915#  4916   4917   4918   4919   4920
                4925#  4926   4927   4932#  4933   4934   4935   4936   4937   4952#  4953   4954   4955   4960#  4961
                4966#  4967   4968   4969   4974#  4975   4976   4977   4982#  4983   4984   4985   4990#  4991   4992
                4995   4994   4995   5000#  5001   5002   5007#  5008   5009   5010   5011   5012   5017#  5018   5019
                5020   5025#  5026   5027   5028   5029   5030   5035#  5036   5037   5042#  5043   5044   5045   5046
                5047   5062#  5063   5064   5069#  5070   5071   5072   5077#  5078   5083#  5084   5085   5086   5091#
                5092   5093   5094   5099#  5100   5101   5102   5103   5104   5109#  5110   5111   5116#  5117   5118
                5119   5120   5121   5126#  5127   5128   5129   5134#  5135   5136   5137   5138   5139   5144#  5145
                5146   5151#  5152   5153   5154   5155   5156   5171#  5172   5173   5174   5179#  5180   5181   5182
                5187#  5188   5193#  5194   5195   5196   5201#  5202   5203   5204   5209#  5210   5211   5212   5217#
                5218   5219   5220   5221   5222   5227#  5228   5229   5234#  5235   5236   5237   5238   5239   5244#
                5245   5246   5247   5252#  5253   5254   5255   5256   5257   5262#  5263   5264   5269#  5270   5271
                5272   5273   5274   5289#  5290   5291   5296#  5297   5298   5299   5304#  5305   5310#  5311   5312
                5313   5318#  5319   5320   5321   5326#  5327   5328   5329   5330   5331   5336#  5337   5338   5343#
```

```
                5344    5345    5346    5347    5348    5353#   5354    5355    5356    5361#   5362    5363    5364    5365    5366
                5371#   5372    5373    5378#   5379    5380    5381    5382    5383    5398#   5399    5400    5401    5406#   5407
                5412#   5413    5414    5415    5420#   5421    5422    5423    5428#   5429    5430    5431    5432    5433    5438#
                5439    5440    5445#   5446    5447    5448    5449    5450    5455#   5456    5457    5458    5463#   5464    5465
                5466    5467    5468    5473#   5474    5475    5480#   5481    5482    5483    5484    5485    5500#   5501    5502
                5507#   5508    5509    5510    5515#   5516    5521#   5522    5523    5524    5529#   5530    5531    5532    5537#
                5538    5539    5540    5545#   5546    5547    5548    5549    5550    5555#   5556    5557    5562#   5563    5564
                5565    5566    5567    5572#   5573    5574    5575    5580#   5581    5582    5583    5584    5585    5590#   5591
                5592    5597#   5598    5599    5600    5601    5602    5617#   5618    5619    5620    5625#   5626    5631#   5632
                5633    5634    5639#   5640    5641    5642    5647#   5648    5649    5650    5651    5652    5657#   5658    5659
                5664#   5665    5666    5667    5668    5669    5875#   5876    5877    6478#   6479    6480    6514#   6515    6542#
                6543    6570#   6571
M$RADI             1#   2098#   8821#   8826#
M$RBRO             1#   2098#
M$RNRO             1#   2098#   5782#   5784
M$SETS             1#   2098#   2104#   2303#   2325#   4859#   4875#   4950#   5060#   5169#   5287#   5396#   5498#   5615#   5690#
             5708#   5726#   5818#   5850#   5869#   5905#   5930#   5964#   5999#   6004#   6043#   6077#   6123#   6167#   6220#
             6280#   6285#   6324#   6329#   6368#   6373#   6412#   6417#   6471#   6597#   6695#   6776#   6780#   6843#   6847#
             6906#   6910#   6980#   6986#   7053#   7063#   7066#   7100#   7143#   7147#   7227#   7372#   7424#   7475#   7540#
             7647#   7707#   7789#   7828#   7919#   8009#   8079#   8166#   8209#   8369#   8467#   8522#   8590#   8631#   8685#
             8742#   8817#   8870#
M$STAR             1#   2098#
M$SVC              1#   2098#   3569    3579    3593    3603    3714    3724    3886    3922    3932    3946    3956    3995    4005
             4043    4053    4065    4075    4225    4234    4257    4267    4280    4290    4303    4313    4326    4336    4472
             4482    4861#   4866    4869#   4870    4877#   4882    4885#   4888    4891#   4896    4899#   4904    4907#   4912
             4915#   4922    4925#   4929    4932#   4939    4942#   4943    4952#   4957    4960#   4963    4966#   4971    4974#
             4979    4982#   4987    4990#   4997    5000#   5004    5007#   5014    5017#   5022    5025#   5032    5035#   5039
             5042#   5049    5052#   5053    5062#   5066    5069#   5074    5077#   5080    5083#   5088    5091#   5096    5099#
             5106    5109#   5113    5116#   5123    5126#   5131    5134#   5141    5144#   5148    5151#   5158    5161#   5162
             5171#   5176    5179#   5184    5187#   5190    5193#   5198    5201#   5206    5209#   5214    5217#   5224    5227#
             5231    5234#   5241    5244#   5249    5252#   5259    5262#   5266    5269#   5276    5279#   5280    5289#   5293
             5296#   5301    5304#   5307    5310#   5315    5318#   5323    5326#   5333    5336#   5340    5343#   5350    5353#
             5358    5361#   5368    5371#   5375    5378#   5385    5388#   5389    5398#   5403    5406#   5409    5412#   5417
             5420#   5425    5428#   5435    5438#   5442    5445#   5452    5455#   5460    5463#   5470    5473#   5477    5480#
             5487    5490#   5491    5500#   5504    5507#   5512    5515#   5518    5521#   5526    5529#   5534    5537#   5542
             5545#   5552    5555#   5559    5562#   5569    5572#   5577    5580#   5587    5590#   5594    5597#   5604    5607#
             5608    5617#   5622    5625#   5628    5631#   5636    5639#   5644    5647#   5654    5657#   5661    5664#   5671
             5674#   5675    5693#   5694    5744#   5745    5750#   5751    5756#   5757    5762#   5763    5782#   5783    5803#
             5804    5821#   5822    5830#   5831    5835#   5836    5854#   5855    5872#   5875#   5879    5882#   5883    5907#
             5908    5932#   5933    5942    5949#   5950    5975    5981#   5982    6004#   6017    6023#   6024    6029#   6030
             6057    6063#   6064    6082#   6098    6104#   6105    6141    6147#   6148    6190    5195#   6202#   6203    6251
             6256#   6265#   6266    6285#   6297    6303#   6304    6309#   6310    6329#   6341    6347#   6348    6353#   6354
             6373#   6385    6391#   6392    6397#   6398    6417#   6429    6435#   6436    6441#   6442    6478#   6482    6508
             6514#   6517    6536    6542#   6545    6564    6570#   6573    6578#   6579    6609    6614#   6629    6634#   6647
             6652#   6664    6669#   6676#   6677    6722    6727#   6738    6743#   6752#   6753    6780#   6795    6800#   6811
             6817#   6818    6823#   6824    6847#   6862    6867#   6877    6883#   6884    6889#   6890    6910#   6925    6930#
             6940    6946#   6947    6952#   6953    6986#   7004    7009#   7020    7026#   7027    7033#   7034    7062#   7063
             7066#   7080    7086#   7087    7093#   7094    7099#   7100    7113    7119#   7120    7122#   7123    7146#   7147
             7171    7176#   7187    7192#   7199#   7200    7202#   7203    7303    7308#   7332    7337#   7349#   7350    7401#
             7402    7450#   7451    7509    7517#   7518    7559    7572    7588    7611    7623    7629#   7630    7688#   7689
             7771#   7772    7809#   7810    7862    7881    7900#   7901    7971    7990#   7991    8043    8060#   8061    8131
             8150#   8151    8182    8189#   8190    8293    8342#   8343    8446#   8447    8501#   8502    8572#   8573    8607#
             8608    8663#   8664    8716#   8717    8789#   8790
M$TLAB             1#   2098#   3569#   3579#   3593#   3603#   3714#   3724#   3886#   3922#   3932#   3946#   3956#   3995#   4005#
             4043#   4053#   4065#   4075#   4225#   4234#   4257#   4267#   4280#   4290#   4303#   4313#   4326#   4336#   4472#
             4482#   4866#   4870#   4882#   4888#   4896#   4904#   4912#   4922#   4929#   4939#   4943#   4957#   4963#   4971#
```

```
                4979#  4987#  4997#  5004#  5014#  5022#  5032#  5039#  5049#  5053#  5066#  5074#  5080#  5088#  5096#
                5106#  5113#  5123#  5131#  5141#  5148#  5158#  5162#  5176#  5184#  5190#  5198#  5206#  5214#  5224#
                5231#  5241#  5249#  5259#  5266#  5276#  5280#  5293#  5301#  5307#  5315#  5323#  5333#  5340#  5350#
                5358#  5368#  5375#  5385#  5389#  5403#  5409#  5417#  5425#  5435#  5442#  5452#  5460#  5470#  5477#
                5487#  5491#  5504#  5512#  5518#  5526#  5534#  5542#  5552#  5559#  5569#  5577#  5587#  5594#  5604#
                5608#  5622#  5628#  5636#  5644#  5654#  5661#  5671#  5675#  5694#  5745#  5751#  5757#  5763#  5783#
                5804#  5822#  5831#  5836#  5855#  5872#  5879#  5883#  5908#  5933#  5942#  5950#  5975#  5982#  6004#
                6017#  6024#  6030#  6057#  6064#  6082#  6098#  6105#  6141#  6148#  6190#  6195#  6203#  6251#  6256#
                6266#  6285#  6297#  6304#  6310#  6329#  6341#  6348#  6354#  6373#  6385#  6392#  6398#  6417#  6429#
                6436#  6442#  6482#  6508#  6517#  6536#  6545#  6564#  6573#  6579#  6609#  6614#  6629#  6634#  6647#
                6652#  6664#  6669#  6677#  6722#  6727#  6738#  6743#  6753#  6780#  6795#  6800#  6811#  6818#  6824#
                6847#  6862#  6867#  6877#  6884#  6890#  6910#  6925#  6930#  6940#  6947#  6953#  6986#  7004#  7009#
                7020#  7027#  7034#  7063#  7066#  7080#  7087#  7094#  7100#  7113#  7120#  7123#  7147#  7171#  7176#
                7187#  7192#  7200#  7203#  7303#  7308#  7332#  7337#  7350#  7402#  7451#  7509#  7518#  7559#  7572#
                7588#  7611#  7623#  7630#  7689#  7772#  7810#  7862#  7881#  7901#  7971#  7991#  8043#  8061#  8131#
                8151#  8182#  8190#  8293#  8343#  8447#  8502#  8573#  8608#  8664#  8717#  8790#
M$TSTL            1#  2098#  3569#  3579#  3593#  3603#  3714#  3724#  3886#  3922#  3932#  3946#  3956#  3995#  4005#
                4043#  4053#  4065#  4075#  4225#  4234#  4257#  4267#  4280#  4290#  4303#  4313#  4326#  4336#  4472#
                4482#  4866#  4870#  4882#  4888#  4896#  4904#  4912#  4922#  4929#  4939#  4943#  4957#  4963#  4971#
                4979#  4987#  4997#  5004#  5014#  5022#  5032#  5039#  5049#  5053#  5066#  5074#  5080#  5088#  5096#
                5106#  5113#  5123#  5131#  5141#  5148#  5158#  5162#  5176#  5184#  5190#  5198#  5206#  5214#  5224#
                5231#  5241#  5249#  5259#  5266#  5276#  5280#  5293#  5301#  5307#  5315#  5323#  5333#  5340#  5350#
                5358#  5368#  5375#  5385#  5389#  5403#  5409#  5417#  5425#  5435#  5442#  5452#  5460#  5470#  5477#
                5487#  5491#  5504#  5512#  5518#  5526#  5534#  5542#  5552#  5559#  5569#  5577#  5587#  5594#  5604#
                5608#  5622#  5628#  5636#  5644#  5654#  5661#  5671#  5675#  5694#  5745#  5751#  5757#  5763#  5783#
                5804#  5822#  5831#  5836#  5855#  5872#  5879#  5883#  5908#  5933#  5942#  5950#  5975#  5982#  6004#
                6017#  6024#  6030#  6057#  6064#  6082#  6098#  6105#  6141#  6148#  6190#  6195#  6203#  6251#  6256#
                6266#  6285#  6297#  6304#  6310#  6329#  6341#  6348#  6354#  6373#  6385#  6392#  6398#  6417#  6429#
                6436#  6442#  6482#  6508#  6517#  6536#  6545#  6564#  6573#  6579#  6609#  6614#  6629#  6634#  6647#
                6652#  6664#  6669#  6677#  6722#  6727#  6738#  6743#  6753#  6780#  6795#  6800#  6811#  6818#  6824#
                6847#  6862#  6867#  6877#  6884#  6890#  6910#  6925#  6930#  6940#  6947#  6953#  6986#  7004#  7009#
                7020#  7027#  7034#  7063#  7066#  7080#  7087#  7094#  7100#  7113#  7120#  7123#  7147#  7171#  7176#
                7187#  7192#  7200#  7203#  7303#  7308#  7332#  7337#  7350#  7402#  7451#  7509#  7518#  7559#  7572#
                7588#  7611#  7623#  7630#  7689#  7772#  7810#  7862#  7881#  7901#  7971#  7991#  8043#  8061#  8131#
                8151#  8182#  8190#  8293#  8343#  8447#  8502#  8573#  8608#  8664#  8717#  8790#
M$WORD            1#  2098#  2174#  2183   2240#  2242   2243   2244   2245   2246   2247   2248   2249   2250   2251
                2252   2253   2254   2255   2256   2257   2258   2259   2260   2261   2262   2263   2264   2265   2266
                2267   2268   2269   2270   2271   2272   2273   2274   2275   2276   2277   2278   2279   2280   2281
                2282   2283   3569#  3570   3571   3572   3579#  3580   3581   3582   3593#  3594   3595   3596   3603#
                3604   3605   3606   3714#  3715   3716   3717   3724#  3725   3726   3727   3886#  3887   3888   3889
                3922#  3923   3924   3925   3932#  3933   3934   3935   3946#  3947   3948   3949   3956#  3957   3958
                3959   3995#  3996   3997   3998   4005#  4006   4007   4008   4043#  4044   4045   4046   4053#  4054
                4055   4056   4065#  4066   4067   4068   4075#  4076   4077   4078   4225#  4226   4227   4228   4234#
                4235   4236   4237   4257#  4258   4259   4260   4267#  4268   4269   4270   4280#  4281   4282   4283
                4290#  4291   4292   4293   4303#  4304   4305   4306   4313#  4314   4315   4316   4326#  4327   4328
                4329   4336#  4337   4338   4339   4472#  4473   4474   4475   4482#  4483   4484   4485   5942#  5943
                5944   5945   5975#  5976   5977   5978   6017#  6018   6019   6020   6057#  6058   6059   6060   6098#
                6099   6100   6101   6141#  6142   6143   6144   6190#  6191   6192   6193   6251#  6252   6253   6254
                6297#  6298   6299   6300   6341#  6342   6343   6344   6385#  6386   6387   6388   6429#  6430   6431
                6432   6508#  6509   6510   6511   6536#  6537   6538   6539   6564#  6565   6566   6567   6609#  6610
                6611   6612   6629#  6630   6631   6632   6647#  6648   6649   6650   6664#  6665   6666   6667   6722#
                6723   6724   6725   6738#  6739   6740   6741   6795#  6796   6797   6798   6811#  6812   6813   6814
                6862#  6863   6864   6865   6877#  6878   6879   6880   6925#  6926   6927   6928   6940#  6941   6942
                6943   7004#  7005   7006   7007   7020#  7021   7022   7023   7080#  7081   7082   7083   7113#  7114
                7115   7116   7171#  7172   7173   7174   7187#  7188   7189   7190   7303#  7304   7305   7306   7332#
                7333   7334   7335   7509#  7510   7511   7512   7559#  7560   7561   7562   7572#  7573   7574   7575
```

```
        7588#   7589    7590    7591    7611#   7612    7613    7614    7623#   7624    7625    7626    7862#   7863    7864
        7865    7881#   7882    7883    7884    7971#   7972    7973    7974    8043#   8044    8045    8046    8131#   8132
        8133    8134    8182#   8183    8184    8185    8293#   8294    8295    8296    8821#   8826#   8901    8902
M$XFER     1#   2098#
OPEN       1#   2098#
POINTE     1#   2098#   2126
PRINTB     1#   2098#   4860    4876    4884    4890    4898    4951    4959    4965    4973    5061    5068    5076    5082
        5170    5178    5186    5192    5200    5288    5295    5303    5309    5397    5405    5411    5499    5506    5514
        5520    5528    5616    5624    5630
PRINTF     1#   2098#   5874    6477    6513    6541    6569
PRINTS     1#   2098#
PRINTX     1#   2098#   4906    4914    4924    4931    4981    4989    4999    5006    5016    5024    5034    5041    5090
        5098    5108    5115    5125    5133    5143    5150    5208    5216    5226    5233    5243    5251    5261    5268
        5317    5325    5335    5342    5352    5360    5370    5377    5419    5427    5437    5444    5454    5462    5472
        5479    5536    5544    5554    5561    5571    5579    5589    5596    5638    5646    5656    5663
READBU     1#   2098#
READEF     1#   2098#   5743    5749    5755    5761
RFLAGS     1#   2098#
SETPRI     1#   2098#   5820    5931
SETVEC     1#   2098#
SLASH      1#   2098#
STARS      1#   2098#
SVC        1#   2096#   2097
XFER       1#   2098#
XFERF      1#   2098#
XFERT      1#   2098#

. ABS.  036054     000

 ERRORS DETECTED:  0

 CZDMRF/I,CZDMRF.SEQ/CRF/SOL/NL:TOC=SVC34R.MLB,CZDMRF.P11
 RUN-TIME: 35 45 4 SECONDS
 RUN-TIME RATIO: 144/85=1.6
 CORE USED:  19K  (37 PAGES)
```